Piotr Jędrzejowicz
Ngoc Thanh Nguyen
Kiem Hoang (Eds.)

Computational Collective Intelligence

Technologies and Applications

LNAI 6923

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

2 Part II

Springer

Lecture Notes in Artificial Intelligence 6923

Subseries of Lecture Notes in Computer Science

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 II

Springer

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. Wyspianskiego 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 978-3-642-23937-3 e-ISBN 978-3-642-23938-0
DOI 10.1007/978-3-642-23938-0
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

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
Toyoaki 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 Małysiak-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	Wrocław University of Technology, Poland
Radosław Katarzyniak	Wrocław 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	Wrocław University of Technology, Poland
Chin-Feng Lee	Chaoyang University of Technology, Taiwan
Xiafeng 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	Wrocław 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 Sobecki	Wrocław University of Technology, Poland
Bogdan Trawiński	Wrocław 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 Szłapczyńska
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 II

Knowledge Management

Agents and Multi-agent Systems, Mobile Agents and Robotics

Modeling, Simulation and Decision Making

Applications of Computational Collective Intelligence in Shipping

Computational Collective Intelligence in Bioinformatics

CCI-Based Optimization Models

Autonomous and Collective Decision-Making

Collective Intelligence in Web Systems - Web Systems Analysis

Computational Collective Intelligence in Economy

Computational Swarm Intelligence and Applications

Computational Swarm Intelligence - Discovering Relationships in Data

Table of Contents – Part I

Keynote Speeches

Machine Learning and Applications

Collective Computations and Optimization

Web Services and Semantic Web

Social Networks

Complex Systems and Intelligent Applications

Ontology Management

Some Properties of Complex Tree Integration Criteria

Marcin Maleszka and Ngoc Thanh Nguyen

Institute of Informatics, Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50370 Wroclaw, Poland
{marcin.maleszka,ngoc-thanh.nguyen}@pwr.wroc.pl

Abstract. Hierarchical data integration becomes important with the abundance of XML based solution in today world. We have previously introduced a model of hierarchical data – the Complex Tree, a model for its integration and some criteria to guide the integration process. In this paper some of those criteria are further analyzed and their properties are described: 1) the completeness criterion based integration may be decomposed into subtasks with the same final result, 2) the minimality criterion based integration may also be decomposed, but the result, while correct, is not guaranteed to be the same, 3) the precision and relationship completeness criteria are in some cases mutually exclusive.

Keywords: XML integration, complex tree, integration task, integration criteria.

1 Introduction

The XML and other hierarchical formats of data (and knowledge) storage have recently became abundant in both the industry and the theoretical works. At the same time, the task of integrating this type of data became more and more important, i.e. during a company fusion, when the database schemas, employee hierarchies, etc. must be merged. Multiple approaches to this problem exist, but they are mostly based on an ad-hoc basis with algorithms designed for one very specific task.

Our work focuses on creating a theoretical background for hierarchical structure integration, with criteria used to parameterize a well defined process, and with the aim to reduce further application work to simply selecting the proper criteria and used an algorithm developed for them.

In our previous work we have developed a hierarchical structure to be used for integration – the Complex Tree [10]. We were then able to propose a formal definition of an Integration Task for hierarchical structure, with criteria used as parameters [10][11]. We developed multiple criteria [11]: completeness, minimality, precision, optimality, sub-tree agreement, etc. In this paper we focus on describing some properties of those criteria, selected on the basis of importance and understandability.

This paper is organized as follows: the second Section provides a short overview of related works on hierarchical structures integration, criteria and their properties;

P. Jędrzejowicz et al.(Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 1–9, 2011.

Section 3 describes the Complex Tree, a structure we use in our work, as well as the task of integration for Complex Trees; Section 4 provides some properties of Complex Tree integration for select criteria; the last Section concludes the paper with information on possible future work in the area.

2 Related Works

The first criteria for tree integration can be traced back to the 1980s [4][6][15][1], where a general variant of a criterion currently known as *optimality* has been defined in a series of papers by bio-mathematicians and evolutionary biologists. Various methods to integrate *n*-trees (a structure used to represent the result of experiments on evolutionary relationships) were proposed as well as well as the first analysis of the criterion.

More recent works on integration criteria are connected with schema matching. A survey by Rahm and Bernstein [13] presents a possible classification of matching approaches. Schema matching can be defined as a much wider area than just tree integration, but the current widespread use of XML (and it's schemas) made the practical applications of schema matching a case of tree integration.

Do [7] goes further than Rahm, describing some criteria for XML schema integration, in four general areas: input criteria, output criteria, quality measures, and effort criteria. The criteria most interesting in terms of this paper are named by the authors as: *schema information* (a criterion based on the size of input schemas), *schema similarity* (the closer the inputs are, the smaller the space to find the output in will be), *element representation* (if the elements are found in the output), *cardinality* (cardinality of relationships in the output), *precision* and *recall* (similar to those defined in information retrieval).

The authors of [12] names three simple criteria for integrating XML schemas: *completeness* (all elements from the input schemas are in the output schema), *minimality* (each unique element from the input schemas is defined only once in the output schema) and *understandability* (output schema is properly formatted). Although those are based on the criteria created for schema integration, authors modify them to be used with a specifically created hierarchical structure. The same authors in [8] propose criteria more known in the literature like *completeness and correctness* (as described in [11]: "the integrated schema must contain all concepts presented in any component schema correctly; the integrated schema must be a representative of the union of the application domains associated with the schemas"), *minimality* (as described in [11]: "if the same concept appears in more than one component schema, it must occur only once in the integrated schema") and *understandability* (as described in [11]: "the integrated schema should be easy to be understood for the designer and the end user; this implies that among the several possible representations of results of integration allowed by a data model, the most understandable one should be chosen"). Similar propositions may be found in many works, among others in [14] and [2].

The *minimality* criterion was thoroughly analyzed by Batista and Salgado in [3] and Comyn-Wiattiau and Bouzeghoub in [5].

For ontologies, similar criteria were described, for example in [16], where the authors define *legibility* (with two sub-criteria: *minimality* – every element appears only once; *clarity* – it is easily readable), *simplicity* (the minimal possible number of elements occur), *correctness* (all elements are properly defined), *completeness* (all relevant features are represented) and *understandability* (it is simple to navigate by the user). The difference with schema integration, is that for ontologies the scope of transformation is much larger during the process. This is the result of not only the amount of knowledge included in the integrated ontology, which is often greater than the sum of knowledge represented in input ontologies, but also of the structure of the output, which might be very different from each other. Additionally, the criteria are defined to describe what the user would gain after the transformation and not to calculate how mathematically correct the effect would be.

3 Complex Tree Model and Integration Task

For purposes of our research, we have developed a hierarchical structure called the Complex Tree [10], which is a tree with complex vertices and additional information. The structure can be used to model most standard hierarchical structures, like XML (and its schemas), n-trees, some ontologies. In this Section we present the currently used definition of the structure, as well as the definition of the Integration Task, that we analyze in later Sections.

3.1 Complex Tree

The Complex Tree was first defined in [10]with further refinements done in [11]. Additional research done since then indicates that using multisets in place of classical sets allows for better properties. The definition presented here is a modified version of one used in [11], with multisets in place of sets.

Definition 1. ***Complex Tree*** **(CT)** *is a tuple* $CT = (T,S,V,E)$, *where*:

T – *a multiset of node types* $T = \{t_1, t_2, t_3, \dots\}$;

S – *a function determining a sequence of attributes for each type of node* $S(t_i) = (a_{i1}, a_{i2}, \dots, a_{in_i})$;

V – *a multiset of nodes* (*vertices*), *where each node* $v \in V$ *is a triple* $v=(l, t, A)$ *in which the* l *is the label of the node,* $t \in T$ *is the type of the node and* A *is the sequence of attributes and their values for this node* (*as determined by the* S *function*);

E – *a multiset of edges,* $E = \{e = (v, w) : v \in V, w \in V\}$.

The Complex Tree definition given above is a general one, when practical uses differentiate between Complex Tree Schema and Complex Tree Instance (i.e. as generalizations of XSD and XML). While the rest of this paper is based on this general definition, here we will explain the specific versions of the Complex Tree.

Complex Tree Schema (CTS), like database schema or XML Schema, is a structure that describes the possible outlook of multiple structure it is representing (Complex Tree Instances (CTI). As such, the CTS has less actual nodes and no attribute values – no actual data. On the other hand the CTS may have some types that will not occur in the instance of this schema.

Complex Tree Instance, like the actual database table or XML document, is a structure that stores the data or knowledge. As such, it must not have all the types from the schema, but all the attributes must have determined values (even if it represents unknown) and the structure of the nodes may become very complex.

We represent the *i*-th Complex Tree Schema in an ordered set as $CT^{(i)}$, where:

$$CT^{(i)} = (T^{(i)}, S^{(i)}, V^{(i)}, E^{(i)}) .$$

Each CTS may have multiple Complex Tree Instances, we represent the *j*-th CTI of *i*-th CTS as $CT^{(i)}_{(j)}$, where:

$$CT^{(i)}_{(j)} = (T^{(i)}_{(j)}, S^{(i)}_{(j)}, V^{(i)}_{(j)}, E^{(i)}_{(j)}) .$$

For ease of readability we will also use the following notation to indicate CTIs:

$$CT_j = (T_j, S_j, V_j, E_j) .$$

3.2 Integration Task for CTI

As the integration of both CTI and CTS are quite similar, here we present only the Integration Task for CTI. Details on the differences between both Tasks and the definition of CTS integration may be found in [10][11]. The Integration Task for CTI is defined as follows:

The input of the integration process is *n* Complex Trees CT_1, CT_2, …, CT_n from the same schema.

$$CT_1 = (T_1, S_1, V_1, E_1), \ldots, CT_n = (T_n, S_n, V_n, E_n) .$$

The output of the integration process is one Complex Tree CT^*, connected with input structures by a group of criteria.

$$CT^* = (T^*, S^*, V^*, E^*) .$$

The parameters of the integration task are the integration criteria $K=\{K_1, K_2, \ldots K_m\}$ tying CT^* with CT_1, $CT_2, \ldots, CT_n$, each at least at a given level $\alpha_1, \ldots, \alpha_n$

$$K_i: M_j(CT^* | CT_1, CT_2, \ldots, CT_n) \geq \alpha_i ,$$

where M_j is a measure of a criterion (as defined in [11]); some of the possible measures are presented in Section 4 of this paper (more criteria may be found in [10][11]). The specific notation of the measure arguments was selected for ease of readability and is used thorough this paper.

Alternatively the integration process may be defined as a function *I*:

$$I: 2^{CTI} \rightarrow CTI ,$$

where *CTI* is the space (set) of all possible Complex Tree Instances.

Extending this notation further, we have that:

$$I_{\{K_1,\dots,K_m\}}(CT_1,\dots,CT_n) = \mathbb{C},$$

where $K_1, \dots, K_m$ are the criteria for the integration process and $\mathbb{C} = \{CT^*\}$ is the set of solutions (for many criteria more than one result is possible).

In some cases the criteria are so specific, that only one solution exists:

$$I_{\{K_1,\dots,K_m\}}(CT_1,\dots,CT_n) = CT^*.$$

These cases are the most interesting from the practical standpoint, as shown in Section 4.1.

4 Properties of Integration Criteria

The integration task defined in Section 3.2. may be parameterized by multiple possible criteria and their measures. Some of these criteria have specific properties, and some are in complex relations with each other. Here we present a selection of those parameters.

4.1 Decomposition of Completeness-Based Integration

In our previous work ([10][11]) we have defined multiple completeness criteria using set measures. Most of those can be subject to decomposition (as a sum of sets is subject to it), here we present only the evaluation for the most commonly used structure completeness.

Definition 2. ***Structure Completeness*** *is a measure representing the number of nodes* (*identified by types and labels*) *from the input structures which remain after the integration.*

$$C_S(CT^* \mid CT_1,\dots,CT_n) = \frac{1}{card\{V_1 \cup \dots \cup V_n\}} \sum_{v \in V_1 \cup \dots \cup V_n} m_V(v, V^*),$$

where:

$$m_V(v, V^*) = \begin{cases} 1, \text{ if } v \in V^* \\ 0, \text{ if } v \notin V^* \end{cases}.$$

We will now use the notation introduced in the previous Section to describe an integration process that meets the following criterion: structure completeness is equal to 1. There is only a single solution possible for the integration meeting this criterion, so the function may be depicted as follows:

$$I_C(CT_1,\dots,CT_n) = CT^*.$$

Let also, for simplicity:

$$I_C(CT_1, CT_2) \stackrel{\text{def}}{=} CT_1 \oplus CT_2.$$

Theorem 1

$$I_C(CT_1, \dots, CT_n) = I_C(I_C(CT_1, \dots, CT_l), \dots, I_C(CT_p, \dots, CT_n)).$$

The integration task meeting the criterion C_S=1 may be decomposed to any number of sub-tasks.

The integration task may be decomposed down to integration of pairs of Complex Trees, which is the fastest in terms of implementation. With multiple integration of pairs providing the same result as the one-step integration, the practical use of this class of criteria becomes more manageable.

For even *n* this may be defined as follows:

$$I_C(CT_1, CT_2, \dots, CT_n) = I_C(CT_1 \oplus CT_2, CT_3 \oplus CT_4, \dots, CT_{n-1} \oplus CT_n),$$

while for odd *n*:

$$I_C(CT_1, CT_2, \dots, CT_n) = I_C(CT_1 \oplus CT_2, CT_3 \oplus CT_4, \dots, CT_{n-2} \oplus CT_{n-1}, CT_n).$$

Outline of the Proof. The integration following the given criterion requires that all elements from the input are in the result. The proof uses the properties of multiset operations to show that the decomposed integration gives the same result.

Full proof is not provided due to paper length limitation.

Additionally, the sum of sets is an commutative operation and the order of operations in the sum may be changed. Thus the order of pairs used for integration may be changed.

Conclusion. *The integration operation that keeps structural completeness equal to 1 is commutative.*

$$(CT_1 \oplus CT_2) \oplus CT_3 = CT_1 \oplus (CT_2 \oplus CT_3).$$

4.2 Decomposition of Minimality-Based Integration

A study conducted among computer science students for the purposes of [11] showed that *minimality* is commonly associated with integration of hierarchical structures. Basing on this study we have developed the measure of *minimality* as a variant of more known *precision* and defined it on a general level. Next we derived some basic *minimality* measures that can be calculated for any tree. As it happens, minimality-based integration may also be done in multiple steps. Here we present one of possible *minimality* measures and the proof of the decomposition possibility.

Definition 3. ***Minimality*** *is a basic measure which enables comparing the size of integrated tree with input trees, where the output tree should be no larger than the sum of the input trees.*

Minimality in terms of the number of nodes may be defined as follows:

$$M_V(CT^* \mid CT_1,...,CT_n) = \begin{cases} 1 \ \ if \ \ card\{V^*\} \le card\{V_1\} + ... + card\{V_n\} \\ \dfrac{card\{V_1\} + ... + card\{V_n\}}{card\{V^*\}} \quad otherwise \end{cases},$$

We will now use the notation introduced in Section 3 to describe an integration process that meets the following criterion: minimality is equal to 1. There are multiple solutions possible for the integration meeting this criterion, so the function may be depicted as follows:

$$I_M(CT_1, \dots, CT_n) = \mathbb{C}.$$

Theorem 2

$$I_M(CT_1, \dots, CT_n) \supseteq I_M(I_M(CT_1, \dots, CT_l), \dots, I_M(CT_p, \dots, CT_n)).$$

The integration task meeting the criterion M=1 may be done as a sequence of integration tasks for pairs of Complex Trees.

Note. Unlike the completeness-based integration the result is not guaranteed to be the same.

Outline of the Proof. Minimality is only equal to 1, if

$$card\{V^*\} \le card\{V_1\} + card\{V_2\} + \cdots + card\{V_n\}.$$

The proof uses the transitivity of the *less than* operator to show that the result of the multistep integration is correct in terms of the criterion (and thus a subset of all possible solutions obtained in the single step integration).

Full proof is not provided due to paper length limitation.

4.3 Mutual Exclusiveness of Precision and Relationship Completeness

An algorithm created for us and presented in [10], under analysis in our later work [11], shows another property of integration criteria, that is: in some cases the maximum value of precision and relationship completeness is mutually exclusive.

Definition 4. ***Precision*** *is a measure representing how many new elements were introduced during the integration and if no duplicate information is in the output tree.*

$$P(CT^* \mid CT_1,...,CT_n) = \alpha_P \cdot \frac{card\{T_1 \cup ... \cup T_n\}}{card\{T^*\}} + \beta_P \cdot \frac{card\{V_1 \cup ... \cup V_n\}}{card\{V^*\}}.$$

Definition 5. ***Relationship Completeness*** *is a measure representing how many of the relationships from the input trees occur in the integrated tree.*

$$C_{rel}(CT^* \mid CT_1,...,CT_n) = \frac{1}{card\{E_1 \cup ... \cup E_n\}} \sum_{e \in E_1 \cup ... \cup E_n} m_e(e, E^*),$$

where:

$$m_e(e, E^*) = \begin{cases} 1, if\ e \in E^* \\ 0, if\ e \notin E^* \end{cases}.$$

For purposes of this paper we will assume α_P=0, β_P=1. Similar properties may be drawn in any case where β_P>0.

Using the notation introduced in Section 3, we have that the integration process meeting the criterion of precision equal to 1 is (for simplicity of notation, we will temporarily assume that there is only one result of the precision based integration):

$$I_P(CT_1, \dots, CT_n) = CT^*,$$

and the integration process meeting the criterion of relationship completeness equal to 1 is:

$$I_R(CT_1, \dots, CT_n) = \widetilde{CT}^*.$$

Theorem 3

$$\exists_{CT_1, CT_2} : I_P(CT_1, CT_2) \neq I_R(CT_1, CT_2).$$

For some pairs of Complex Trees it is not possible for the result of integration of those trees to have both precision and relationship completeness equal to 1.

An example of the structures generating the conflict are Complex Trees CT_1 and CT_2 with identical multisets of vertices and a single vertex as a child of different vertices.

Outline of the proof. The proof is based on the premise of multiset addition operation. The multiset generated by precision-based integration has different cardinality than the one generated by relationship completeness based integration.

Full proof is not provided due to paper length limitation.

5 Future Work

The properties presented in this paper are only a select few of, but they are, especially the decomposition of completeness-based integration, among the most important. The possibility to obtain the same result with integrating smaller subsets of input structures in multiple steps allows for more robust practical applications of the theory here presented.

Our future work will focus on finding more properties of the criteria proposed earlier, especially if those properties have any important theoretical purpose or large practical impact. The criteria proposed so far are also mainly structure-based and we aim, in our future work, to extend this theory into semantic area.

We have also started developing algorithms using the criteria described in this and our earlier papers. We aim to develop more of those, among them algorithms that may be decomposed for practical applications.

Acknowledgements. This paper was partially supported by Polish Ministry of Science and Higher Education under grant no. N N519 407437.

References

[1] Adams, E.N.: N-Trees as Nestrings: Complexity, Similarity, and Consensus. Journal of Classification 3, 299–317 (1986)
[2] Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Computing Surveys (CSUR) 18(4), 323–364 (1986)
[3] Batista, M.d.C.M., Salgado, A.C.: Minimality Quality Criterion Evaluation for Integrated Schemas. In: Proceedings of 2nd International Conference on Digital Information Management, ICDIM 2007, pp. 436–441 (2007)
[4] Barthelemy, J.P., McMorris, F.R.: The Median Procedure for n-Trees. Journal of Classification 3, 329–334 (1986)
[5] Comyn-Wattiau, I., Bouzeghoub, M.: Constraint confrontation: An important step in view integration. In: Loucopoulos, P. (ed.) CAiSE 1992. LNCS, vol. 593, pp. 507–523. Springer, Heidelberg (1992)
[6] Day, W.H.E.: Optimal Algorithms for Comparing Trees with Labeled Leaves. Journal of Classification 2, 7–28 (1985)
[7] Do, H.H., Melnik, S., Rahm, E.: Comparison of Schema Matching Evaluations. In: Chaudhri, A.B., Jeckle, M., Rahm, E., Unland, R. (eds.) NODe-WS 2002. LNCS, vol. 2593, pp. 221–237. Springer, Heidelberg (2003)
[8] Madria, S., Passi, K., Bhowmick, S.: An XML Schema integration and query mechanism system. Data & Knowledge Engineering 65, 266–303 (2008)
[9] Maleszka, M., Nguyen, N.T.: Methods for XML documents integration. In: Grzech, A., Swiatek, P., Drapala, J. (eds.) Advances in Systems. Warsaw "Exit", pp. 333–342 (2010)
[10] Maleszka, M., Nguyen, N.T.: A model for Complex Tree integration tasks. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011, Part I. LNCS, vol. 6591, pp. 36–46. Springer, Heidelberg (2011)
[11] Maleszka, M., Nguyen, N.T.: A Method for Complex Hierarchical Data Integration. In: Cybernetics and Systems (to appear)
[12] Passi, K., Lane, L., Madria, S., Sakamuri, B.C., Mohania, M., Bhowmick, S.: A Model for XML Schema Integration. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, pp. 193–202. Springer, Heidelberg (2002)
[13] Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Jour-nal 10, 334–350 (2001)
[14] Schmitt, I., Saake, G.: A comprehensive database schema integration method based on the theory of formal concepts. Acta Informatica 41(7/8), 475–524 (2005)
[15] Stinebricker, R.: s-Consensus Trees and Indices. Bulletin of Mathematical Biology 46, 923–935 (1984)
[16] Trinkunas, J., Vasilecas, O.: Ontology Transformation: from Requirements to Conceptual Model. Scientific Papers, University of Latvia. Computer Science and Information Technologies, vol. 751, pp. 52–64 (2009)

Semantically Enhanced Collaborative Filtering Based on RSVD

Andrzej Szwabe, Michal Ciesielczyk, and Tadeusz Janasiewicz

Institute of Control and Information Engineering, Poznan University of Technology
pl. Marii Skladowskiej-Curie 5, Poznan, Poland
{andrzej.szwabe,michal.ciesielczyk,tadeusz.janasiewicz}@put.poznan.pl

Abstract. We investigate a hybrid recommendation method that is based on two-stage data processing – first dealing with content features describing items, and then handling user behavioral data. The evaluation of the proposed method is oriented on the so-called find-good-items task, rather than on the low-error-of-ratings prediction. We focus on a case of extreme collaborative data sparsity. Our method is a combination of content features preprocessing performed by means of Random Indexing (RI), a reflective retraining of preliminary reduced item vectors according to collaborative filtering data, and vector space optimization based on Singular Value Decomposition (SVD). We demonstrate that such an approach is appropriate in high data sparsity scenarios, which disqualify the use of widely-referenced collaborative filtering methods, and allows to generate more accurate recommendations than those obtained through a hybrid method based on weighted feature combination. Moreover, the proposed solution allows to improve the recommendation accuracy without increasing the computational complexity.

Keywords: Hybrid recommender systems, dimensionality reduction, information retrieval, random indexing, singular value decomposition.

1 Introduction

Hybrid recommender systems were introduced in order to effectively deal with the behavioral data sparsity [1] [3]. This kind of application is designed to exploit the complementary advantages of content-based filtering and collaborative filtering. In the relevant literature [2], one may find that one of the most challenging problems, which has to be considered when designing a state-of-the-art hybrid recommender system, is that typically there are many more content features describing items than the corresponding behavioral data, preferably collected during the system operation. However, according to the authors of widely-cited publications mentioned above, sparser behavioral data is more valuable (from the perspective of their usefulness for generating high quality recommendations) than the related content features (usually much denser). Therefore, in order to effectively use these two mentioned data sources, architectural separation of data processing functions may be considered as a good design pattern.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 10–19, 2011.

So far, many solutions have been proposed in the area of hybrid recommendation techniques [1] [4] [8] [11]. The main practical distinction in these techniques corresponds to the way how the data processing stages are defined. The most widely referenced (considering the experiments on popular data sets [11] [18]) and also the simplest approach is to balance the results of separated data processing stages. Typically, in these so-called weighted hybrid recommendation systems, the collaborative filtering data and content features are initially processed independently, and then the dual predictions are combined in the final stage of the algorithm. More advanced hybrid systems are usually based on joint processing of content and collaborative features or are realized as cascade systems [3] [12]. An example of a cascade system may be found in [2], where, in the first stage, a content-based model is built for each user, and then the ratings of data are used to combine these features in order to generate the final user profiles, used for recommendation purposes.

SVD is the most widely used matrix factorization method for collaborative filtering and text retrieval. Consequently, it is also used in hybrid recommendation systems. Random Indexing (RI) and Reflective Random Indexing (RRI) have been introduced [6] [15] in order to meet the problem of computational complexity and storage requirements of SVD. However, neither RI nor RRI is capable of producing a full set of accurately approximated SVD matrices. As a result, these methods cannot be used for a low-dimensional input matrix reconstruction in a high accuracy recommender system based on rate estimation [10] [17]. To our knowledge, the only method that is able to provide a full set of approximated SVD matrices is Randomized Singular Value Decomposition (RSVD) [5].

Our hybrid recommendation method – herein referred to as Semantically Enhanced Collaborative Filtering based on RSVD (SECF-RSVD) – addresses the above-mentioned issues. It should be mentioned that the term 'semantically enhanced' should be understood as defined by the authors of [11]. Therefore, the meaning of this term is different from the way it is used by authors of ontology-oriented methods.

2 Objectives

In this paper, we investigate a novel approach to feature integration, which is based on reflective (i.e., two-step) item vector retraining performed according to collaborative filtering data. This technique resembles reflective data processing, which is the core element of the RSVD-RRI method [5]. It is worth noting that, to the best of our knowledge, no author has investigated the applicability of the reflective context vector retraining technique to feature integration for recommender systems.

Building a model of collaborative similarity from a model of content-based inter-item relations is considered by several authors of relevant publications [11] [16] to be the most promising hybrid recommendation technique. In this paper, we focus on an approach where the item content features are reflectively retrained according to the collaborative characteristics, which is clearly different

to the case where both data sources are individually processed and combine the results at the end. However, in both solutions, the architectural separation of the processing path allows to add new behavioral data (i.e., from a user tracking system), while having a static model of the item content features. This may be a very practical feature, since the user data are updated much more frequently (usually in real time) than the textual information describing the items. Because of the fact that we want to infer as much as possible from both data sources, one of the purposes of our research is to investigate whether it is necessary (in order to obtain a better recommendation quality) for a hybrid recommender system to perform such a two-stage processing.

This paper, in contrast to many publications in the relevant literature [18], focuses on a case of extremely high data sparsity of collaborative filtering data. A high data sparsity scenario, despite being the most challenging one, is also the most typical in the area of e-commerce. We assume that usually an e-commerce recommender system, especially in an early stage of its operation, does not have more than a few ratings per user [13] [14].

Moreover, in order to develop a useful large-scale recommender system, high accuracy of low-dimensional recommendations has to be combined with high computational efficiency. Therefore, our research is oriented on developing a technique which effectively copes with high sparsity of behavioral data, and which at the same time is not more computationally complex than the state-of-the-art collaborative filtering methods presented in the relevant literature.

3 Proposed Method

We propose a new class of recommender systems involving multi-stage data processing, which is typical of hybrid recommender systems, and which is also the core element of dimensionality reduction methods based on reflective random indexing [5] [6]. First-stage item vectors are obtained by means of content feature dimensionality reduction based on random indexing [15]. Then they are reflectively retrained according to collaborative filtering data in a way resembling retraining of context vectors used in the reflective random indexing method [6], and finally SVD is applied to obtain vectors in a low-dimensional space. Thanks to the reflective way of data processing, and the ability of sparse data processing (which is typical of all dimensionality reduction methods based on RI/RRI), SECF-RSVD enables the computationally efficient integration of collaborative and item data (at different processing stages). Despite a radical increase in the amount of processed data, we were able to achieve similar execution times to the RSVD method based only on collaborative data presented in [5].

In this paper we use a notation, which is consistent with the naming convention proposed in [6] and followed in [5]. Let m denote the number of users in the system, let n denote the number of items, let p denote the number of features per item. We use d to denote the number of dimensions used for random vectors representation (determining the extent to which the preliminary dimensionality reduction is performed), s to denote the number of non-zero values in

every random vector (so-called seed, $s << d$), and k to denote the final number of dimensions of the output matrices ($k < d$). The user-item matrix containing binarized ratings is denoted as $C_{m \times n}$, and the term-item matrix representing the content features is denoted as $T_{p \times n}$.

Let R be a random $p \times d$ matrix, where each row represents a random vector (also referred to as index vectors). The values of matrix R are preliminarily set so that for each row vector there are exactly s coordinates, and which are selected uniformly at random, and set to be equal to $s^{-0.5}$ or $-s^{-0.5}$ (with the probability equal to 0.5). Let $D = [d_{i,j}]_{n \times d}$ be a matrix, where each row represents a context vector, and is produced as follows:

$$D = T^T R. \tag{1}$$

The obtained context vectors are then normalized according to the formula:

$$D' = norm(d), \tag{2}$$

where $d'_{i,j} = d_{i,j}(\sum_{j=1}^{k} {d_{i,j}}^2)^{0.5}$.

The normalized context vectors are retrained (3) using C - an $m \times n$ binary user-item matrix, resulting in new index vectors, as presented in:

$$R' = CD'. \tag{3}$$

The $m \times d$ matrix prepared in this way is, in turn, normalized (4) and used to generate new context vectors (the new version of matrix D) (5).

$$\widehat{R} = norm(R'). \tag{4}$$

$$\widehat{D} = C^T \widehat{R}. \tag{5}$$

When the $n \times d$ context vectors matrix $\widehat{D}$ is prepared (of significantly lower dimensionality than the input matrices T and C), SVD is used to decompose it into a set of three matrices:

$$\widehat{D}^T = U'_{tmp} S' {V'}^T. \tag{6}$$

U'_{tmp} is a matrix containing the left singular vectors of $\widehat{D}^T$, V' is a matrix containing the right singular vectors of $\widehat{D}^T$ and S' is a diagonal matrix with the diagonal values equal to the singular values of $\widehat{D}^T$. Only two of these matrices – in this case V' and S' – are used in the further steps of the RSVD procedure.

In order to reduce the impact of the distortion caused by dimensionality reduction, matrices S' and V' are truncated, so that the least significant concepts $(d - k)$ are removed. This step represents the second stage of RSVD-based dimensionality reduction. As a result of such an operation, $k \times k$ matrix S'_k and $n \times k$ matrix V'_k are produced.

In the next step, $m \times k$ matrix U'_k (containing vectors representing user profiles) is calculated by multiplying the input matrix C, matrix V'_k (obtained as a result of the previous step of the RSVD procedure), and matrix ${S'_k}^{-1}$:

$$U'_k = CV'_k {S'_k}^{-1}. \tag{7}$$

Many of the most accurate collaborative filtering methods are based on the estimation of the missing entries of the input matrix. In order to reconstruct the approximated input matrix C_k, one has to perform matrix multiplication as in (8), which is the last step of the RSVD procedure.

$$C_k = U'_k S'_k {V'_k}^T. \tag{8}$$

4 Evaluation Methodology

4.1 Data Sets and Recommendation Accuracy Measure

We evaluated all the methods by using one of the most widely referenced CF data sets the MovieLens ML100k set [17]. The ML100k data set contains 100,000 ratings for 1682 movies given by 943 unique users. In companion to movie ratings data, we use The Internet Movie Database [19] resources as a source of content features we took into account the movie title, the genre, the director, the cast and the release year. As far as the release year is concerned, we performed discretization to represent a set of intervals as attributes. Finally, we selected only those attributes which reoccur in at least two movies, and we ended up with a semantic vector, with 5058 dimensions, representing each movie.

We randomly partitioned the ML100k data set into two pairs of a train set and a test set, as proposed in [17]. The data was divided according to the specified training ratio, denoted by x. This was done in order to address the high sparsity scenario while using the ML100k data set, but we have additionally used several values of training ratio below the range investigated in [17] and [11]. To compensate for the impact that the randomness of the data set partitioning has on the results of the presented methods, all figures in this paper show series of values that represent the averaged results of 10 individual experiments.

Following the approach proposed in a widely cited paper by Sarwar et al [17], we evaluate our method by using the F1 measure. In particular, in the case of our experiments, we have selected F1@10 (just as in [11] [17]), which means that a system recommends the Top-10 items for every user.

4.2 Evaluated Recommendation Methods

We have compared the accuracy of our method to the accuracy of other methods presented in the relevant literature. In order to perform such comparison, we have developed our implementations of the SVD-based kNN recommendation method proposed in [17] (in this paper referred to as 'kNN-Sar'), the semantically enhanced collaborative filtering method proposed in [11] (in this paper referred to as 'SECF-Mob'), and the PureSVD algorithm which was reported [7] as outperforming the state-of-the-art RMSE-oriented approaches.

We compared our solution to the system presented in [11], which is the most widely referenced hybrid recommendation solution that has been evaluated using MovieLens ML100k. However, we follow the IR-oriented (based on F1 measure)

approach to recommendation quality evaluation, rather than performing MAE-based evaluation (used by the author of SECF-Mob). Therefore, in order to verify whether our implementation of the SECF-Mob is appropriately similar to the original one presented in [11], we analyzed MAE results obtained when using our implementation before evaluating the method from the perspective of the recommendation accuracy measured by means of F1.

In the PureSVD algorithm, the user vectors are represented as a combination of item vectors, without any specific parameterization. Despite the fact that a PureSVD method is quite simple and achieves poor results in terms of RMSE [7], it outperforms (in terms of F1) several more sophisticated RMSE-oriented methods.

Fig. 1 presents the recommendation accuracy results that we have obtained using kNN-Sar and SECF-Mob, which are very similar to those presented in [17] and [11] correspondingly. As [11] contains MAE results rather than F1 results, here we show the ability of our SECF-Mob implementation to produce MAE results close to those presented in [11]. We evaluate the recommendation accuracy from the perspective of the find-good-items task [9], therefore we measure the accuracy of SECF-Mob, using F1.

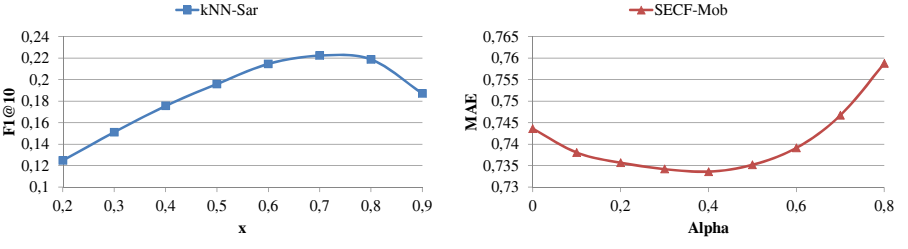


Fig. 1. Accuracy of the kNN-Sar low dimensional neighborhood method (left) and the impact of the weighting parameter (Alpha) on recommendation accuracy of SECF-Mob - MAE results (right) obtained for our implementation of SECF-Mob method

The PureSVD algorithm was evaluated in [7] using a different methodology, which is new. Therefore, so as not to obscure the reasoning introduced here, we do not report our attempts to reproduce the results presented in [7]. However, as this algorithm is rather simple, we were able to achieve results that are close to the original ones without much difficulty.

5 Accuracy of Recommendations Based on SECF-RSVD

In addition to investigating the highest achievable recommendation accuracy, we analyze the proposed method from the perspective of the relation between dimensionality reduction and recommendation quality.

5.1 Recommendation Accuracy Evaluation

The results of the comparison (in the form of F1@10 values), for sparser sets (x values lower than 0.2) and denser data sets (x values higher than 0.1), are presented in Fig. 2 and Fig. 3, respectively. We evaluated the recommendation methods at two values of k (the final number of dimensions used for rate estimation or kNN). We used a lower value (i.e., $k = 4$) for sparser data, and higher (i.e., $k = 14$) for denser data sets.

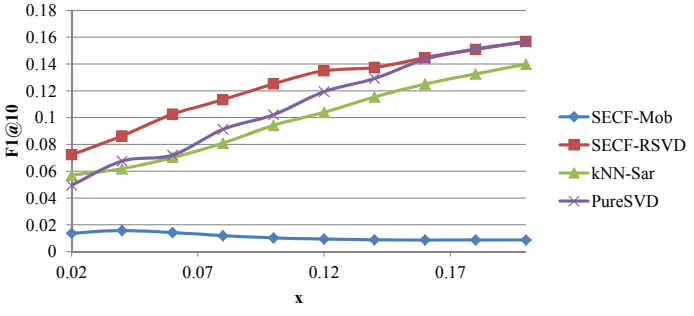


Fig. 2. Accuracy of kNN-Sar, SECF-Mob, PureSVD and SECF-RSVD; F1@10 for recommendations in the 4-dimensional space at x values lower than 0.2

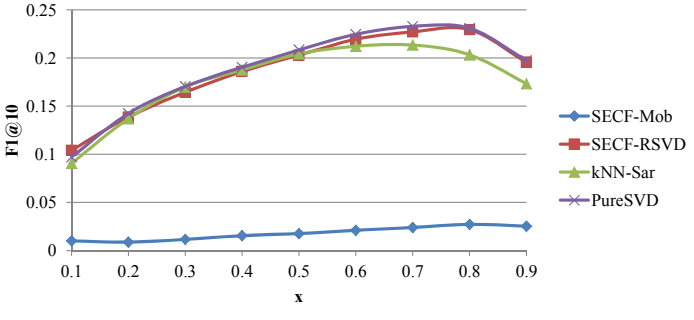


Fig. 3. Accuracy of kNN-Sar, SECF-Mob, PureSVD and SECF-RSVD; F1@10 for recommendations in the 4-dimensional space at x values higher than 0.1

SECF-RSVD is much more accurate (in terms of F1) than the method proposed in [17]. This shows that good MAE results are not necessarily associated with good F1-scores. Although the SECF-Mob method may be used to achieve satisfying MAE results, it has very poor performance when applied to the task of finding the Top-N recommendations.

By combining textual and collaborative data in SECF-RSVD, we were able to obtain better results, as far as low values of the training ratio are concerned, than the two CF methods introduced (kNN-Sar and PureSVD). As we have confirmed

experimentally, SECF-RSVD may be used to achieve recommendation accuracy comparable to that achieved by other methods even while using two times less data (when sparse data sets are considered). In particular, the SECF-RSVD method achieves F1@10 score of 0.1 at x equal to 0.06, whereas the kNN-Sar method achieves the same score at x equal to 0.12.

5.2 Feature Space Dimensionality and Accuracy

We have investigated the way parameter k of the introduced methods is correlated with the recommendation accuracy. The results that we obtained for SECF-RSVD are presented in Fig. 4.

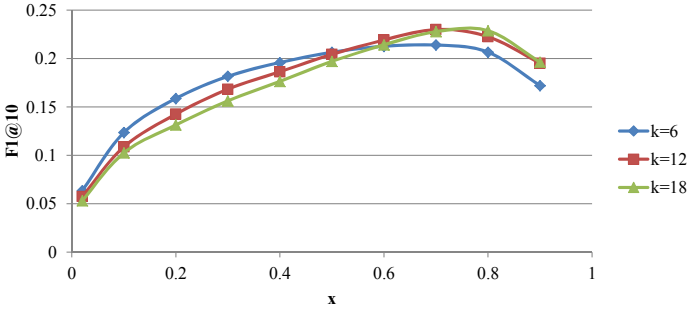


Fig. 4. Recommendation accuracy for SECF-RSVD; each curve representing F1@10 results obtained for a specified value of k and several values of x

The presented method shows relatively high independence of the recommendation accuracy from k. Nevertheless, it is still noticeable that for smaller training ratios, smaller k should be chosen.

5.3 Computational Effectiveness

Due to the fact that SECF-RSVD is based on RI, the complexity of the preliminary data dimensionality reduction [5] may be flexibly adapted to the amount of available computational resources (SECF-RSVD may be flexibly adapted to the number of available computational resources, i.e., memory and processing power, which are directly dependent on parameter d).

In the case when matrix decomposition is applied to the ML100k data set, the execution time of RSVD (closely approximating the SVD-based results) is usually several times shorter than the execution time of a comparable (in terms of F1) SVD application. The accuracy of the compared methods is presented in Fig. 5 as functions of the execution time (on a regular PC).

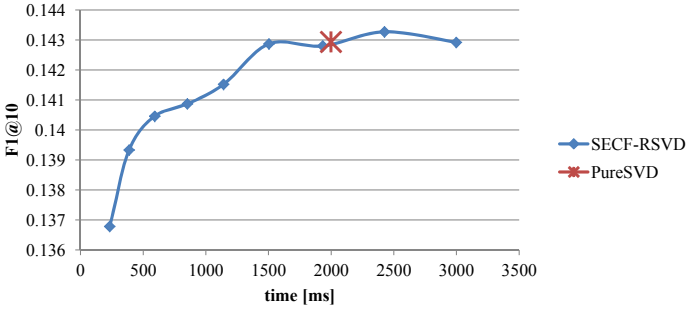


Fig. 5. The trade-off between matrix decomposition quality and efficiency: each series of F1 as a function of execution time, for the case of k=14, s=1 and x=0.2

6 Conclusions

Our approach allows to achieve significantly better results (in terms of F1) than the collaborative semantically-enhanced collaborative filtering technique proposed by Mobasher in [11]. Moreover, in the case of a small amount of training data, we were able to generate better Top-N item recommendations than the methods based only on collaborative filtering data proposed in [17] and [7]. The superiority of our method is especially visible when the most challenging range of low x values is taken into account. For such x values, our method is able to achieve the same recommendation accuracy as that achieved by the widely-referenced methods proposed in [17] and [11].

Thanks to using the technique of random vector retraining, our method offers adjustability of computational resource requirements, while still giving similar or (when carefully configured) slightly better results compared to those obtained when using the SVD-based method. In our experiments, despite the fact that the SECF-RSVD had much more data to process (the content features matrix is several times bigger than the collaborative filtering matrix), a SECF-RSVD system computed recommendations usually meaningfully faster than a SVD-based system.

Acknowledgments. This work is supported by the Polish Ministry of Science and Higher Education, grant N N516 196737.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In: Fifteenth National Conference on Artificial Intelligence, pp. 714–720 (1998)

3. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
4. Campos, L., Fernandez-Luna, J., Huete, J., Rueda-Morales, M.: Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. Int. J. Approx. Reasoning 51, 785–799 (2010)
5. Ciesielczyk, M., Szwabe, A.: RI-based Dimensionality Reduction for Recommender Systems. In: Procs. of 3rd International Conference on Machine Learning and Computing. IEEE Press, Singapore (2011)
6. Cohen, T., Schvaneveldt, R., Widdows, D.: Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. Journal of Biomedical Informatics 43(2), 240–256 (2010)
7. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys 2010), New York, NY, USA, pp. 39–46 (2010)
8. Gunawardana, A., Meek, C.: A unified approach to building hybrid recommender systems. In: RecSys 2009: Proceedings of the third ACM Conference on Recommender Systems, pp. 117–124 (2009)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Trans. Information Systems 22(1), 5–53 (2004)
10. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009)
11. Mobasher, B., Jin, X., Zhou, Y.: Semantically enhanced collaborative filtering on the web. In: Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G. (eds.) EWMF 2003. LNCS (LNAI), vol. 3209, pp. 57–76. Springer, Heidelberg (2004)
12. O'Sullivan, D., Wilson, D., Smyth, B.: Preserving recommender accuracy and diversity in sparse datasets. The International Journal on AI Tools 13(1), 219–235 (2004)
13. Papagelis, M., Plexousakis, D., Kutsuras, T.: Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 224–239. Springer, Heidelberg (2005)
14. Pilaszy, I., Tikk, D.: Recommending new movies: even a few ratings are more valuable than metadata. In: Proceedings of the Third ACM Conference on Recommender Systems (RecSys 2009), New York, NY, USA, pp. 93–100 (2009)
15. Sahlgren, M.: An introduction to random indexing. In: 7th International Conference on Terminology and Knowledge Engineering, Copenhagen, Denmark (2005)
16. Salter, J., Antonopoulos, N.: CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. IEEE Intelligent Systems 21, 35–41 (2006)
17. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Application of dimensionality reduction in recommender system - a case study. In: ACM WebKDD 2000 Web Mining for E-Commerce Workshop, Boston, MA, USA (2000)
18. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce (EC 2000), pp. 158–167. ACM, New York (2000)
19. The Internet Movie Database, http://www.imdb.com

Hybrid Recommendation Based on Low-Dimensional Augmentation of Combined Feature Profiles

Andrzej Szwabe, Tadeusz Janasiewicz, and Michal Ciesielczyk

Institute of Control and Information Engineering, Poznan University of Technology, pl. Marii Curie-Skladowskiej 5, Poznan, Poland
{andrzej.szwabe,tadeusz.janasiewicz,michal.ciesielczyk}@put.poznan.pl
http://www.put.poznan.pl

Abstract. We introduce a new hybrid recommendation method that is based on four data processing steps: 1) preprocessing of content features describing items, 2) preliminary dimensionality reduction applied to user/item vectors expressed in content features space (performed by means of SVD), 3) augmentation of normalized low-dimensional preliminary user/item vectors according to collaborative filtering data and leading to the reconstruction of user/item vectors (based on final item/user vectors and the original input matrix), and 4) the estimation of missing entries in the user-item ratings matrix. In the experiments presented in the paper, we focus on the most challenging case of extreme collaborative data sparsity. We show that a low-dimensional space is suitable for recommendation generation, despite collaborative data sparsity disqualifying the use of methods widely referenced in the relevant literature. In particular, we demonstrate that the proposed low-dimensional feature augmentation method is more effective than the well-known weighted feature combination method.

Keywords: Hybrid recommender systems, dimensionality reduction, feature augmentation, singular value decomposition.

1 Introduction

According to the authors of several widely-cited surveys on recommendation systems, hybrid recommendation methods represent the state of the art in recommender system technology because they deal effectively with the well-known behavioral data sparsity problem [1], [4], [5]. Usually hybrid recommendation methods exploit complementary advantages of the two most popular basic approaches to recommendation - collaborative filtering and content-based filtering [4], [1]. As identified in [2], [7], one of the most important issues that have to be taken into account when designing a hybrid recommender system (as it has a direct impact on recommendation quality) is the fact that the most valuable (from the perspective of the usefulness for achieving high quality recommendation) behavioral/transactional data are usually much sparser than the corresponding

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 20–29, 2011.

data specifying content features (less valuable than collaborative data, but much denser).

In the case of hybrid recommendation solutions proposed so far [13], [1], [9], [6], the difference in the practical value of the two mentioned data sources is usually directly reflected in architectural separation of data processing functions that are performed to obtain a content feature model independently of producing a collaborative user preference model. Such architectural separation enables us to deal effectively with an unequal practical value of collaborative and content data, while also allowing to cope with the unpredictability of the relative importance of the mentioned data sources [24].

Weighted feature combination is the simplest and the most widely referenced hybrid recommendation solution, especially as far as experimentation on popular data sets is concerned [24], [7], [13]. However, the most advanced hybrid recommender systems are based on clearly different two-stage data processing – usually realized in the form of feature augmentation or as a cascade system [5]. For example, in the case of a system proposed in [2], the content-based recommender builds a model for each user and then the user rating data is combined with the product features.

2 Objectives

The main aim of the research presented in this paper is to evaluate a new hybrid recommendation method that is based on low-dimensional feature augmentation and which uses two sources of information – content feature data and collaborative filtering data. We analyze the results of using Singular Value Decomposition (SVD) as a dimensionality reduction method for first-stage content feature modeling. The proposed feature augmentation hybrid approach is based on simple multiplication of a post-processed collaborative filtering data matrix and a matrix of vectors representing preliminary profiles of users (i.e., user profiles obtained as a result of the preliminary dimensionality reduction applied to both content and collaborative data).

Several authors of relevant publications regard building a model of behavioral similarity distinctly from a model of inter-item content feature similarity as the most appropriate approach to hybrid recommendation [13], [19]. Hybrid recommendation methods are frequently based either on separated processing of collaborative and content data and the combination of dual predictions or on incorporating some content-based characteristics into the collaborative approach or vice versa [13], [19], [12]. Architectural separation of processing paths may be additionally motivated by the fact that it is practical to use a dedicated processing path for the update of the behavioral similarity model (i.e., to 'track' new behavioral data). Morover, such a separation is beneficial since item feature data is much less dynamic than behavioral data.

In this paper, we investigate whether the abovementioned separation of data processing functions is necessary in the context of recommender systems based on low-dimensional feature augmentation. Our aim is to exploit information from

both main data sources as much as possible. One of the aims of the research presented in this paper is to show that – at least in some application scenarios – collaborative data does not necessarily have to be neglected as a source of features 'describing' items, in parallel with regular content-describing features (usually having the form of text data and/or metadata).

In contrast to the research objectives set by many authors [21], [22], our research is oriented on the most challenging scenario of extreme sparsity of behavioral data. Some authors – especially those investigating application areas different than e-commerce – assume that it is typical to provide a recommender system with more than a few ratings per user [16]. In contrast to such a perspective, our research is oriented on a scenario of extreme behavioral data sparsity – a scenario which is typical of many e-commerce applications, especially in an early stage of their operation [14].

3 Proposed Method

We propose a new recommendation method in which low-dimensional representation of content features is augmented according to collaborative filtering data. In our approach – referred to as Collaborative Augmentation of Content Feature Profiles (CACFP) – preliminary dimensionality reduction (applied to the collaborative and content features) is performed by means of SVD. The first-stage user vectors (i.e., normal user vectors prepared for the feature augmentation stage) are expressed in a low-dimensional space. The basis of this space consists of eigenvectors corresponding to the most principal first-stage feature components.

3.1 Matrix Preparation

Usually the ratings matrix is denoted as $A_{m\times n}$, where rows represent users and columns represent movies rated by those users [3]. As we investigate hybrid recommendation, we represent the input matrix in a different way: the rating data are accompanied by additional block matrix $B_{p\times n}$, representing the movie features. In this case, matrix $X = [x_{i,j}]$ (which is prepared to be used in the next pre-processing step) is an $r \times n$ matrix, where n is the number of movies and $r = m + p$ is the number of attributes (both collaborative ratings and semantic features). Matrix X is constructed as follows:

$$X_{r\times n} = \begin{bmatrix} A_{m\times n} \\ B_{p\times n} \end{bmatrix} \tag{1}$$

In order to analyze the impact of excluding collaborative data from X, we additionally investigate a variant of CACFP referred to as CACFP-Simple. In such a case, A is used as X.

3.2 Term Weighting

In the next step of CACFP, matrix X is processed using the weight computing method similar to IDF(*inverse document frequency*) [11] referred to

as – IIFW(*inverse item/feature weight*). The procedure $iifw(X)$ works as follows: for each non-zero input matrix entry, its new value is set to be equal to:

$$\log_2 \frac{c}{\sum_{i=1}^{n} x_{i,j}}, \tag{2}$$

where c is the number of columns of the input matrix. IIFW is performed multiple times and in every step (except the first one) the input matrix is transposed before calculation. For example, a two-step-long IIFW weight change operation produces the result shown in equation (3).

$$\begin{aligned} X_{step1} &= iifw(X) \\ X_{tmp} &= iifw(X^T_{step1}) \\ X_{step2} &= X^T_{tmp} \end{aligned} \tag{3}$$

We observed that the best results were achieved when the IIFW method was performed four times. Moreover, we have observed that the optimality of the IIFW iterative application may be quite accurately modeled by means of information-theoretic measures. However, this issue is beyond the scope of this paper.

3.3 Matrix Decomposition and Dimensionality Reduction

In the next step, matrix X', which is obtained as a result of the multiple $iifw$ procedure application, is decomposed using SVD. In concequence of this decomposition (4), three matrices U, S and V are obtained, where U – is a matrix containing the left singular vectors of X' (the eigenvectors of $X'X'^T$), V – is a matrix containing the right singular vectors of X' (the eigenvectors of $X'^T X'$) and S – is a diagonal matrix with the diagonal values equal to the singular values of X'.

$$SVD(X') = U \cdot S \cdot V^T \tag{4}$$

Only the part of matrix U which stores information about users associations (the first m rows) is used in the next step of computation. To capture the major relations and to eliminate the 'noise' from the data, we reduce its dimensionality by using only the first k dimensions of matrix U. The rows of the obtained k-reduced user matrix $U' = [u'_{i,j}]_{m \times k}$ are normalized according to the formula:

$$U_{norm} = norm(U'), \text{ where } (u_{norm})_{i,j} = \frac{u'_{i,j}}{\sqrt{\sum_{j=1}^{k} (u'_{i,j})^2}} \tag{5}$$

3.4 Item Vector and Input Matrix Reconstruction

Having followed the steps defined above, we then reconstruct $n \times k$ matrix V_{reco} using the *reconstV* procedure (6), which represents items in the k-dimensional

concept space. According to the *reconstV* procedure, the row of V_{reco} representing a given item is built as a sum of all the rows of U_{norm} which correspond to the users that rated the given item (according to input matrix A).

$$V_{reco} = reconstV(U_{norm}, A) \tag{6}$$

The final step of the method is to estimate the values of the 'missing' entries of A, which is done by multiplying users matrix U by the reconstructed matrix V_{reco}, as shown in (7).

$$A' = U' \cdot V_{reco}^T \tag{7}$$

4 Evaluation Methodology

As described in this section, to make our research comparable to the results presented by other authors, we use freely-available and widely-referenced data sets. We apply a recommendation quality measure appropriate to the so-called *find good items task* [10] and we compare the accuracy of our method to the accuracy of the widely-referenced methods presented in the relevant literature.

4.1 Data Sets

We used a MovieLens ML100k data set, which is probably the only collaborative filtering data set referred to in the relevant literature that is accompanied by precise specifications of SVD-based recommendation algorithms [20], [17] and reliable experimental results obtained through these algorithms [17], [20], [25]. To couple collaborative data with content attributes we used The Internet Movie Database (IMDb) from which we extracted information about almost every movie appearing in ML100k (like in [15] we noted lack of information for only 5 of them). As attributes of each movie we used: the movie title, the genre, the director, the cast and the release year. In accordance with the method presented in [13] we discretized the release year and represented it as a set of intervals. At the last stage of input matrix preparation, only the attributes which occurred in at least two movies were selected. Finally, we obtained the semantic vector with 5058 dimensions describing each movie. The most typical real-world scenario of a recommender system application is an online retailer [22]. In this context ML100k may be regarded as relatively dense. For example, it is 100 times denser than a customer-product matrix of an e-commerce system presented in [20]. Considering this fact, we processed the ML100k data set according to the partitioning procedure shown in [20], [17], in order to obtain density more typical for a e-commerce application. We used several values of training ratio (as proposed in [20] denoted as x). To reduce the impact of the partitioning on the results of recommendation accuracy evaluation, we averaged the results obtained for each data set division. The figures presented in this paper show series of values that have been obtained by averaging results of 10 individual experiments.

4.2 Recommendation Accuracy Measure

We evaluate the proposed method by using a well-known recommendation accuracy measure of F1 proposed in [18]. It considers both the *precision* and *recall* metric of the recommendation correctness.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{8}$$

Our choice of the recommendation accuracy measure is in line with the observations presented in [8], where the authors remarked that error metrics like MAE(*Mean Absolute Error*) or RMSE(*Root Mean Square Error*) often do not translate into top-N recommendation accuracy.

4.3 Evaluated Recommendation Methods

The results of using the proposed recommendation method were compared with the accuracy of two widely referenced methods. To make an F1-centric comparison possible, we implemented the SVD-based kNN method presented in [20] (in this paper denoted as "kNN-Sar") and the semantically enhanced collaborative filtering method proposed in [13] (denoted as "SECF-Mob").

The method put forward in [13] is one of the most widely known hybrid recommendation solutions evaluated using the ML100k data set and IMDb – a widely referenced source of content feature data [23], [12].

We apply the F1 measure to evaluate the recommendation quality rather than the MAE-based used in [13]. Thus, to verify our implementation of SECF-Mob at first we compared the results known from [13] of MAE measure with those obtained through our implementation. After confirming the compliance of the MAE results, we proceeded with the comparison with the SECF-Mob from the F1 perspective.

Accordingly to the hybrid recommender system classification proposed in [4], SECF-Mob is based on weighted feature combination, which is regarded as less promising approach to hybrid solutions than the feature augmentation systems. Nevertheless, to our knowledge, no feature augmentation recommendation system has been presented in a scenario involving usage a ML100k data set.

The recommendation accuracy obtained through our implementation of the kNN-Sar method, and the SECF-Mob method is very similar to that presented in [20] and [13] (shown on Fig. 1). The algorithm presented in [13] is not fully precisely specified, and some of the method parameters were not given. Therefore, we were not able to precisely reconstruct the content features that were extracted from IMDb. However, as shown in Fig. 1 (right), the results of our implementation are only slightly different from the results presented in [13].

Here, we demonstrate the ability of our SECF-Mob implementation to produce MAE results close to those presented in [13], while in the next section of this paper we measure recommendation accuracy of SECF-Mob method using the F1 measure.

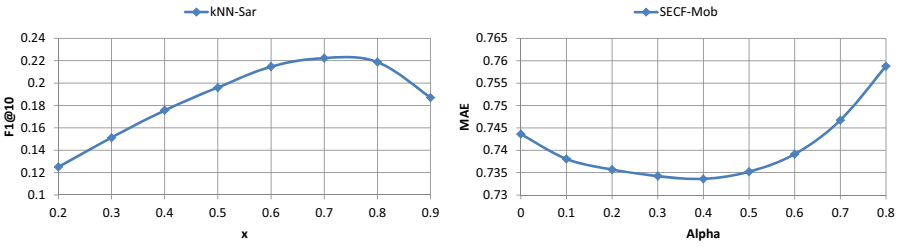


Fig. 1. Recommendation accuracy for the kNN-Sar low dimensional neighborhood method (left) and the impact of the weighting parameter (*Alpha*) on recommendation accuracy of SECF-Mob (MAE results obtained for our implementation of SECF-Mob method proposed in [13])

5 Accuracy Evaluation Results

We evaluated the recommendation methods at two values of k (the final number of dimensions used for rate estimation). We used $k = 4$ for sparser data sets (of x values lower than 0.2), and $k = 14$ (the optimum found by the authors of [20]) for denser data sets (x values higher than 0.1). The results are presented in Fig. 2 and Fig. 3 respectively in the form of F1@10 values. CACFP is much

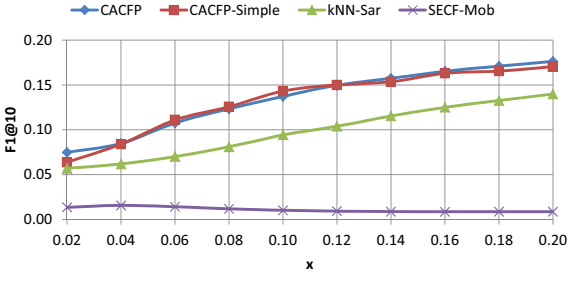


Fig. 2. Recommendation accuracy for method kNN-Sar, SECF-Mob and CACFP; F1@10 results obtained for the methods generating recommendations in the 4-dimensional space at x values lower than 0.2

more accurate (in terms of F1) than the methods proposed in [20] and [13]. As we have confirmed experimentally, CACFP may be used to achieve recommendation accuracy comparable to that achieved by other methods even when using two times less data (when sparse data sets are considered). For values of x higher than 0.4, the accuracy of the method using only non-collaborative data for the preliminary feature combination – i.e., the simplified variant of CACFP (referred to as CACFP-Simple) – is lower than the accuracy of the full version of CACFP. Nevertheless, for x lower than 0.4, the results of recommendation accuracy of both CACFP methods are comparable, as shown on Fig. 3.

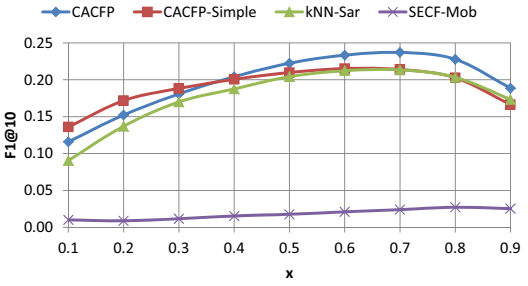


Fig. 3. Recommendation accuracy for method kNN-Sar, SECF-Mob and CACFP; F1@10 results obtained for the methods generating recommendations in a 14-dimensional space at x values higher than 0.1

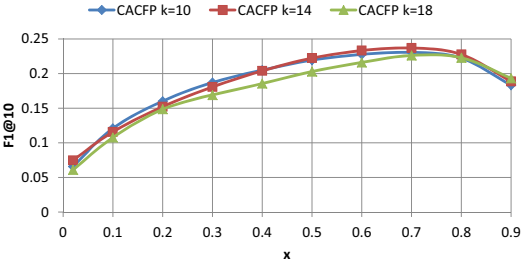


Fig. 4. Recommendation accuracy for CACFP; each curve representing F1@10 results obtained for a specified value of k and several values of x

We investigated the way the parameter k of the introduced methods is correlated with the recommendation accuracy. The results that we obtained for CACFP are presented in Fig. 4. We observe relatively high independence of recommendation accuracy from k. Nevertheless, it is still noticeable that for smaller training ratios, smaller values of k should be chosen.

6 Conclusions

When compared to the widely-referenced recommendation method purely based on collaborative filtering that has been proposed in [20] or to the collaborative semantically-enhanced collaborative filtering technique proposed in [13], our approach to hybrid recommendation proves to cope effectively with behavioral data sparsity. The advantage of our method is particularly visible when it comes to the most challenging range of x values lower than 0.1. In this case, our method achieves the same recommendation accuracy for a train set twice smaller than that achieved by the widely-referenced recommendation methods proposed in [20] and [13].

Based on the results of experimental comparisons, we are able to state that a low-dimensional space obtained as a result of the augmentation of content features driven by behavioral data is especially suitable for recommendation

generation in cases when collaborative data sparsity disqualifies the use of the widely-referenced methods presented in [20], [13].

We observe that using collaborative augmentation of the first-stage user vectors (normal user vectors prepared for the feature augmentation stage) in a low-dimensional space of the most principal content feature components; gives the best results (in terms of recommendation accuracy) when very sparse data sets are considered. Moreover, one may notice that when CACFP is applied, the F1 score does not depend on k so much as in the case of the other methods – the feature that may be useful in real-world applications.

We have shown that collaborative data may be used both as a source of data on items 'popularity', and as a source of additional content features enabling us to more deeply model user interdependencies. Such an approach is fairly new when compared to hybrid recommendation methods presented in the relevant architecture – one may state that what we propose is to 'hybridize two hybrid recommendation approaches': feature combination and feature augmentation [5]. Thanks to building a unified model for content-based and collaborative characteristics, the system based on CACFP is able to treat differently items that are distinguishable only when both data sources are taken into account.

Acknowledgments. This work is supported by the Polish Ministry of Science and Higher Education, grant N N516 196737.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In: Fifteenth National Conference on Artificial Intelligence, pp. 714–720 (1998)
3. Berry, M., Dumais, S., O'Brien, G.: Using linear algebra for intelligent information retrieval. SIAM Rev. 37, 573–595 (1995)
4. Burke, R.: Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007)
5. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
6. Campos, L., Fernandez-Luna, J., Huete, J., Rueda-Morales, M.: Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. Int. J. Approx. Reasoning 51, 785–799 (2010)
7. Claypool, M., Gokhale, A., Miranda, T.: Combining content-based and collaborative filters in an online newspaper. In: ACM SIGIR Workshop on Recommender Systems-Implementation and Evaluation (1999)
8. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys 2010), New York, NY, USA, pp. 39–46 (2010)

9. Gunawardana, A., Meek, C.: A unified approach to building hybrid recommender systems. In: RecSys 2009: Proceedings of the third ACM Conference on Recommender Systems, pp. 117–124 (2009)
10. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Trans. Information Systems 22(1), 5–53 (2004)
11. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
12. Melville, P., Mooney, R., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: Dechter, R., Kearns, M., Sutton, R. (eds.) Eighteenth National Conference on Artificial Intelligence, Menlo Park, CA, USA. American Association for Artificial Intelligence, pp. 187–192 (2002)
13. Mobasher, B., Jin, X., Zhou, Y.: Semantically enhanced collaborative filtering on the web. In: Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G. (eds.) EWMF 2003. LNCS (LNAI), vol. 3209, pp. 57–76. Springer, Heidelberg (2004)
14. Papagelis, M., Plexousakis, D., Kutsuras, T.: Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 224–239. Springer, Heidelberg (2005)
15. Peralta, V.: Extraction and Integration of MovieLens and IMDb Data. Laboratoire PRiSM, Technical Report. Universit de Versailles, Francia (2007)
16. Pilaszy, I., Tikk, D.: Recommending new movies: even a few ratings are more valuable than metadata. In: Proceedings of the Third ACM Conference on Recommender Systems (RecSys 2009), New York, NY, USA, pp. 93–100 (2009)
17. Rashid, M., Lam, S., Karypis, G., Riedl, J.: ClustKNN: A Highly Scalable Hybrid Model&MemoryBased CF Algorithm. In: Computer Science and Engineering, University of Minnesota, Minneapolis (2006)
18. Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworth-Heinemann, Newton (1979)
19. Salter, J., Antonopoulos, N.: CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. IEEE Intelligent Systems 21, 35–41 (2006)
20. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Application of dimensionality reduction in recommender system - a case study. In: ACM WebKDD 2000 Web Mining for E-Commerce Workshop, Boston, MA, USA (2000)
21. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce (EC 2000), pp. 158–167. ACM, New York (2000)
22. Schafer, J.B., Konstan, J.A., Riedl, J.: E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery 5(1/2), 115–153 (2001)
23. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, pp. 253–260 (2002)
24. 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)
25. Symeonidis, P., Nanopoulos, A., Papadopoulos, A.N., Manolopoulos, Y.: Collaborative recommender systems: Combining efectiveness and eficiency. Expert Syst. Appl. 34, 2995–3013 (2008)

Statement Networks Development Environment *REx*

Wojciech Cholewa, Tomasz Rogala, Pawel Chrzanowski,
and Marcin Amarowicz

Department of Fundamentals of Machinery Design, Silesian University of Technology,
Konarskiego 18, 44-100 Gliwice, Poland
{wojciech.cholewa,tomasz.rogala,pawel.chrzanowski,
marcin.amarowicz}@polsl.pl
http://www.kpkm.polsl.pl

Abstract. The paper describes the development environment *REx*, being a collection of tools used in the construction and development of multilayer statement networks applied in diagnostic expert system shell. Described is the process of inference and its implementation based on a multilayer network of statements that make it possible to include the construction of a knowledge base by a group of experts, a hierarchical organization of knowledge, possibility of taking into account knowledge from many sources.

Keywords: Multimodal Statement Networks, Knowledge Representation, Knowledge-Based Systems, Probabilistic Reasoning, Uncertain Reasoning.

1 Introduction

Inference is the process of formulating conclusions on the basis of available information. This task, most often done by humans, can be implemented by a computer system. This is done by a number of solutions to carry out the process of inference in an algorithmic manner and / or based on the use of knowledge. Among the latter, particularly important are systems that enable decoupling the process of inference from the considered domain. This requires to develop interference systems that use independent knowledge bases.

Inference systems are currently being intensively developed in many fields of science. An example might be their use for the purposes of technical diagnostics, with the primary task identifying the technical state of the object based upon all available information. This task is particularly difficult for objects which are subject to slow-changing processes of wear.

The increase in complexity of technical artefacts for both their structure and dynamic processes occurring in them contributed to the excessive number of available data on the object. Since these data are at varying degree covariate to changes of state, defining knowledge about relations between the observed outputs of the object, and the technical state is difficult. One example are turbine

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 30–39, 2011.

generators, for which the description of the possible definitions of the symptoms can be very complex and involve a variety of mechanical, magnetic and aerodynamic interactions. To the difficulties of relationships also comes uncertainty of the data which is caused by measurement errors, random factors, and sometimes the approximate nature of the data. Hence the need to develop inference systems allowing the identification of the state under conditions of uncertainty.

The difficulty of articulating the knowledge defining the symptoms for complex objects as well as the development of methods of numerical modeling of multidimensional data, contributed to the development of diagnostic methods using models of objects. One example is the so-called model-based diagnosis, which is based on analyzing the differences between the results of observation of the object and the results of the model tuned to this object and allows simulating the operation of the object with the given condition [2].

In the case of diagnostics subject to slow-changing process of wear the most common approach is to use object models as simulators to allow for the acquisition of additional data about the object. Performing active diagnostic experiments, conducting a simulation experiment (using the object model), with desired technical states and observing environmental interaction allows acquisition of data corresponding to the states, and thus understanding the relationship between states and the corresponding symptoms. Data obtained in this way represent the implicit knowledge about the object. This knowledge can still be uncertain. This is due to the fact that highly tuned models of object are also inaccurate models (e.g., do not take into account the unknown and difficult to assess affects of environment on the object). A convenient way to represent implicit knowledge, derived on the basis of a simulation experiment or diagnostic tests are diagnostic models. These models have been developed mostly by machine learning, and show mapping object between model interaction with the environment and the state of the object, and can be applied in model-based inference systems.

Effectiveness of the reasoning process depends on the availability and quality of knowledge about the object. This leads to the need to simultaneously take into account the sources of explicit knowledge expressed by experts as well as implicit knowledge [2]. The integration of knowledge for model based inference systems is a difficult task requiring the development of complex models taking into account the knowledge derived from many experts and many data sources. This task can be simplified by considering a number of simple models with lesser degrees of complexity representing the selected sources of knowledge. The main disadvantage of this approach is the need for the synthesis of responses from different types of incompatible individual models, showing conflicting data. Examples are diagnostic models created by experts in expressing different opinions on the considered domain.

The issue of recognition of the technical state of complex objects is a difficult task that requires consideration of incomplete, inaccurate and approximate data. Moreover, inferences about the state are usually performed in conditions of imprecise, incomplete and partially contradictory knowledge. This leads to

the need for the development and application of tools enabling to take into account the nature of the inference process. One of the convenient solutions for conducting inference under uncertainty and the possibility to include knowledge from many sources are multilayer statement networks.

2 Multilayer Statement Networks

2.1 Graphical Models

Multilayer statement networks belong to the class of graphical models which use representations of the domain description in the form of graphs. Individual variables are represented by graph vertices, and the dependencies between them by graph edges. Network models are models most often connecting two fields of science, graph theory and the area under which it is possible to apply the inference process. Examples are probabilistic graphical models using probability theory, where all variables are considered as random variables.

The task of inference in the network models boils down to searching for an equilibrium network by changing values of specific variables [1]. For example, the search for equilibrium in a probabilistic graphical models lies in the fact that when you change the specified values of random variables, resulting for example from the observation of the object, the answer is obtained on the probability distribution of the variables that describe such state. This action corresponds to the inference carried out using a diagnostic model.

The basic advantage of network models is their transparency. Representation of dependencies between variables allows easy understanding of the described domain, and therefore its editions. This means that graphical models can be identified and tuned by hand, as well as automated using a set of machine learning algorithms allowing identification of the structure and / or network parameters based on data obtained during the numerical experiment. These models, as opposed to numerical models of black boxes, do not require re-learning when connecting new examples to the set of learning data.

The process of inference in network models as opposed to inference based on numerical models of black boxes is not necessarily unidirectional. It is possible to both place known values for corresponding variable states and the launching of a process requesting the reading of values of corresponding variable symptoms.

Thanks to that it is also possible to indicate, among other things, the most likely symptoms, which should appear in the event of a particular technical state. This is a particularly desirable property in expert systems requiring explanations of the inference process being carried out.

2.2 Statements and Statement Networks

Graphical models such as probabilistic graphical models can be used as inference modules of expert systems [3]. Their application requires the development of an appropriate way to communicate with users of these systems. It is necessary

to pay attention to the possible use of statements occurring for example in the form of logical sentences. To obtain an unambiguous interpretation of statements, taking into account the considered context, may be accompanied by adequate explanation of their interpretation.

The use of statements is particularly convenient for systems intended to support technical diagnostics, where there is a clear necessity to interpret the data value (e.g., feature value of the measuring signals) and their changes [1] [4]. The statement is information on the recognition of expression resulting from observed facts or representing an opinion. Usually it is described by a pair [1]

$$s =< c, v > \tag{1}$$

where c is the statement content, and v is the statement value. The content can be a declarative sentence, to which is attributed one of the logical values (true, false). Use of statements in expert systems allows the introduction of a complex system of aid, containing an explanation of used terms, links to sources and various comments. It is assumed that in inference systems statement content is constant and statement value can vary. This assumption allows the creation of thesauri, or sets of constant statement content. Thesauri help manage the explanations of statements content and control the degree of detail of explanations.

Graphical models in which statements are represented by variables are called statement networks [1]. For the purpose of constructing a convenient statement network, it is possible to consider simple statements containing only one variant of content with its value or complex statements, which are specified as the pair:

$$s =< \underline{\mathbf{c}}, \underline{\mathbf{v}} >=< \underline{c}_{1:n}, \underline{v}_{1:n} > \tag{2}$$

where $\underline{\mathbf{c}}$ is n the n-element vector of the statement contents variants, and $\underline{\mathbf{v}}$ is the corresponding n-element vector of values for different variants of content. In the case of statement network it is assumed that variants of complex statements constitute an exhaustive set of mutually excluding elements. An example of complex statement content is the triple vector $\underline{c}_{1:3}$, where [1]:

$$\begin{aligned} c_1 &= \text{apple is red;} \\ c_2 &= \text{apple is green;} \\ c_3 &= \text{apple has a different color than red or green.} \end{aligned} \tag{3}$$

Methods of inference in statement networks are dependent upon the accepted definitions of statement values. Approximated statements are introduced in order to enable inference in imprecise, incomplete, and even contradictory information environment. It is assumed that the contents of these statements are accurate and constant; however, their values are approximate. Approximate values of statements can be defined as degrees of truth or degrees of belief in the truth of statements. It is possible to consider values of approximate statements as point values and interval values.

2.3 Approximate Statement Networks

Approximate statements, which values are expressed as degrees of belief in the truth of statement, are used in statement networks represented as belief networks [4], [5]. Their use requires the definition of conditional probabilities tables for all nodes. The values of these tables are hard to determine. Another variant of the statement network, which allows approximate inference are approximate statement networks in which knowledge is represented as necessary and sufficient conditions [1].

If the belief in the truthfulness of the statement s_p is always accompanied by the belief in the truthfulness of the statement s_n, but not necessarily inversely, then s_p is sufficient condition to establish s_n. At the same time s_n is necessary condition for s_p. Information about the fact that s_p as a sufficient condition for s_n can be written as [1]

$$b(s_p) \leq b(s_n) \tag{4}$$

where, $b(\cdot)$ is a statement value. Approximate necessary and sufficient conditions are defined as presupposing that the dependence (4) can be satisfied with small permissible deviation δ. The value of an acceptable degree of approximation can be taken in different ways, for example as:

- value δ equal for all considered conditions

$$b(s_p) - \delta \leq b(s_n) \quad \delta \geq 0 \tag{5}$$

- values δ_p i δ_n assigned individually for each statement s_p and s_n

$$b(s_p) - \delta_p \leq b(s_n) + \delta_n \quad \delta_p \geq 0 \quad \delta_n \geq 0 \tag{6}$$

- values $\delta_{p,n}$ assigned individually to each condition for a pair of statements s_p and s_n

$$b(s_p) - \delta_{p,n} \leq b(s_n) \quad \delta_{p,n} \geq 0 \tag{7}$$

The use of deviation δ written as in (5) (6) (7) allows the application of varying degrees of importance of statements. Systems containing inequalities (5) (6) (7) can be considered as statement networks, where statements are represented as nodes and each inequality is represented by the corresponding edge of the network. A statement network solution based on the use of statements of necessary and sufficient conditions when there is no contradiction in the network does not require the use of complex algorithms for searching for network solutions. In the case of a solving network, in which there is a contradiction, to seek solutions for the network could rely on finding the equilibrium minimizing the value of the criterion function written as [1]:

$$e = \sum_{p,n} k_{p,n}^2 \delta_{p,n}^2 \tag{8}$$

where $k_{p,n}$ is a parameter determining the degree of importance of the condition representing influence of statement s_p, on statement s_n appearing in the approximate network.

2.4 Multilayer Statement Networks

It is possible to develop diagnostic models in the form of complex statement networks for complex technical objects. This task is difficult, however, requires consideration of various aspects of the operation of the object, as well as many of its components. A much more convenient way is to decompose the considered domain into subdomains, which can be described by simpler component models. It should be noted that the decomposition does not have to be related to the spatial structure of the object but for example to a selected aspect of its operation. It allows to create component models of a limited number of inputs (symptoms) and outputs (classes of states), which allows easier interpretation, identification and subsequent tuning [2].

A convenient tool for building component models and their subsequent synthesis, are multi-layered statement networks [1]. The structure of these networks is defined in a similar way as multimodal networks, where individual layers are treated as modes of multimodal network. Multimodal statement network is defined as a directed hypergraph, described by ordered triple $< V, E, \Gamma >$, where V is the set of all vertices of the hypergraph, E is a set of hypergraph edges and Γ is a set of mods of the hypergraph [7],[6]. The essence of the system is that individual modes of the hypergraph represent component networks for individual layers of the model and the selected nodes of the networks can occur in many layers. An example of a multilayer network is shown in Figure 1.

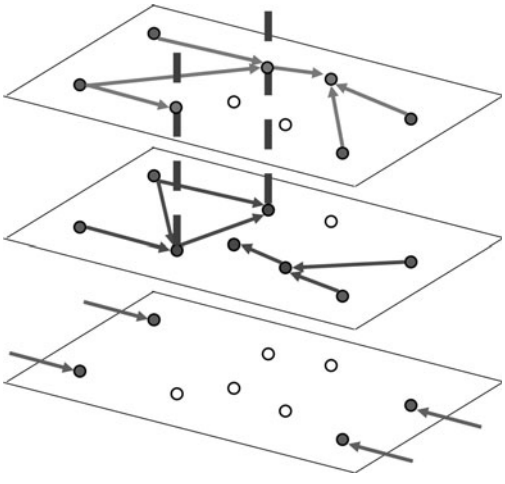


Fig. 1. An example of multilayer statement network

Synthesis of multilayered network response can be implemented in two ways::

– as a process of aggregation, where individual nodes act as independent instances in subsequent layers, and their values are determined in an independent manner, for example, when all the network layers were developed as statement networks, represented by statements of beliefs, or

– as the reconciliation process, where each individual node acts as a single copy in all layers. This would require an appropriate adaptation of procedures for solution of each layer of the network. An example might be using approximate statement networks in all layers of a multilayered network.

A characteristic feature of multilayered networks is that the individual networks created within the considered layer can be statement networks represented by different types of network models. The possibility of constructing a statement network as a set of component networks opens the possibility of collective creating of individual network layers, independently by several experts. Depending on the used decomposition of the considered domain, such networks can be built as networks of diverse structure, such as multiaspected or multiscaled networks.

3 Development Environment *REx*

Inference systems based on the use of multi-layer statement networks are a promising tool to enhance the process of identifying the technical state of complex objects. They take into account the characteristics and needs of diagnostic inference, among other things:

– imprecise, incomplete and partially contradictory data and knowledge,
– the possibility of taking into account knowledge from many sources,
– no need to re-identify the model in the case of new examples in a set of training data,
– possibility of backward inference, in order to carry out the process explanation for the conclusion reached,
– simple and understandable way to communicate with users of the system using statements, the possibility of independent components to build models based on different sources,
– greater transparency of knowledge represented by a multilayer statement network, easier interpretation,
– identification and tuning of the component models and achieving the better generalization properties for these models,
– the use of different decomposition of domains for the subdomains connected with various aspects of an object, or associated with the spatial decomposition of the object.

Building expert systems to aid the diagnosing of complex objects requires to use special exchange of information between different modules of such systems. The application of the general concept of a *blackboard* is here specially reasonable [4]. The blackboard (Bulletin Board component) is a place where messages containing information about values of statements are available to receivers. The statements which appears on the table are active. It means that changes of statement values included in a model A can initiate sequences of operation which cause automatic changes of the values of other statements in models B,C ect. The table contains also information about the modules which are responsible for

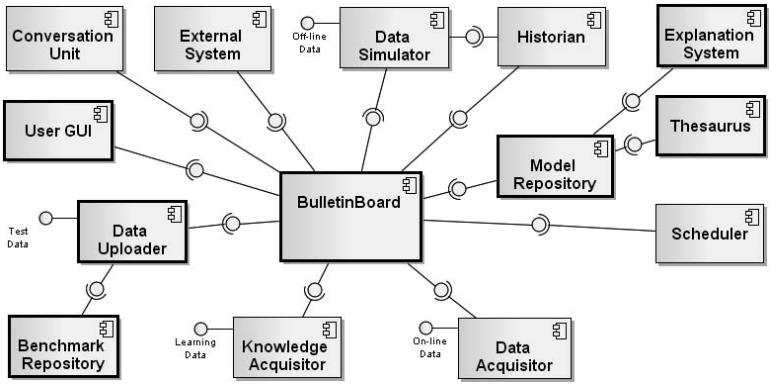


Fig. 2. Components diagram of development environment *REx*. The current release of *REx* (http://www.kpkm.polsl.pl in section *Projects*) includes the components marked in bold frames.

the change of statements values. An interesting remark is that the blackboard can be understood as a hierarchically ordered database, designed for storing solutions generating by autonomous modules.

The structure of the environment *REx* in the form of components diagram is presented in figure 2. It is an experimental environment in which it is possible to conduct appropriate comparative studies aimed at determining the most effective, from the viewpoint of the effectiveness of the process, inference tools. These include among others the research of:

- aggregation processes in the application of many different types of networks which constitute the multilayered network. Obtained in this way results should help identify appropriate methods for determining the response of the inference system,
- algorithms for solving the statement networks, including in particular the approximate statement networks,
- algorithms for identifying the structure and network parameters on the basis of training data,
- other aspects of the testing of interoperability of network layers, depending on the structure of multilayer networks, for example, multilayer networks represented as a multiscale network.

The proper conduct of comparative research requires the development of a numerous sets of training and testing data and sets of examples of different networks, that can be used repeatedly during the mentioned studies and evaluating the performance of the inference system. For this purpose environment *REx* includes a repository of data and models (Benchmark and Model Repository) with differential network structures and differential training and testing data.

Building complex models represented by component models that are created based on various sources of explicit and implicit knowledge required to ensure

that the content of statements entered into the system is unequivocally interpreted by all individuals developing particular component models. The need to ensure the proper integrity of the developed set of statements is associated with the respective statement descriptions and avoiding the introduction of repeated, or semantically similar, contents of statements. The *REx* system includes the possibility of building thesauri for this purpose, creating appropriately classified dictionaries of accessible contents of statements, which can be built by many people working on a dictionary for the considered domain. Since statement collections can be numerous and the possibility of verification of the thesaurus in terms of formal and semantic correctness may be time consuming, appropriate methods of managing the various versions of the thesaurus are considered.

The overall process of building a multilayer network involves the following steps:

- introduce a set of possible statements for the considered domain through the creation of a new thesaurus. Defining the thesaurus involves building simple statements by indicating the type of values of the statement and the introduction of its content and the necessary guidance on the interpretation of the statement. In the next step, based on the set of simple statements it is possible to create complex statements in a similar manner.
- define a new model of multilayer network and determine the number of layers,
- define the individual layers of the network by developing component networks. This task, for each layer, requires the development of the structure of the graph, indicating the selected vertices representing particular statements and the edges between the vertices of the graph. The next step defines parameters for the network model if required for the considered type of network model used in the particular layer. An example might be the necessity to define a table of conditional probabilities for the layer described by the statement network represented as a belief network,
- select merging strategy for layers (i.e. aggregate or reconciliate layers),
- start-up of calculations.

The environment *REx* was developed in *R* language. The use of this language gives, among other things, the ability to develop the environment and its components using free software.

4 Summary

Contemporary development of artificial intelligence allows the conduct of the process of inference with inaccurate, incomplete and even contradictory information. In the case of inference systems based on knowledge, which can also be inaccurate, incomplete and partially contradictory, inference is difficult. One example is to conduct inference on the technical state for a complex object. The performance of an inference system shell in this case depends on whether the requesting system allows both the consideration of the nature of the inference process, as well as, the quality and possible use for this purpose of various available sources of knowledge.

The paper describes a general concept of the development environment *REx* which is a set of tools used for the construction and development of multilayer networks used for diagnostic expert system shells. Described is the process of approximate inference realized by multilayered statement networks, which among others, gives opportunity to take into account knowledge from multiple sources. The paper describes the basic properties of statement networks and the ability to use approximate statement networks as component to build multi-layer statement networks. We report the construction of an experimental environment enabling *REx* to carry out comparative studies on both the testing of different algorithms and evaluation of the efficiency of the processes of inference. The developed version of the environment *REx* is available for download in the section *Projects* on the website of Department of Fundamentals of Machinery Design, at `http://www.kpkm.polsl.pl`

Acknowledgements. Described herein are selected results of study, supported partly from scientific funds, as statutory research and research projects 4T07B05028 entitled„ Reverse models and diagnostic modeling", N504478434 entitled„ Construction of diagnostic advisory systems using multiscaled statement networks" and from the budget of Research Task No. 4 entitled„ Development of integrated technologies to manufacture fuels and energy from biomass, agricultural waste and others" implemented under The National Centre for Research and Development (NCBiR) in Poland and ENERGA SA strategic program of scientific research and development entitled „Advanced technologies of generating energy."

References

1. Cholewa, W.: Multimodal statements networks for diagnostic applications. In: Sas, P., Bergen, B. (eds.) Proceedings of the International Conference on Noise and Virbation Engineering ISMA 2010, September 20-22, pp. 817–830. Katholique Universiteit Lueven, Lueven (2010)
2. Cholewa, W., Korbicz, J., Kościelny, J., Patan, K., Rogala, T., Syfert, M., Witczak, M.: Diagnostic methods. In: Korbicz, J., Kościelny, J.M. (eds.) Modeling, Diagnostics and Process Control-Implementation in DiaSter System, pp. 153–231. Springer, Heidelberg (2010)
3. Dawid, F., Cowell, R., Lauritzen, S., Spiegelhalter, D.: Probabilistic Networks and Expert Systems. Springer, New York (2007)
4. Korbicz, J., Kościelny, J.M., Kowalczuk, Z., Cholewa, W. (eds.): Fault diagnosis: Models, Artificial Intelligence, Applications. Springer, Heidelberg (2004)
5. Jensen, V.J.: Bayesian networks and decision graph. Springer, New York (2002)
6. Heath, L., Sioson, A.A.: Multimodal networks: Structure and operations. IEEE//ACM Trans. on Computational Biology and Bioinformatics 6(2), 271–280 (2009)
7. Sioson, A.: Multimodal networks in biology. Phd Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (2005)

Domain Based Semantic Compression for Automatic Text Comprehension Augmentation and Recommendation

Dariusz Ceglarek[1], Konstanty Haniewicz[2], and Wojciech Rutkowski[3]

[1] Poznan School of Banking, Poland
dariusz.ceglarek@wsb.poznan.pl
[2] Poznan University of Economics, Poland
konstanty.haniewicz@ue.poznan.pl
[3] CIBER, Poland
w.rutkowski@astronet.pl

Abstract. This works presents an application of semantic compression where domain frequency dictionaries are used to augment comprehension of documents. This is achieved by incorporating user's feedback into proposed solution. Experiments and examples of actual output are given. Moreover, a measure that allows for evaluation of changes in a structure of available groups is defined and presented.

Keywords: semantic compression, recommendation system, comprehension adjustment, semantic network, WiSENet, SeNeCaNet, natural language processing.

1 Introduction

The aim of this work is to present the newest artefacts that are a result of ongoing research on semantic compression. Semantic compression itself is a result of active research in domain of intellectual property protection that began with publication on Semantically Enhanced Intellectual Property Protection System (SEIPro2S [1]). Throughout research activities, authors developed various enhancements to semantic compression along with important tools and algorithms that find application in derived solutions. This work discusses merits of domain based semantic compression, describes differences between it and its general version and introduces research results along with envisioned application employing them. This application leverages existing technology coupled with feedback from users in order to build a system that is able to recommend text documents suiting user's proficiency in any of supported domains.

2 General Semantic Compression

An idea of semantic compression has been introduced by the authors in 2010 [2] as a method of improving text document matching techniques both in terms of

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 40–49, 2011.

effectiveness and efficiency. Compression of text is achieved by employing a semantic network and data on word frequencies (in form of frequency dictionary). The least frequent terms are treated as unnecessary and they are replaced with more general terms (their hypernyms stored in semantic network). As a result, a reduced number of words can be used to represent a text document without significant information loss, which is important from a perspective of processing resources (especially when one would like to apply a vector space model [12],[10]).

Furthermore, the reduction of words' number helps in dealing with linguistic phenomena, which are problematic in natural language processing and information retrieval [9]. Most commonly referenced phenomena of this type are polysemy and synonymy. When multiple terms used to describe the same or very similar concept occur relatively rarely, they can be replaced by one common, more general, concept. Due to employment of statistical analysis in domain context, remarked frequency dictionaries are prepared and let system deal with polysemic words with less effort and lower error rate than solutions not employing such technique.

As remarked, one cannot allow for significant loss of information which is a result of replacing more specific concepts with more general ones. To exemplify, let us consider that a document is concerning some biological study. A number of species is mentioned in Latin. When one is to categorize text in an automatic manner, these Latin terms extend vector describing document, thus they complicate the whole process. One can have an observation that every Latin name of some particular fish can be replaced by a fish concept. In result, the entire task is carried out with less resources with no significant information loss. Of course, this can only be applied for a specific corpus of documents where these Latin names are rare, thus omissible. The choice of concepts for generalization is domain dependent.

In general, the semantic compression enables information retrieval tasks, such as text matching, to operate on a concept level, rather than on level of individual terms. This can be achieved not only by gathering terms around their common meanings (known from synset based approach), but also replacing longer phrases with their more compact forms.

The emphasized concept level allows for capturing of common meaning expressed with different set of terms.

Although this is a lossy compression, the loss of information is minimal by selecting the least frequent words and replacing them by more general terms, so their meaning remain as similar to the original as possible. The compression ratio can be tuned easily, by setting a number of concepts to be used to describe text documents. Experiments, that were conducted to measure quality of the method in information retrieval tasks showed, that the number of words can be reduced to about 4,000 without significant deterioration of classification results [3]. To authors best knowledge similar solutions are not available, yet the whole research is inspired by works such as [5], [6], [7] and [8].

3 Domain Based Semantic Compression

Semantic compression combines data from two sources: term frequencies from frequency dictionary, and concept hierarchy from semantic network. Usually, one extensive semantic network is used for a given language (e.g. WiSENet for English based on WordNet [4], SeNeCaNet for Polish) and thus it is able to include linguistic knowledge covering multiple domains.

The varying characteristics basing on term frequency in domain has a crucial impact on functioning of semantic compression's algorithm, as term frequencies are decisive factor in identifying the least frequent terms to be replaced by their hypernyms in given domain. To give an illustrative example, consider a document originating from nature studies where a life of common rodents is discussed. On the other hand, let us consider document from Information Technology focused on Human Computer Interfaces. Both documents have passages where a term mouse is to be found. Domain based semantic compression allow for generalising mouse into two more general concepts, different for every domain. In nature studies one may generalise mouse as a rodent and dealing with it in Information Technology one would use pointing or electronic device. General semantic compression with low number of output concepts, thus high level of compression, would choose hypernym according to overall frequency what might introduce unnecessary chaos.

Further experiments conducted by authors confirm (described in greater detail in following section), that natural language processing tasks employing domain based semantic compression yield better results than using its general form. As exemplified, this is because domain frequency dictionaries better reflect language characteristics.

To summarize, semantic compression is more effective, when text domain is identified, and an appropriate domain frequency dictionary is used to perform the process.

4 Results of Experiments on Domain Based Semantic Compression

Domain based semantic compression is easily implementable with already crafted artefacts as long as a certain procedure is applied to processed text. When domain is established for a text fragment, one can apply a specific domain frequency dictionary to perform local generalisation, avoiding extreme cases. By extreme, one is to understand concepts that are too general in context of given domain. This differentiates domain based semantic compression from general one, as the latter maximizes savings in terms of possibly shortest length of vector that represents processed documents. Yet, in envisioned application, avoiding introduction of unfamiliar (from domain point of view) concepts is an important perk that is readily exploited for the benefit of interested users.

Proposed procedure was tested on a group of users. This was done to measure whether it increased level of text comprehension. One cannot measure this

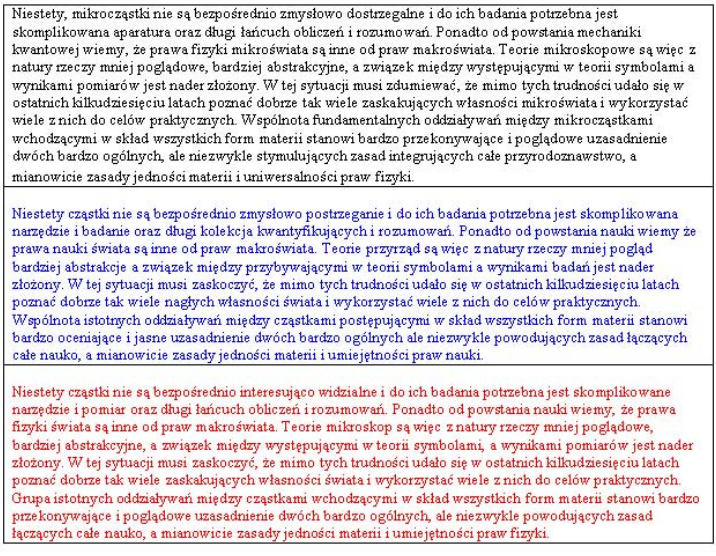


Fig. 1. An example of domain based semantic compression for Polish as seen by first experiment participants

without human user, due to the elusive nature of evaluation of explored problem. As an outcome, authors gathered feedback from users.

To summarize the results, one can state that application of locally adjusted domain frequency dictionary improves readability and allows for exclusion of generalised concepts that do not fit into context.

The performed experiment consisted of a set of four samples presented to participants. There were 32 participants. Participants surveyed had not received any training in domains that occurred in samples presented to them. Every sample was built with 3 text fragments. First fragment was an unmodified text fragment taken from corpus at random. The only constraint that every fragment had to follow was that its domain should be as unambiguous as possible. Chosen fragment was then transformed by application of first domain based semantic compression and second general semantic compression. Participant had to make a choice of whether he preferred better first transformation or the second one. He had to make a choice three more times and at the end share his opinions on his decisions and motivation for them. Sample is given in figure 1. Please notice that an initial experiment was in Polish.

Whole experiment was conducted using SeNeCaNet and extensions derived from project Morfologik [11]. Inclusion of Morfologik dictionary allowed for automatic changes of declination and conjugation of word forms. This is difficult task for languages such as Polish due to large number of forms possible for every word (including gender,person and grammatical aspect). Effort was made to achieve over 95,5 % of correct transformations where every transformation was a two phase process. One has to underline that for an error authors understand

change from a verb or an adjective to a noun. First, a term was identified, then it was checked whether it is candidate for generalization. Second, whether one had a term to be generalised a proper form had to be applied in order to present a user performing evaluation with maximally uncluttered text fragment.

5 Domain Based Semantic Compression for Comprehension Augmentation

Positive results of preliminary experiments let authors present a new application which is based on employment of domain based semantic compression.

Let's envision a scenario where a user is presented with an interdisciplinary text. His understanding of the whole might be dependent on how well is he coping with various fragments of given text. In ideal case one would like to posses a tool that would enable him to filter out fragments that are beyond his comprehension and leave those of interest to him. This can be achieved by modifying already existing means and enhancing their capabilities with domain based semantic compression. Thus, every incomprehensible passage would be transformed in order to match user level.

To begin with, any text that may be a source for further transformation with semantic compression has to be split up into a set of paragraphs. This enables one to categorise every paragraph. The overall number of categories available and as result domain frequency dictionaries is dependent on the number of text corpora that were previously classified and processed.

Suppose that given text consists of fragments that were identified as biology, astronomy and generic ones. For the example's sake, lets assume that user which was presented with text in question, stated that he is an expert in astronomy and he has elementary understanding of biology topics. His background on other domains is irrelevant to this example; one has enough data to proceed with process that shall finish with a body of text that was altered to make it more comprehensive for the given user. As every paragraph was tagged with a category, it can be processed so that its content is adjusted to level indicated by user in the first place. With exemplary setup, this means that all paragraphs classified as belonging to astronomy will not be altered at all. When those tagged as biology are encountered, modifications are made. Algorithm for envisioned scenario uses SeNeCaNet and WiSENet as semantic networks. Two semantic networks are used as both Polish and English is supported. Depending on choice of domain, a different domain frequency dictionary is used for text generalisation. As referred in earlier section, when a concept which introduces ambiguity is processed, frequency dictionary provides an answer which parent concept (its hypernym) from semantic network shall be used to replace it and transform text into more general one. Figure 2 depicts the output for English.

Processing whole semantic network by singling out paths from root to furthest child nodes allows computing a mean for length of available paths. This mean is useful as it can be used as a measure to quantify inherently fuzzy terms such as advanced, beginner or intermediate. One could argue that these terms could

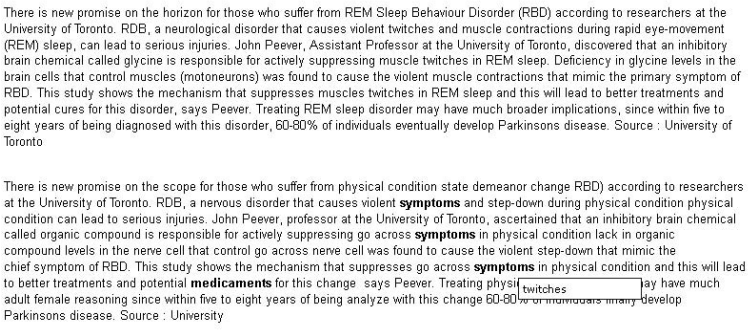


Fig. 2. Sample from application of domain based semantic compresion on English corpus. Please notice that original concept is acessible for the user.

be mapped in a fixed manner as some percentage of computed mean. This can be misleading, as every domain can have various values covering these terms. Of course, one has to provide some initial values to compute first proposition for interested user. However, throughout usage, user can provide feedback on whether concepts proposed by system suit his level. Statistical analysis enables to adjust level represented by introduced labels. This adjustment can be fully bidirectional as text could have been over generalised or under generalised. Ideally, this adjustment process should strive to become a continuous one where user could input his feedback in a friendly manner.

The generalization process can be driven by a method that was used in earlier experiments [2]. It was used with a success in proving positive results of semantic compression when applied to text categorisation and evaluated by clustering. It is based on a number of concepts that are used in semantic compression. A set of not generalised terms is built with every term from text (apart from those from stoplist), and then further sets are computed where each has lesser number of terms compared with previous one. Used sets contain from 2000 concepts up to 8000 concepts. The greater the number of terms in a set, the less generalised texts will be results of semantic compression.

The ability to interact with a system in such a manner that it can respond for requests of greater or lesser generalisation brings one to the key point of domain based semantic compression.

6 Domain Based Semantic Compression for Automatic Document Recommendation

Described earlier functionality led to a system that is capable of gathering of user preferences towards different texts. Their feedback is included so that any text can be made more accessible. What is more, after sufficient number of sessions with given user, the system is able to compute stable pattern for given user. Whether user shall be a member of some group, his computed pattern shall influence his group's pattern.

Computed patterns are valuable resource as they allow analysing documents in terms of their value for a user. When a document that has a structure resembling pre-computed pattern is processed, it is marked as possibly interesting for a group or an individual.

The idea was implemented as a prototype system where 3 preconfigured categories were available to choose from. They were denoted as BEG (beginner), MED (medium) and ADV (for advanced users). Domain semantic compression for MED group was established at 6000 concepts. Level BEG had compression set at 2000 concepts, and ADV group had no compression whatsoever (circa 25600 concepts in total). Texts presented to users were chosen from biology, astrobiology and astronomy domain.

We invited previously surveyed users and some additional ones to take part in experiment. A total of 50 participants joined experiment. Initially, there were 18 users in MED, 15 in ADV group and remaining 17 in BEG group.

Every participant was presented with 20 randomly chosen texts coming from astronomy and biology. Every text was processed according to declared level of participants knowledge in astronomy and biology. Level of astronomy knowledge is high due to the background of additionally invited users. Authors came to a conclusion that changes in comprehension for astronomy are results of individual's knowledge growth. Therefore, no intergroup flow was analyzed. Broader analysis is planned as future developments of author's research.

MED group participants read 360 texts and there were 125 requests of changing level of compression. BEG group participants read 340 texts with 122 requests for change.

Authors enumerated 6 users from MED group with the highest number of requests to change level of compression and additional 5 users from BEG group. These two subgroups were excluded from their initial groups and formed INT (intermediary group). Thus there were 15 users in ADV, 12 in MED group, 11 in INT and 12 in BEG. Detailed summary is given in table 1.

Table 1. Results of the experiment performed with a group of 50 participants. Declared participant level (**Level**), Number of concepts - semantic compression strength (**Strength**), Signalled change to more advanced group (c^+), Signalled change to less advanced group(c^-), Total number of participants in group(**Total**).

Level	**Strength**	c^+	c^-	**Total**
First iteration				
ADV	-	-	18,7 %	15
MED	6000	13,3 %	34,7 %	18
BEG	2000	35,9 %	-	17
Second iteration				
ADV	-	-	18,7 %	15
MED	6000	17,7 %	24,2 %	12
INT	4000	7,3 %	4,5 %	11
BEG	2000	22,1 %	-	12

In addition to reassignment, INT group level of compression was set for 8000 concepts. Further, participants categorized as INT group members were asked to read additional 10 randomly chosen texts. After completion of tasks, there were only 8 requests for levelling up semantic compression (7,3 % request to total text ratio) and 5 requests for levelling it down .

One can notice an interesting development when viewing data as presented in table 2. Stronger compression in text fragments classified as biology related, does not incur a loss of concepts important from the domain's standpoint. Authors believe that it is a result of availability of sizeable biological taxonomy in WiSENet (derived from Wordnet). In addition, one can observe that some of the concepts are generalised, while those with high domain frequency are not generalised at any of semantic compression strength's levels.

This experiment demonstrates notion of stabilizing the system in context of adjusting a number of represented levels of comprehension to user requests. In near future, author's goal is to build algorithm that would enable one to set optimal number of comprehension groups based on users' feedback on semantic compression strength and a number of concepts marked by user as incomprehensible thus candidates for generalization. Envisioned algorithm will enable one to produce an indicator taking into account flow of users among groups calculated by system. As an initial attempt to achieve this goal one could propose a stabilisation indicator that is based on comparison of a number of changes among categories. One has to take into account number of changes to more advanced group and to less advanced group in every iteration. Classes of users belonging to a partition (groups: ADV, MED, INT or BEG) are not disjoint from one another - because of overlapping classification. So, we propose indicator of stability of partition into groups - Q based on [13].

Indicator of quality of classification: $C = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i + \beta c_i^{+} + \gamma c_i^{-})^2}$
Where:

- β - parameter indicating how much compression's strength should be influenced by changes from more compressed to less compressed groups
- γ - parameter indicating how much compression's strength should be influenced by changes from less compressed to more compressed groups
- c^{+} - number of signals indicating that semantic compression strength's level shall be adjusted so that less concepts should be generalized
- c^{-} - number of signals indicating that semantic compression strength's level shall be adjusted so that more concepts should be generalised
- C_i - being a member of i-th group (c_1 =ADV, c_2 =MED, c_3 =BEG, c_4 =INT)

Values of the proposed indicator demonstrate an improvement between first and second iteration ($C_{first} = 0,83$, $C_{second} = 0,945$) with β and γ set to $\frac{1}{2}$ to indicate equal strength of changes among groups in both directions. One can assume improved stability of classification which directly results from less requests for group reassignment.

Envisioned recommendation system can find its applications in various organizations. One of possible applications can be an semi-automatic classification

inside a pool of available specialists where their level is only a declarative one. Session with recommendation system can easily arrange specialists by their fluency in one of domains present in this procedure. Situation in which somebody would like to be perceived as more specialised that he really is beyond scope of this work. Yet, one could approach the problem by extending the procedure with complimentary questions that shall reflect declared level of comprehension and penalize incorrect answers by decreasing user's rank in given domain. Another possible application of this recommendation system are reviewing systems. Instead of tiresome browsing through documents one could undergo a procedure that would allow calibrating his level in various domains represented in to be reviewed text. As a result he would only be presented with only those where his expertise is highest or whatever other criterion was proposed.

Table 2. Overview of experiment's results in terms of: average depth of concepts per compression level (**Avg. depth**), generalization changes compared with compression one level weaker (**Change**) and a total number of domain concepts per compression level (**Domain concepts**). Domains: A- astronomy, B - Biology.

	2000		4000		6000		8000	
	A	B	A	B	A	B	A	B
Avg. depth	5,54	5,53	5,93	6,06	6,19	6,35	6,30	6,41
Change	3,70%	2,70%	4,10%	3,70%	4,50%	3,90%	-	-
Domain concepts	9,67 %	10,82 %	11,04 %	11,40 %	11,92%	12,15 %	12,00%	12,37%

7 Summary

Presented domain based semantic compression can find its application in the variety of domains. First and foremost, the envisioned system which prototype is based on domain based semantic compression is dedicated to knowledge portals that serve content to satisfy a diversified range of user needs. Diversity stems not only from interdisciplinary nature of available documents but also from varied level of knowledge of interested users per every represented domain. Performed experiments were focused on astronomy and biology, yet this is by no means the end of possibilities. Given enough number of documents and interested users one can address any domain.

Applying prepared methods to various documents provides good overall level of comprehension among its users. What is more, this solution can be tailored to specific needs by gathering user feedback and adjusting pre-computed levels to better suit interested group. Authors have presented an indicator that can be applied to measure whether initial user to group assignement is met with real user preferences. This is of importance, as it can allow for further advancements in system usability and flexibility.

Presented application is in early stage of development and many features have to be implemented to reach into a phase of publicly available prototype. Nevertheless, already performed experiments show that results are promissing and encaurage further research.

References

1. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Semantically Enchanced Intellectual Property Protection System - SEIPro2S. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 449–459. Springer, Heidelberg (2009)
2. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Semantic compression for specialised Information Retrieval systems. In: Nguyen, N.T., Katarzyniak, R., Chen, S.-M. (eds.) Advances in Intelligent Information and Database Systems. SCI, vol. 283, pp. 111–121. Springer, Heidelberg (2010)
3. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Quality of semantic compression in classification. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS, vol. 6421, pp. 162–171. Springer, Heidelberg (2010)
4. Fellbaum, C.: WordNet - An Electronic Lexical Database. The MIT Press, USA (1998) ISBN: 978-0-262-06197-1
5. Percova, N.N.: On the types of semantic compression of text. In: Proc. of the 9th Conference on Computational Linguistics (COLING 1982), vol. 2, pp. 229–231. Academia Praha, Czechoslovakia (1982)
6. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. ACM Transactions on Information Systems 10(2), 115–141 (1992)
7. Mihalcea, R., Csomai, A.: Senselearner: Word sense disambiguation for all words in unrestricted text. In: Proc. of the 43rd ACL, pp. 53–56 (2005)
8. Resnik, P.: Using information content to evaluate semantic similarity. In: Proc. of the 14th IJCAI, Canada, pp. 448–453 (1995)
9. Sinha, R., Mihalcea, R.: Unsupervised graphbased word sense disambiguation using measures of word semantic similarity. In: Proc. of the IEEE International Conference on Semantic Computing (2007)
10. Erk, K., Pado, S.: A structured vector space model for word meaning in context. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 897–906. Association for Computational Linguistics, Stroudsburg (2008)
11. Milkowski, M.: Automated Building of Error Corpora of Polish. In: Lewandowska-Tomaszczyk, B. (ed.) Corpus Linguistics, Computer Tools, and Applications - State of the Art. PALC 2007, Peter Lang. Internationaler Verlag der Wissenschaften, pp. 631–639 (2008)
12. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
13. Gordon, A.D.: Classification, 2nd edn. Chapman and Hall/CRC, USA (1999) ISBN: 978-1-58488-013-4

Model of Community-Build System for Knowledge Development

Przemysław Różewski

West Pomeranian University of Technology in Szczecin,
Faculty of Computer Science and Information Systems,
ul. Żołnierska 49, 71-210 Szczecin, Poland
prozewski@wi.zut.edu.pl

Abstract. The community-build system (CBS) is a system for content creation by a community operating on a dedicated engine (e.g. Wiki). The CBS demonstrated how collaborative efforts can be a powerful feature to build a massive knowledge repositories. In the paper the model of community-build system is presented. The model is focused on the knowledge-base content creation process. The editor, creator and the repository are main model's actors. Moreover, the proposed CBS model allows to calculate the performance measurements like average number of editors/creators, average on-line editors/creators time. Based on the Markovian Chain theory the dynamic aspect of repository development is analyzed as well. In addition, the presented model is interpreted in the e-learning context.

Keywords: community-build system, cooperation, knowledge development.

1 Introduction

The community-build system is a system for content creation by a community operating on a dedicated engine (e.g. Wiki) [4]. The community-build system demonstrated how collaborative efforts can be a powerful feature to build a massive knowledge repositories [6,12]. The community-build system is an important research issue in the Internet context because of the community-build system becomes one of the main ways to create high-quality content.

So far the community-build system has been analyzed from the point of view of efficiency and technical infrastructure. The storage and community-built databases are one of the research issues [13]. The community-built databases usually keep and process large amount of data (i.e. flicker, youtube). In addition, the community-build system provides opportunities for collaborative knowledge building [17]. Another important research area is a social aspect related to community knowledge creation process. The social aspects of community-built system are discussed in [4,11,18]. The community-built system supports community-oriented authoring, in order to rapidly and collaboratively build the content by editing the page online in a web browser [4]. Moreover the community-build system also represents a class of asynchronous

P. Jędrzejowicz et al.(Eds.): ICCCI 2011, Part II, LNCS 6923 , pp. 50–59, 2011.

communication tools. Another important research issue is a difference between community-build system and other content-related tools (such as blogs or threaded discussions) [22]. So far the community-build system model for knowledge-base content creation process has been missed in the literature.

In the paper the model of community-build system is proposed. The model recognized the creator and editor roles in knowledge development process. The knowledge is modeled using the ontological approach [25], which allows further computer processing. The proposed community-build system model can be used to calculate the performance measurements, like average number of editors/creators, average on-line editors/creators time. Based on the Markovian Chain theory the dynamic aspect of repository development is analyzed as well. As the case study the presented model is interpreted in the e-learning context.

2 Community-Build System

All of our discussions regarding the community-built system refer to, base on [14]. The community-built system can be defined as a system of virtual collaborations organized to provide an open resource development environment within a given community [17]. Virtual collaborations, according to [21], are a type of collaborations in which individuals are interdependent in their tasks, share responsibility for outcomes, and rely on information communication technology to produce an outcome, such as shared understanding, evaluation, strategy, recommendation, decision, action plan etc. A common goals or deliverables are needed in order to create the community-built system [12]. In this case, the common goal is to develop open resources according to the requirements of the teaching-learning process. Moreover, the distance learning network, based on the community-built system, is backbone of education organisation not only for knowledge distribution, but also for creating high-level content through the community's work [15]. The tab. 1 presents main community-built system characteristics.

The operational environment of a community-built system is defined as a collaborative service provided within a collaborative workplace, supporting collaborative activities of a collaborative group [7]. Collaborative systems, groupware, or multi-user applications allow groups of users to communicate and cooperate on common tasks [19]. Such systems are modelled as dynamic and inter-independent, diverse, complex, partially self-organizing, adaptive and fragile [1]. The community-based system operates using the asynchronous pattern, what is considered as promoting and supporting cooperation rather than competition among its users [3]. Basing on [2] we can formulate key assumptions necessary for a successful community-built system:

- knowledge is created and it is shared,
- individual's have prior knowledge they can contribute during a discussion/collaboration,
- participation is critical to open resources development,
- individuals will participate, if given minimal conditions.

Table 1. Main Community-built system characteristics

Community-built system concepts	Community-built system technical features	Community-built system characteristics
Open	Open source	Each member can create, modify and delete the content
Organic	Rapidness	The structure and content are dynamic and non-linear and can be rapidly constructed, accessed and modified
Universal	Simplicity	Every member can shift roles on the fly
Observable	Maintainable	System maintains a version database, which records its historical revision and content
Tolerant	Self-repair	The content must fulfill interface requirements, the quality is decided by the community

3 Model of Community-Build System

3.1 Model's Structure

In the community-build system, having in mind the process of creating content, we can identify two roles: creator and editor. Repository is the system that integrates them (fig. 1.). The process of content creation always includes the social, research and education aspects and occurs across cognitive, information and computer-based levels of community-built systems. At each of these levels, the editors and the creators have their own roles and levels of involvement. The creator is a person responsible for creating and improving the content. The editor is a person responsible for content assessment.

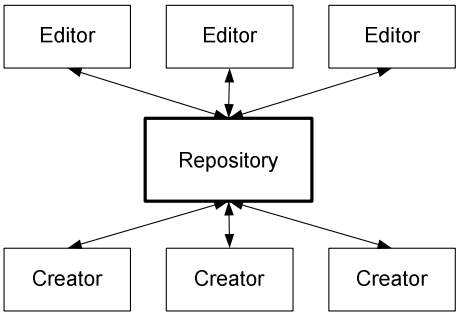


Fig. 1. Model of community-build system

In the community-build system model the repository plays mainly communication function. The main task of the repository is to ensure for each user the possibility to read, copy, distribute, print, search or link the full texts of documents and other

materials and documents placed in the repositories [16]. The repositories are a place for storing collections of digital documents, shared them in the network with the use of appropriate interface for a selected group of people or with unlimited access [8]. Developing the repositories is determined by many factors. They become a bridge between the rapid outdating of domain knowledge and the time of reaction to these changes, giving the possibility to quickly adapt and update the repository content. Another factor is a depth and context of knowledge stored in the repository. The quality of repository depends of the depth of domain description. The repository system may limit the access to the knowledge based on the creator's and editor's knowledge.

Analysis of the repository-based content creation process that is part of the community-build system model allows for identifying two situations. The first one assumes that it is impossible to change the role in the network (fig. 2a). Individual nodes of the community-build system play either the role of the editors or of the creators. Such an approach can be called as a community-built system with authority. The best example of this type of community-build system is an e-learning system, where students work under teacher supervision. The students fill up the repository with new knowledge objects [16] . The second approach assumes that a node can, in a certain moment, be either the editor or the creator, and the roles can be changed (fig. 2b). The Wikipedia is a best example of the second community-build system type.

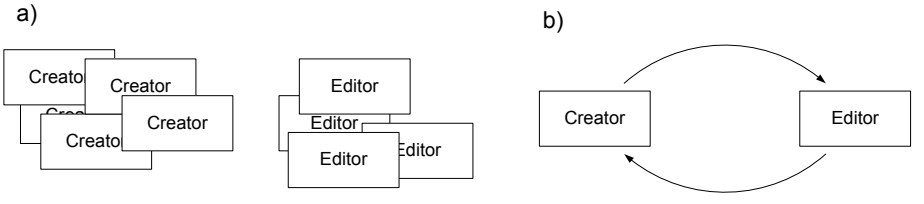


Fig. 2. Different relations between creators and editors

3.2 Actors Definition

All system users must be registered in the system. The system works in on-line mode. Let's introduce the following variables:

$E = (e_1, e_2, \ldots . e_j, \ldots . .)$ – set of editors, who are responsible for the content assessment,

$C = (c_1, c_2, \ldots . c_z, \ldots . .)$ - set of creators, who are responsible for the content creation,

$N_C(t)$ – number of creators at instance *t*,

$N_E(t)$ – number of editors in at instance *t*,

N – limited population of users, $N_C(t) + N_E(t) \le N$,

$S(t)$ – system's state at instance *t*, $S(t) = \begin{cases} (N_E(t) + N_C(t)) \\ N_0(t) = N - N_E(t) - N_C(t) \end{cases}$,

Where $N_E(t) + N_C(t) = N_A(t)$ - number of active actors and $N - N_A(t) = N_0(t)$ - number of passive actors.

The active actors are taken part in system working routine. The active (on-line) actors have following characteristics creator (content creation), editor (content evaluation and consultation). The passive actors are outside of the system, but they are registered in the system and can become active without any cost. The passive (off-line) actors have following characteristics creator (thinking, knowledge increasing process), editor (applied knowledge).

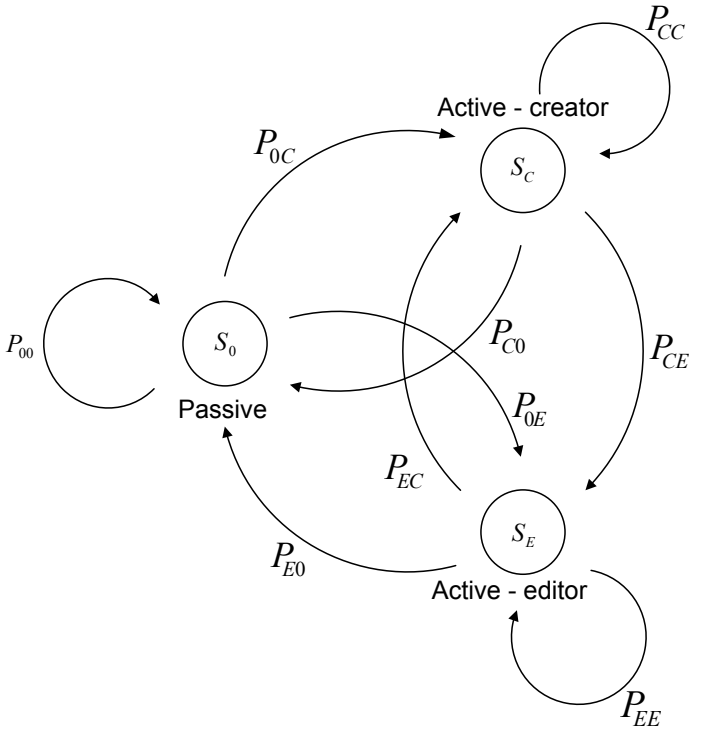


Fig. 3. Transition diagram for one registered member

The figure 3 presents the transition diagram for one registered member. Between the passive, active-editor and active-creator states the transitions probability can be distinguished. The most import transitions probabilities have following description:

P_{0C} -transition probability that actor has enough knowledge to become creator in time *t*

P_{0E} -transition probability that actor has enough knowledge to become editor in time *t*.

P_{EC} -transition probability that actor has enough knowledge to change role from editor to creator in time *t*.

P_{CE} -transition probability that actor has enough knowledge to change role from creator to editor in time *t*.

The actor is more likely willing to change state if she/he posses base knowledge adequate to the repository content. In some cases the actor's knowledge is so out of context (from other domain) that she/he should not change the state from the passive one.

3.3 Performance Measurements for the Content Creation Process

From the point of view of repository knowledge structure the editors as well as creators change the repository content. The creators add new content to the repository and the repository is growing. A main editor duty is to evaluate repository content. Some part of the repository can be deleted and modified. As a result the repository is decreasing. From the point of view of time the editors and creators activities can be formulated as a sequence of states. The fig.3 presents all possible states and transitions. Tab. 2 presents related transitions matrix $\overline{P} = [p_{ij}],\quad i,j = \overline{1,e} \Rightarrow (0,E,C)$ for corresponding fig. 3. Initial state for the system $\overline{\Pi}^{(0)} = (\overline{\Pi}_0^{(k)}, \overline{\Pi}_1^{(k)}, ..., \overline{\Pi}_e^{(k)}) \Rightarrow (n_0^{(0)}, n_E^{(0)}, n_C^{(0)})$. State of system at *k* step is $\overline{\Pi}^{(k)} = \overline{\Pi}^{(0)} \overline{P}^k$, where $\overline{P}^k = [p_{ij}^k],\quad i,j = \overline{1,e}$.

Table 2. Transitions probability matrix

	S_0	S_C	S_E
S_0	P_{00}	P_{0C}	P_{0E}
S_C	P_{C0}	P_{CC}	P_{CE}
S_E	P_{E0}	P_{EC}	P_{EE}

Transitions equation for states (*0, E, C*):

$$\begin{cases} n_0^{(k+1)} = n_0^{(k)} p_{00} + n_E^{(k)} p_{E0} + n_C^{(k)} p_{C0} \\ n_E^{(k+1)} = n_0^{(k)} p_{0E} + n_E^{(k)} p_{EE} + n_C^{(k)} p_{CE} \\ n_C^{(k+1)} = n_0^{(k)} p_{0C} + n_E^{(k)} p_{EC} + n_C^{(k)} p_{CC} \end{cases},$$

and corresponding Chapman–Kolmogorov equation $\hat{n}^{(k+1)} = \hat{n}^{(k)} \times P$.

For steady state $\lim_{k \to \infty} \hat{n}_j^{(k)} = \sum_{i=1}^{e} \hat{n}_i^{(k)} P_{ij}^{(k)}$.

Let's interprets the content creation process as a birth-death process. In time unit (*t*): $\lambda_C(t) = P_{0C} N_0(t) + P_{EC} N_E(t)$ - rate of the creator's arrival, $\lambda_E(t) = P_{0E} N_0(t) + P_{CE} N_C(t)$ - rate of the editor's arrival. From the ergodicity conditions:

$$N_C = \lim_{t \to \infty} N_C(t),\ N_E = \lim_{t \to \infty} N_E(t),$$

$$\lambda_C = \lim_{t \to \infty} \lambda_C(t), \lambda_E = \lim_{t \to \infty} \lambda_E(t),$$

Local balance equations: Birth equations: $\lambda_C = P_{0C} N_0 + P_{EC} N_E$, $\lambda_E = P_{0E} N_0 + P_{CE} N_C$. death equations: $\mu_C = (P_{C0} + P_{CE}) N_C$, $\mu_E = (P_{E0} + P_{EC}) N_E$.

$N_E = \sum_{c=0}^{N}\sum_{e=0}^{N} cP(c,e)$ - average number of editors,

$N_C = \sum_{e=0}^{N}\sum_{c=0}^{N} eP(c,e)$ - average number of creators,

$N_0 = N - N_E - N_C$ - average number of actors in passive state

$\lambda_E = P_{0E}N_0 + P_{CE}N_C$ - rate of editor's arrival in steady state

$\lambda_C = P_{0C}N_0 + P_{EC}N_E$ - rate of creator's arrival in steady state

Based on the Little's theorem: $T_E = \frac{N_E}{\lambda_E}$ - average on-line editors time, $T_C = \frac{N_C}{\lambda_C}$ - average on-line creators time.

3.4 The Sequential Model of Repository Knowledge Growing

The editor's and creator's knowledge can be represented in the form of ontology. In the [25] adequate formulization is presented. The ontology is defined as a tuple [25]: $\Omega =< S, \Pi >$, where $S = \{\phi_i\}, i = 1,...,n$ is a set of concepts from a specific domain. $\Pi : S \times S \to R$ is a mapping from an ordered pair of concepts to the set of connections *R={IS_A, PART_OF, ∅}*. The most important part of ontology is a set of concepts, especially its semantical deep regards to given domain. One of the approach to this problem is a matrix of the concept's depth [10]. The number of matrix's rows equals the number of attributes of the concept and the number of columns symbolizes the amount of objects merged into one concept's class [20]. The number of elements in the matrix t_{ij} is called the concept's depth in the specified domain. The concept ϕ, is defined as a tuple [10]: $\phi =< X, T >$, where *X* is a set of information, which provided the concept's description. *T* is a matrix of the concept's depth $T = [\hat{t}_{ij}]$ and *N(ϕ)* is a name of the concept ϕ. All elements of the matrix *T* belong to the specified domain, as a result, they are included in a ontology $\{N(\phi)$, *Object i, Attribute j,* $t_{ij}\} \in \Omega$, for $i = 1,...,I$, $j = 1,...,J$. Matrix *T* can be considers as concept's abstraction [10].

The editors and creators interact with the repository using ontology modeling methods [5,10,24,25]. Basing on the Markovian Chain approach the creator's and editor's interaction with the repository can be analysed as a time sequential process [23]. The repository changes the state: $G_R^{(0)}, G_R^{(1)}, G_R^{(2)}, ..., G_R^{(k)}, G_R^{(k+1)}, ...$. In the time *k+1* the repository can take following state: $G_R^{(k+1)} = \{G_R^{(k)} \setminus G_E^{(k)} \mid p_{0E}$; $G_R^{(k)} \bigcup G_C^{(k)} \mid p_{0C}$; $(G_R^{(k)} \setminus G_E^{(k)}) \bigcup G_C^{(k)} \mid p_{EC}$; $(G_R^{(k)} \bigcup G_C^{(k)}) \setminus G_E^{(k)} \mid p_{CE}\}$.

Transition equation for repository state $k \to k+1$: $\left|G_R^{(k+1)}\right| = \left|G_R^{(k)} \setminus G_E^{(k)}\right| p_{0E}^{(k)} + \left|G_R^{(k)} \bigcup G_C^{(k)}\right| p_{0C}^{(k)} + \left|G_R^{(k)} \setminus G_E^{(k)} \bigcup G_C^{(k)}\right| p_{EC}^{(k)} + \left|G_R^{(k)} \bigcup G_C^{(k)} \setminus G_E^{(k)}\right| p_{CE}^{(k)}$. Transition equation for repository state $0 \to k$: $\left|G_R^{(k)}\right| = \sum_{e=0}^{k} (\left|G_R^{(e)} \setminus G_E^{(e)}\right| p_{0E}^{(e)} + \left|G_R^{(e)} \bigcup G_C^{(e)}\right| p_{0C}^{(e)})$.

The repository content is increasing or decreasing on the *k* step: $\Delta G_R^{(k)} = \left|G_R^{(k+1)}\right| - \left|G_R^{(k)}\right|$. The formula helps evaluate the development of the repository.

4 e-Learning Application

In the [16] the collaboration environment for knowledge management in competence based learning basing on the CBS model is proposed. The collaboration take place in e-learning system mode. In the system the teacher plays editor role and the student is creator. The system supported the competency-based learning process by means of social network creation in order to exchange ontology in the repository environment. The process of the competency-based learning always includes the social, research and education aspects and occurs across cognitive, information and computer-based levels of community-built systems. At each of these levels, the editors (teacher) and the creators (students) have their own roles and levels of involvement.

The model of an e-learning information system integrates three sub models. Each of them add new component to the community-built system structure. Lest' analyzed each sub model separately in order to establish the relation with community-build system:

- Knowledge management model, The first level (knowledge level) concerns proper preparation of the learning process from the point of view of the structure of domain knowledge. In accordance with the accepted definition of knowledge, the knowledge structurization process includes preparation of a portion of theoretical and procedural knowledge, and of corresponding test tasks, to verify students' skills/abilities. This model supports the content creation process for community-built system.
- Social network model, The social network level is the basis for determining conditions for cooperation between the teacher and the students. This cooperation creates a new dimension for social networks, described by appropriate functions of motivation [9].
- Model for quantity system assessment Checking the conditions of cooperation takes place at the management level (student – e-learning information system – teacher interaction). The established cooperation conditions are analyzed from the point of view of the existing constrains (human resource, time resource). The obtained results can be a signal to change the cooperation model. In addition, feedback information is transmitted, concerning further repository development and learning process participants collaboration.

5 Conclusion

The proposed model is one of the possible interpretations of collaboration knowledge process. The quantity nature on the model is important in context of future knowledge management system for community-build system. The biggest challenge is a detailed description of ontology modeling process in community-build system environment. One of the solutions is an analogy with the knowledge object (the learning object) from e-learning systems [25]. The e-learning knowledge object has the meat-description and communication mechanisms as well.

The motivation for collaboration is other important research issue in community-build system area [14]. Using the motivation model, the activity of the

community-built system can be analyzed equally on both technical and the knowledge levels. The motivation model covers two functions important for motivation: that of the creator and that of the editor and describes their unique interests in supplying a knowledge repository using the Wiki mechanism.

References

1. Brown, J.S.: Leveraging technology for learning in the cyber age – opportunities and pitfalls. In: International Conference on Computers in Education (ICCE 1998) (1998) (invited Speaker)
2. Cole, M.: Using Wiki technology to support student engagement: Lessons from the trenches. Computers & Education 52(1), 141–146 (2009)
3. De Pedro X., Rieradevall M., Lopez P., Sant D., Pinol, J., Nunez, L.: Writing documents collaboratively in Higher Education using traditional vs. Wiki methodology (I): qualitative results from a 2-year project study. In: International Congress of University Teaching and Innovation, Barcelona (July 5–7, 2006)
4. Ebersbach, A., Glaser, M., Heigl, R.: Wiki: Web Collaboration, 2nd edn. Springer, Heidelberg (2008)
5. Fernández-Breis, J., Martiinez-Bejar, R.: A cooperative framework for integrating ontologies. International Journal of Human-Computer Studies 56(6), 665–720 (2002)
6. Ho, C.-T., Chen, Y.-M., Chen, Y.-J., Wang, C.-B.: Developing a distributed knowledge model for knowledge management in collaborative development and implementation of an enterprise system. Robotics and Computer-Integrated Manufacturing 20(5), 439–456 (2004)
7. ISO 19778: Information technology - Learning, education and training - Collaborative technology - Collaborative workplace. International Organization for Standardization (2008)
8. Jones, R.: Giving birth to next generation repositories. International Journal of Information Management 27, 154–158 (2007)
9. Kusztina, E., Zaikin, O., Tadeusiewicz, R.: The research behavior/attitude support model in open learning systems. Bulletin of the Polish Academy of Sciences: Technical Sciences 58(4), 705–711 (2010)
10. Kusztina, E., Różewski, P., Zaikine, O.: Extended ontological model for distance learning purpose. In: Reimer, U., Karagiannis, D. (eds.) PAKM 2006. LNCS (LNAI), vol. 4333, pp. 155–165. Springer, Heidelberg (2006)
11. Lowyck, J., Poysa, J.: Design of collaborative learning environments. Computers in Human Behavior 17(5-6), 507–516 (2001)
12. Ommeren, E., Duivestein, S., de Vadoss, J., Reijnen, C., Gunvaldson, E.: Collaboration in the Cloud - How Cross-Boundary Collaboration Is Transforming Business. Microsoft and Sogeti (2009)
13. Pardede, E.: Community-Built Database: Research and Development. Springer, Heidelberg (2011)
14. Różewski, P., Kushtina, E.: Motivation Model in Community-Built System. In: Pardede, E. (ed.) Community-Built Database: Research and Development. Springer ISKM Series, pp. 183–205. Springer, Heidelberg (2011)
15. Różewski, P.: A Method of Social Collaboration and Knowledge Sharing Acceleration for e-learning System: the Distance Learning Network Scenario. In: Bi, Y., Williams, M.-A. (eds.) KSEM 2010. LNCS (LNAI), vol. 6291, pp. 148–159. Springer, Heidelberg (2010)

16. Różewski, P., Ciszczyk, M.: Model of a collaboration environment for knowledge management in competence based learning. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 333–344. Springer, Heidelberg (2009)
17. Sheehy, G.: The Wiki as Knowledge Repository: Using a Wiki in a Community of Practice to Strengthen K-12 Education. TechTrends 52(6), 55–60 (2008)
18. Shen, D., Nuankhieo, P., Huang, X., Amelung, C., Laffey, J.: Using Social Network Analysis to Understand Sense of Community in an Online Learning Environment. Journal of Educational Computing Research 39(1), 17–36 (2008)
19. Tolone, W., Ahn, G.-J., Pai, T., Hong, S.-H.: Access Control in Collaborative Systems. ACM Computing Surveys 37(1), 29–41 (2005)
20. Tsichritzis, D.C., Lochovsky, F.H.: Data models. Prentice Hall – Professional Technical Reference (1982)
21. Wagner, C.: Wiki: A technology for conversational knowledge management and group collaboration. Communications of the AIS 13, 265–289 (2004)
22. West, J.A., West, M.L.: Using Wikis for Online Collaboration: The Power of the Read-Write Web. Jossey-Bass a Wiley Imprint (2008)
23. Wheeler, S., Bean, N., Gaffney, J., Taylor, P.: A Markov analysis of social learning and adaptation. Journal of Evolutionary Economics 16(3), 299–319 (2006)
24. Yao, Z., Liu, S., Han, L., Reddy, Y.V.R., Yu, J., Liu, Y., Zhang, C., Zhaoqing, Z.: An Ontology Based Workflow Centric Collaboration System. In: Shen, W., Luo, J., Lin, Z., Barthès, J.-P.A., Hao, Q. (eds.) CSCWD. LNCS, vol. 4402, pp. 689–698. Springer, Heidelberg (2007)
25. Zaikine, O., Kushtina, E., Różewski, P.: Model and algorithm of the conceptual scheme formation for knowledge domain in distance learning. European Journal of Operational Research 175(3), 1379–1399 (2006)

A Multi-Agent Scheduling Approach for the Joint Scheduling of Jobs and Maintenance Operations in the Flow Shop Sequencing Problem

Si Larabi Khelifati and Fatima Benbouzid-Sitayeb

Laboratoire des Méthodes et Conception de Systèmes (LMCS)
Ecole nationale Supérieure d'Informatique (ESI)
BP 68M Oued-Smar Algiers (Algeria)
{s_khelifati,f_sitayeb}@esi.dz

Abstract. In this paper, we propose a distributed approach using multi-agent paradigm for scheduling independent jobs and maintenance operations in the flowshop sequencing problem. The proposed system introduces a dialogue between two communities of agents (production and maintenance) leading to a high level of cooperation. It also provides a framework in order to react to the disturbances occurring in the workshop. This aspect is mainly concerned by corrective maintenance activities (machine breakdowns). The proposed multi-agent system is based on the integrated strategy which consists of a simultaneous scheduling of both production and preventive maintenance activities. The objective is then to optimize an objective function which takes into account both maintenance and production criterion. The main point is to show how the proposed multi-agent system is used to generate a joint production and preventive maintenance schedule that provides a better compromise between the satisfactions of respective objectives of the two functions.

Keywords: Production, Preventive Maintenance, Corrective Maintenance, Joint Scheduling, Flow Shop, Multi-Agent System.

1 Introduction

Scheduling is one of the most significant activities of the control system since it aims to plan orders on machines to ensure customer satisfaction (through the respect of the due dates) and system profitability (through an efficient use of the resources). One of the most frequent production scheduling problems is the Flow Shop Problem (FSP). This problem implies a natural ordering of the machines in the shop, in such a way that different jobs go through the same machines in the same order.

Another task closely related to production scheduling in industrial settings is maintenance. However, both activities conflict since they act on the same resources (machines). As known, maintenance operations consume production

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 60–69, 2011.

time but are mandatory to ensure the long term availablity of production system. However, delaying maintenance operations because of production demands, may increase the probability of machine failure. Classical scheduling approaches known as static schedudiling, usually solve the problem with optimal or suboptimal schedule [1] [2] [4] [5] [13], but they can easily become unfeasible in real manufacturing environment since they assume highly unrealistic assumptions. Hence, such dynamic manufacturing systems require dynamic scheduling.

The Multi-Agent Systems (MAS) are a distributed artificial intelligence domain in fully effervescence. The multi-agent approach introduces a high reliability facing the evolution and the disturbances of the context. MAS aim at solving complex problems by cooperation between several autonomous agents [8] [16]. A state-of-the-art of multi-agent approaches for intelligent manufacturing can be found in [14]. Considering the integration of production and maintenance services, many works refer to the scheduling problem of both systematic preventive maintenance and production activities. In this way, a cooperation process using the multi-agent paradigm has been proposed in [6] [7].

This paper develops a distributed approach using multi-agent paradigm for scheduling independent jobs and maintenance operations in the flowshop sequencing problem based on the integrated strategy [10]. This strategy consists of a simultaneous scheduling of both production and preventive maintenance activities. The objective is then to optimize an objective function which takes into account both maintenance and production criterion. It also provides a framework in order to react to the disturbances occurring in the workshop which is mainly concerned by corrective maintenance operations (machine breakdowns).

The remainder of this paper is organized as follows. In section 1, we discuss some relevant studies on multi-agents system for scheduling, and then we describe some approaches aiming at making production and maintenance cooperate. In section 2, we present the formulation of the static permutation flow shop problem, the basic concepts on maintenance and the objective functions to optimize. Section 3 and section 4 present respectively the proposed joint production and maintenance multi-agent scheduler and the reactive cooperation strategy. Finally, in the last section the results of this approach are presented regarding the computational experience followed by suggested future research.

2 Models and Problem Statement

The current section describes the production and the maintenance data and the common objective function to optimize.

2.1 Formulation of the Static Permutation Flowshop Problem

The FSP can be stated as follows: each of n Production Operations (PO) $1, ..., n$ has to be processed on m machines $1, ..., m$ in that order. The processing time of job j on machine i is p_{ij} and its completion time is c_{ij}. The processing times are fixed, nonnegative, and may be 0 if a job is not processed on some machine.

Here we consider the Permutation Flow Shop Problem (PFSP) i.e, the same job order is chosen on every machine. The objective is then to find a sequence, i.e, a permutation of the numbers $1, ..., n$ that minimizes the completion time C_{max}, also called makespan, of the last job on the last machine.

2.2 Scheduling and Maintenance

The PM operation can be stated as follows: each of m machines $1, ..., m$ has to be maintained periodically at known intervals of time. The PM operations are periodic interventions occurring every T periods and each occurrence depends on the ones preceding it on the same machine. The processing time of PM operation j on machine i is p'_{ij} and its completion time is is c'_{ij}. The processing times are fixed, nonnegative and evaluated with more or less certainty. Moreover, the periodicity T of these operations can vary in a tolerance interval noted $[T_{min}, T_{max}]$ (Fig.1).

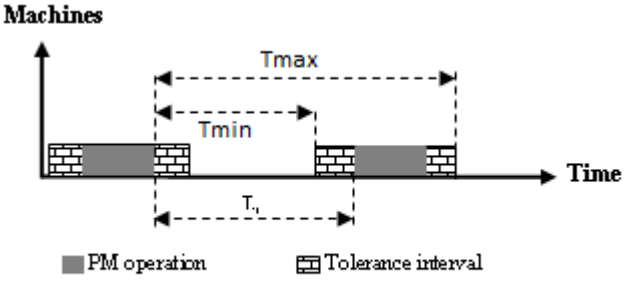


Fig. 1. Tolerance interval of a PM operation

A machine i can be subject to different PM operations that will be repeated periodically. Let:

- M_{ij}: the PM operation j on machine i.
- T_{ij}: periodicity of the PM operation M_{ij}.
- $T_{min_{ij}}$: shortest time separating two consecutives occurrences of M_{ij};
- $T_{max_{ij}}$: longest time separating two consecutives occurrences of M_{ij};

The PM operations are periodic interventions occurring every T periods. Thus, the k^{th} occurrence of the PM operation M_{ij} is denoted M_{ijk}. Ideally a PM operation is programmed inside the interval $[Tmin_{ij}, Tmax_{ij}]$. However, it can be programmed before $Tmin_{ij}$ and thus be considered in advance (the advance is noted E'_{ijk}), or after $Tmax_{ij}$ and thus be considered as late (the delay is noted L'_{ijk}). Advance and delay of M_{ijk} is computed as follows:

- t'_{ijk}: execution time of the k^{th} occurrence of M_{ij}.
- E'_{ijk}: advance of the k^{th} occurrence of M_{ij}.

$E'_{ijk} = max(0, c'_{ijk} + p'_{ij} + T_{min_{ij}} - c'_{ijk-1}$

- L'_{ijk}:delay of the k^{th} occurrence of M_{ij}
$L'_{ijk} = max(0, c'_{ijk-1} - c'_{ijk} - p'_{ij} - T_{max_{ij}})$

As opposed to PM operations, whose scheduling is a necessity known in advance, CM operations are operations that have to be scheduled on purpose in case one or more machine undergoes a sudden breakdown. A CM operation can be stated as a PM one with particularity that its tolerance interval is equal to 0. The CM operations notations are the same as PM ones unless those concerning the tolerance interval.

2.3 Objective Functions

The aim of optimization must be a compromise between the target objective production and maintenance functions. The production objective function is the completion time of the last job on the last machine (C_{max}). We note f_1 this function:

$$f_1 = C_{max} = Max(c_{ij})$$

Let f_2 the temporal deviation of the maintenance operations w.r.t. their ideal time allocations expressed as tolerance temporal intervals. f_2 is computed as follows:

$$f_2 = \sum_{i=1}^{m} \sum_{j=1}^{m_i} \sum_{k=1}^{m_{ij}} (E'_{ijk} + L'_{ijk})$$

where m_i represents the number of PM operation on machine i. m_{ij} represents the occurrence number of the PM operation M_{ij}

To optimize the two criteria, we take into account the following common objective function:

$$f = f_1 + f_2$$

3 A Dynamic Joint Scheduling Approach Based on a Multi-Agent System

We have to find, on a limited horizon, a joint production and preventive maintenance schedule. This schedule must indicate for each operation, of the shop floor, the machines on which it has to be achieved and at what times. We suggest an approach based on a located multi-agent system. A joint schedule emerges from the interactions between the different agents.

3.1 Principles of the Proposed Approach

Based on the above insights, we use the Vowels methodology [12] to design the MAS. This methodology requires the definition of the agents, the definition of the environment, the definition of the interactions and the definition of the model dynamic. This led us to an architecture that has the following five types of agents, whose role is detailed in the next section :

- a production supervisor agent (Prod-Sup-Agent);
- one maintenance supervisor agent per machine (Maint-Sup-Agent);
- one machine agent per machine (Machine-Agent);
- maintenance agents (Maint-Agent);
- production agents (Prod-Agent).

Agent's communication is achieved by message sending. There are three kinds of communication:

- communication between machines;
- communication between each Maint-Sup-Agent and the maintenance agents under its supervision and communication between each Maint-Sup-Agent and its associated Machine-Agent;
- communication between the Prod-Sup-Agent and the Prod-Agents and communication between the Prod-Agents and the Machine-Agents.
- communication between the production agents and the maintenance agents is achieved through the Machine-Agents.

3.2 Dynamic Scheduling Processes

An integrated resolution of the joint production and PM scheduling problem using the multi-agent paradigm requires the definition of a preceding order among all the operation to be ordered (production and PM ones). The proposed MAS operates in two steps: (1) Prod-Sup-Agent starts the generation of a production schedule which respects the production operation characteristics, under the objective of optimizing the production objective function f_1 defined in the previous section. On the other hand, each Maint-Sup-Agent creates and initializes the maintenance agents and starts the generation of partials PM operations schedules on the machine to which it is associated, under the objective of optimizing the maintenance objective function f_2 defined in the previous section; (2) each Machine-Agent chooses the next operation (production or PM) to be performed on the machine to which it is associated depending on the data that it receives from the preceding machines.

The production scheduling process uses a two-step scheduling process: first Prod-Sup-Agent schedules the production operations according to their late start date on the first machine (Let S1 be the resulting schedule) and it schedules the production operations according to their early start date on the last machine (Let S2 be the resulting schedule). Then Prod-Sup-Agent uses S1 and S2 to generate a production schedule S as detailed in algorithm 1.

As stated above, scheduling maintenance activities cannot be directly assimilated to scheduling manufacturing activities [11] [9]. The maintenance planning is generated using the periodicity T of each PM operation. Each Maint-Sup-Agent schedules the PM operations according to their T_{min} on each machine. When they occur, conflicts are handled as outlined in Algorithm 2.

When the scheduling of both production and PM operations are achieved (respectively by Prod-Sup-Agent and the Maint-Sup-Agents), each machine agent

Algorithm 1. Production Scheduling

```
Begin
While  S1 is not empty Do
  Begin
  If there is no conflicts  between two or more production operations
  in S1
  Then Schedule in S the PO as they are scheduled in S1
  Else Schedule in S the PO in conflict as they are scheduled in S2
  End
End.
```

Algorithm 2. Maintenance Scheduling

```
Begin
If two PM operations or more have the same Tmin
  Then Begin
       Schedule these PM operations according to their Tmax
       If two PM operations or more have the same Tmax
       Then  Schedule these PM operations according to their
       processing time (the PM operation with the highest
       processing time should be placed first)
       EndIf
       End
End.
```

asks for the Prod-Sup-Agent to indicate him the next production operation to schedule and for its Maint-Sup-Agent to indicate him the next PM operation to schedule, aiming to perform a joint production and PM schedule. Two states can occur:

- only one operation is proposed (either a production one or a PM one): this operation is scheduled;
- a production operation and a PM one are proposed: there is in this case an access conflict to the machine. A conflict between a production operation PO_{ij} and PM operation M_{ik} on the machine i is resolved with the function O as detailed above:

$$O = Max(Weight_{PO_{ij}}, Weightmaint_{M_{ik}})$$

Where

$$Weight_{PO_{ij}} = Tmax_{ij} - (c_{ij} + p_{ij})$$
$$Weight_{M_{ik}} = laststartdateofp_{ij} - (c'_{ik} + p'_{ik})$$

The machine agent schedules the operation with the higher weight. It informs Prod-Sup-Agent or Maint-Sup-Agent. The performed operation is removed from the pending operation list. It, then, sends the following message to the agent to whom the execution of the operation behoves: "the operation PO must be executed at the time t" where t is the completion time of the operation under

execution. If the conflict still exits after the calculation of the O function i-e $Weight_{prod} = Weight_{maint}$, the concerned machine agent asks for its associated Maint-Sup-Agent to widen the PM operation tolerance interval i.e., the tolerance interval bounds are widened. The process is then re-launched.

4 The Reactive Cooperation Strategy

A reactive cooperation strategy can be developed between the different agents in order to take into account the CM operations. This strategy aims to guarantee as long as possible the stability of the orders released into the shop. The proposed strategy emphasizes a local re-scheduling rather than a complete one. A CM operation is associated to a maintenance agent. It is characterized by a tolerance interval with its T_{min} equal to 0. A priority level can also be considered if available. A CM operation is only planned in the case of reactive scheduling, when a machine breakdown occurrence has been reported.

After a CM operation is launched, the process consists in identifying the Set of Perturbed Production Operations (SPPO), the Set of Concerned Machine Agents (SCMA). The perturbed operations are identified by a machine agent by propagation of the disturbance within its local plan. The concerned machine agents are the machine agents, who has first been perturbed and all the machine agents that can achieve the activities required by the perturbed operations. Considering these two sets, the following steps consist in performing a new joint production and PM schedule of the affected operations as detailed in the preceding section.

5 Tests and Results

In this section, we present the results of a series of computational experiments conducted to test the effectiveness of the proposed multi-agent approach. We report on two experiments. In the first one, we have undertaken an analysis of performance of the integrated strategy. The second one analyses the effectiveness of the proposed reactive cooperation strategy. The complete details are reported below.

To our knowledge, there is no instance for the joint production and PM scheduling problem. Therefore, the test problems for evaluating our approach are non-standard ones. They have been generated by following the procedure given by [4] for generating joint production and PM benchmark flowshop scheduling problems. The objective functions taken into consideration are detailed in Section 2.

In this study, we consider the machine breakdown disturbances. We suppose that the number of disturbances is low, that the number of operations to schedule is finished, and that there is no closure time for the machines. The machines are either available, or in processing or in failure after a disturbance.

The MAS as well as the simulation environment was implemented in JADE [3]. Each simulated entity was implemented as a separate agent.

One difficulty faced by researchers in scheduling is to compare their developed heuristics with those of other researchers. If the standard set of test problems is accessible, different algorithm's performances can be compared on exactly the same set of test problems. In the case of the joint production and PM scheduling problem, few computational results can be found and no optimal or near optimal results exists. Since no competitive results for this problem are found in the literature, we compare our results with the upper bound values of makespan yielded by [15] according to the performance loss after maintenance insertion. When comparing the solutions given by our SMA, the best one is the one that gave the least increase. Figure 2 presents the results of the evaluation of the integrated strategy by considering the objective of optimizing the objective function f detailed in section 2.

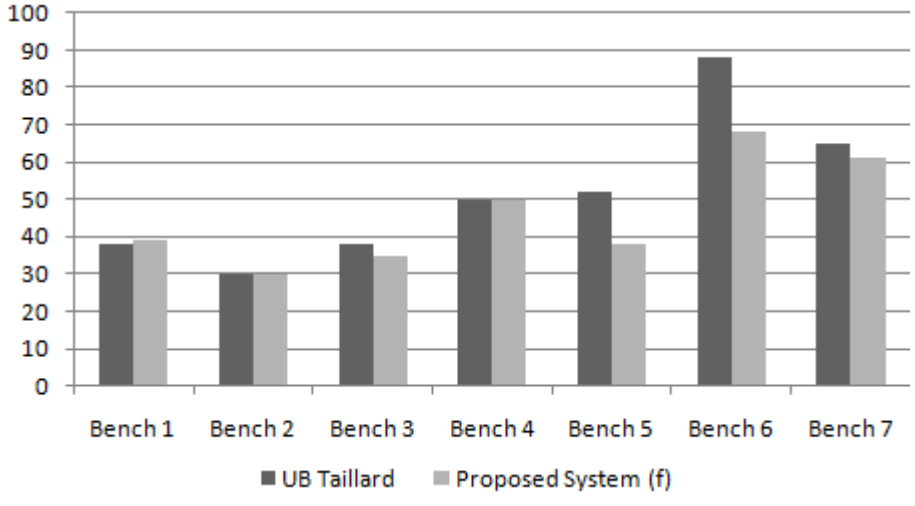


Fig. 2. Comparaison between the C_{max} given by [15] and the proposed strategy

It is evident that the best values for makespan given by Taillard are better than the values yielded by the proposed SMA. The primary reason for such a performance is that the insertion of PM operations increases the makespan values. A fair comparison of our results with the upper bound values of makespan presented by [15] is difficult since Taillard benchmarks are dedicated to production scheduling only while ignoring PM. Moreover, the computational results of the proposed SMA are obtained with a set of non-standard test problems generated by following the procedure given by [4]. Figure 3 shows the results obtained when the reactive cooperation strategy is introduced in the global process.

Figure 3 shows that on the whole, the reactive cooperation strategy decreases the performances of the proposed MAS. It does suggest that taking into account the CM operations may delay the scheduling process. This worst performance

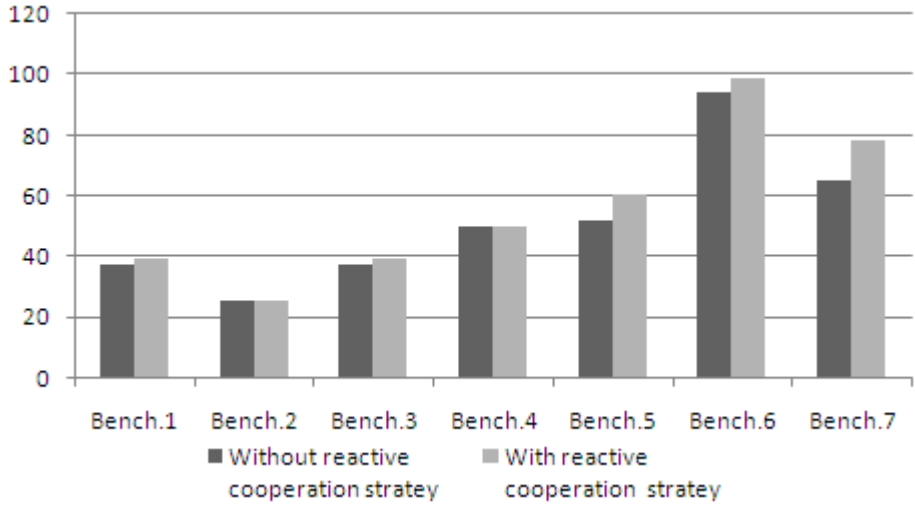


Fig. 3. The Reactive Cooperation Strategy

is mainly due to the fact that the CM operations are additive PM operations which have no tolerance interval.

6 Conclusion

A new method based on the multi-agent techniques has been developed in this paper to address the problem of the joint production and PM scheduling in permutation flowshop. Moreover, we suggest a reactive cooperation strategy to take into account the CM activities.

The perspectives for this work concern new developments in order to have a better model of competences for scheduling. In this paper, we do not consider the human resources. In reality, the problem is more complex and need more appropriated models. Other developments concern the possibility to gather different maintenance activities (corrective, systematic and condition based).

References

1. Aggoune, R.: Minimizing the makespan for the flow shop scheduling problem with availability constraints. European Journal of Operational Research 153, 534–543 (2004)
2. Allaoui, H., Artiba, A.: Integrating simulation and optimization to scheduling a hybrid flow shop with maintenance constraints. Computers and Industrial Engineering 47, 431–450 (2004)
3. Bellifemine, F., Poggi, A., Rimassa, G.: Developing Multi-agent Systems with a FIPA-compliant Agent Framework. Software: Practice and Experience 31(2), 103–128 (2001)

4. Benbouzid, F., Varnier, C., Zerhouni, N.: Resolution of joint maintenance/production scheduling by sequential and integrated strategies. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, pp. 782–789. Springer, Heidelberg (2003)
5. Cassady, C.R., Kutanoglu, E.: Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. IIE Transactions 35(6), 503–513 (2003)
6. Coudert, T., Grabot, B., Archimede, B.: Production /Maintenance Co-operative Scheduling Using Multi-Agents and Fuzzy Logic. International Journal of Production Research 40(18), 4611–4632 (2002)
7. Coudert, T., Grabot, B., Archimede, B.: Systmes multi-agents et logique floue pour un ordonnancement cooperatif Production/Maintenance. Journal of Decision Systems 13(1), 27–62 (2004)
8. Ferber, J.: Multi-Agent Systems: an Introduction to Distributed Artificial Intelligence. Addison-Wesley, Reading (1999)
9. Gits, C.W.: Structuring maintenance control systems. International Journal of Operations and Production Management 14(7), 7–15 (1994)
10. Lee, C.Y., Chen, Z.L.: Scheduling jobs and maintenance activities on parallel machines. Naval Research Logistics 47, 145–165 (2000)
11. Paz, N.M., Leigh, W.: maintenance scheduling: issues, results and research needs. International Journal Of Operations and Production Management 14(8), 47–69 (1994)
12. Ricordel, P.-M.: Programmation Oriente Multi-Agents: Dveloppement et Dploiement de Systmes Multi Agents Voyelles, Ph.D. Thesis, Grenoble University, France (2001)
13. Ruiz, R., Garca-Daz, J.C., Maroto, C.: Considering scheduling and preventive maintenance in the flowshop sequencing problem. Computers and Operations Research 34, 3314–3330 (2007)
14. Shen, W., Norrie, D.H.: Agent-based system for intelligent manufacturing: a state of the art survey. Knowledge and Information Systems 1(2), 129–156 (1999)
15. Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research 64, 278–285 (1993)
16. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. The Knowledge Engineering Review 10(2), 115–152 (1995)

Aligning Simple Modalities in Multi-agent System

Wojciech Lorkiewicz[1,2], Grzegorz Popek[1,2],
Radosław Katarzyniak[1], and Ryszard Kowalczyk[2]

[1] Wroclaw University of Technology,
Institute of Informatics
{wojciech.lorkiewicz,grzegorz.popek,radoslaw.katarzyniak}@pwr.wroc.pl
[2] Swinburne University of Technology
Faculty of Information and Communication Technologies
{wlorkiewicz,gpopek,rkowalczyk}@groupwise.swin.edu.au

Abstract. The most important role of a language is to facilitate interplay between individuals and to allow the entire population to communicate. As such, it is crucial for a distributed system of interacting agents to share a coherent language. In particular, in order to maintain consistent and understandable descriptions of the dynamic states of the external world the agents need to autonomously develop and adjust their individual semantics. In this paper we consider a set of agents capable of grounding and aligning autoepistemic statements with operators of possibility, belief, and knowledge. Assuming basic initial constraints and a pair-wise interaction scheme we show how a population of agents is able to align the meaning of utilised statements, thus allowing the agents to successfully communicate about the state of the external world. Using a simulated framework we further characterise the dynamic behaviour of the proposed solution.

Keywords: multi-agent systems, semantic communication.

1 Introduction

Language is strictly used in a social setting in order to aid communication between individuals. In particular, it assists the process of coordination, allows the exchange of information, and helps utilising multiple social functions - as Cangelosi notes 'the main and most obvious function of language is its pragmatic role, i.e. that of facilitating interaction and communication between cognitive agents'.

An approach presented in [4,5,6] follows Dennett's [1] stance that '*exposure to x – that is, sensory confrontation with x over suitable period of time – is the normally sufficient condition for knowing (or having true beliefs) about x*' and proposes an intuitive mechanism of how beliefs can be meaningfully related to the empirical experience. It introduces a concept of the epistemic satisfaction relation which defines the proper binding between the linguistic form of semiotic symbols

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 70–79, 2011.

and the internal meaning. The epistemic satisfaction relation is implemented through a system of modality thresholds over simple statistics which satisfies common-sense requirements, i.e. imposed on formed beliefs in a given modal language of communication.

It should be stressed that the general restrictions presented in [6] reflect only the common-sense requirements that are imposed on the epistemic satisfaction relation, while its precise realisation should also reflect an appropriate social consensus. In particular, concrete implementation of epistemic satisfaction relation should result from the shape of the external environment the system is situated in and the social context of the interacting population, i.e., common point of view accepted by the population. Moreover, a consistent epistemic satisfaction relation within the society is needed not only as it allows the individuals to communicate with each other, but as it also allows to incorporate already existing reasoning mechanisms, e.g., allowing the agents to share meaningful information and integrate it with their individual perspectives [7].

In this paper we assume a population of cognitive agents internally organized as in [4]. As such, section 2 describes the epistemic satisfaction relation and its crucial connection with an embodied meaning, and section 3 specifies the general model used in the paper including an internal organization of an agent and a process of grounding of simple modalities. Further, in section 4 we describe a pair-wise interaction scheme based on language games [9] and in section 5 we show how it enables the society of agents to reach the shared meaning adjusted to the environment. A brief summary of the paper is given in section 6.

2 Epistemic Satisfaction Relation

Semiotic symbols [2] – with respect to the so-called semiotic definition of symbols – should be considered in relation to cognitive agents that are carriers of their sense [4]. The role of semiotic symbols in semantic communication is defined by three elements of the semiotic triangle [8]: the 'material' form of the symbol, e.g. linguistic form available to the entire population, the sense assigned to the 'material' form by a certain cognitive agent, and a real object usually located in an external world to which the semiotic symbol is referred by a certain cognitive agent.

In essence, the epistemic satisfaction relation [4] binds the 'material' form of a semiotic symbol with its embodied meaning. As such, it defines agent's internally grounded meaning of a semiotic symbol and defines its individual naming convention. Consequently, effective communication of agents can be realized if and only if similar implementations of epistemic satisfaction relations are grounded in all agents. The epistemic satisfaction relation should be developed in a certain social context and reflect a common point of view accepted by an entire population of communicative agents.

Research presented in [5] assumes particular common-sense claims and guarantees rationality by evaluating and stating boundaries on the epistemic satisfaction relation in form of a system of modality thresholds (the way in which the system is applied is discussed in section 3.2).

In the current approach the final form of the relation needs to be stated a priori by a domain expert. In this paper we show how to avoid the need for an extensive extrinsic knowledge and present an example on how to develop the relation during runtime in an automatic and autonomous way in a distributed society of cognitive agents.

3 General Model

External world (after [5]) is a dynamic system consisting of objects $o \in O$ exhibiting properties $P \in \Delta$. Properties change over time but at each particular point of time each property is or is not exhibited by each particular object.

At each particular time point t the current state of the world can be given by a relation

$$\mathbb{R}_t : O \times \Delta \rightarrow \{0, 1\}$$

where $\mathbb{R}_t(o, P) = 1$ holds if and only if an object o at a time point t exhibits P.

3.1 Agent

Each agent observes the external environment and collects the resultant observations in it's private and strictly internal knowledge base. Certainly such observations do not have to cover the entire state of the world, as for instance agent might be only focused on a particular part of the actual state. Formally the observations are encoded in a relational form called base profiles:

Definition 1. *Base Profile. An agent a stores results of its observation at each particular time point t in a relational form of base profiles. After [5], structure of the base profile is given as follows:*

$$BP^a(t) = \left\langle O, P^+_{a,1}(t), P^-_{a,1}(t), P^+_{a,2}(t), P^-_{a,2}(t), \ldots, P^+_{a,K}(t), P^-_{a,K}(t) \right\rangle.$$

where:

- $P^+_{a,i}(t) \subseteq O$ *and for each object* $o \in O$ *the condition* $o \in P^+_{a,i}(t)$ *holds iff. the agent perceived o as exhibiting the property P_i at the time point t,*
- $P^-_{a,i}(t) \subseteq O$ *and for each object* $o \in O$ *the condition* $o \in P^-_{a,i}(t)$ *holds iff. the agent perceived o as not exhibiting the property P_i at the time point t.*

Observations are consistent with the environment in the sense that if $o \in P^+_{a,i}(t)$ *then* $\mathbb{R}_t(o, P_i) = 1$ *and if* $o \in P^-_{a,i}(t)$ *then* $\mathbb{R}_t(o, P_i) = 0$.

As already noted, observations are stored in strictly private and internal database, called knowledge state:

Definition 2. *Knowledge State. All observations gathered by an agent up to a time point t_j are represented by a knowledge state $KS^a(t)$ given as:*

$$KS^a(t) = \{\langle t_i, BP^a(l) \rangle : t_i \in T \text{ and } i < j\}.$$

3.2 Grounding

Following Dennett [1] the theory of grounding [4] defines a mechanism for an agent to ascribe autonomously created mental 'patterns' extracted from the past empirical experiences to the unobserved parts of the environment. Each mental 'pattern' corresponds to a particular realisation of an unobserved property in a given object. Moreover each episode in which an agent observed an object o exhibiting P (not exhibiting P, respectively) makes a corresponding mental model stronger in relation to all complementary models. These strengths are called relative grounding strengths and are denoted as $\lambda^a(t, P(o))$ (and $\lambda^a(t, \neg P(o))$).

Based on relative grounding strengths an agent grounds appropriate modal statements according to the epistemic satisfaction relation. A key role in this process is played by the system of modality thresholds. At each particular point of time it is given for an agent a as $\lambda^a_G(t) = \langle \lambda^a_{\text{minPos}}(t), \lambda^a_{\text{minBel}}(t), \lambda^a_{\text{maxBel}}(t) \rangle$.[1] Based on $\lambda^a_G(t)$.[2] the epistemic satisfaction relation for a language of communication is given as follows (compare [5]):

- $KS^a(t) \models_G Pos(P(o))$ holds if and only if $o \in \big(O \setminus (P^+_a(t) \cup P^-_a(t))\big)$ and $\lambda^a_{\text{minPos}}(t) \leq \lambda^a(t, P(o)) < \lambda^a_{\text{minBel}}(t)$.
- $KS^a(t) \models_G Bel(P(o))$ holds if and only if $o \in \big(O \setminus (P^+_a(t) \cup P^-_a(t))\big)$ and $\lambda^a_{\text{minBel}}(t) \leq \lambda^a(t, P(o)) < \lambda^a_{\text{maxBel}}(t)$.
- $KS^a(t) \models_G Know(P(o))$ and $KS^a(t) \models_G P(o)$ hold if and only if a condition $o \in P^+(t)$ holds or if conditions $o \in \big(O \setminus (P^+_a(t) \cup P^-_a(t))\big)$ and $\lambda^a(t, P(o)) > \lambda^a_{\text{maxBel}}(t)$ hold.
- $KS^a(t) \models_G Pos(\neg P(o))$ holds if and only if $o \in \big(O \setminus (P^+_a(t) \cup P^-_a(t))\big)$ and $\lambda^a_{\text{minPos}}(t) \leq \lambda^a(t, \neg P(o)) < \lambda^a_{\text{minBel}}(t)$.
- $KS^a(t) \models_G Bel(\neg P(o))$ holds if and only if $o \in \big(O \setminus (P^+_a(t) \cup P^-_a(t))\big)$ and $\lambda^a_{\text{minBel}}(t) \leq \lambda^a(t, \neg P(o)) < \lambda^a_{\text{maxBel}}(t)$.
- $KS^a(t) \models_G Know(\neg P(o))$ and $KS^a(t) \models_G P(o)$ hold if and only if a condition $o \in P^+(t)$ holds or if conditions $o \in \big(O \setminus (P^+_a(t) \cup P^-_a(t))\big)$ and $\lambda^a(t, \neg P(o)) > \lambda^a_{\text{maxBel}}(t)$ hold.

Statements for which the following relation is fulfilled are considered grounded.

4 Alignment of Modality Thresholds

The ultimate goal of the alignment of modality thresholds is the development of consistent models of modal levels of possibility, belief and knowledge that are shared among the interacting agents. The consistency of the models is expressed in terms of the ability of the population to maintain successful communication and the ability to develop a similar, with respect to the shape of external world, set of modal thresholds λ^a_G (See section 3). As such the proposed mechanism

[1] It is assumed that $\lambda_{\text{maxPos}} = \lambda_{\text{minBel}}$.

[2] Every agent is initiated with a random $\lambda^a_G(t_0)$ fulfilling conditions $\lambda^a_{\text{minPos}}(t_0) \in [0, \frac{1}{2}]$, $\lambda^a_{\text{minBel}}(t_0) \in [\frac{1}{2}, \frac{3}{4}]$, and $\lambda^a_{\text{maxBel}}(t_0) \in [\frac{3}{4}, 1]$.

of alignment of modality thresholds should eventually lower the dissimilarity between agents (see Equation 2) and increase the number of successful communications (see Equation 1).

4.1 Language Game Model

In this paper we incorporate a well established pair-wise interaction pattern defined in Language Game Model (LGM) [9]. Originally, LGM focusses on the development of an intuitive and reasonable scheme of communication that would allow a population of interacting agents to emerge a shared lexicon, i.e. reach stable and coherent lexical conventions. In this approach a population of agents, without the direct ability to share their internal representations, engages in a series of pairwise and conventionalized linguistic interactions. In short, at each iteration two agents are randomly selected from the population, one as a speaker and the other as a hearer, and both are exposed to the same state of the external environment, thus establishing a joint attention scene. In such settings the speaker using linguistic clues tries to draw attention of the hearer to the intended topic of its utterance. As a result the hearer is able to relate the linguistic utterance to the current state of the external environment, furthermore applying its internal linguistic processes over multiple episodes allows it to build adequate correlations between linguistic symbols and states of the environment.

In our approach the situation is more complicated, as the linguistic clues are not directly connected to the current state of the external world. Instead, as each uttered modal statement represents agent's individual belief, i.e. internally grounded modal representation, it is only indirectly correlated with the shared state of the external world. As such the traditional form of cross-situational learning mechanism, commonly used in LGM settings, cannot be directly applied and there is a need of more specialised alignment mechanism.

4.2 Alignment Mechanism

As already noted, we assume a pair-wise communication pattern as defined in LGM. In each iteration a randomly selected agent a_S acts as a speaker and expresses its belief concerning the state of a particular property of a given object that is currently outside of its scope of perception. Following the grounding mechanism the speaker utters its current belief (see Section 3). Further, a randomly selected hearer a_H receives the utterance and compares it against its individual belief concerning the state of a particular property of a given object. If this belief is consistent with the utterance, i.e. identical modal levels were grounded by both agents, then the communication is considered as a success. Otherwise, the hearer is being influenced by the speaker and adapts its modal thresholds $\lambda_G^{a_H}$ according to the received utterance. In particular, depending on the hearers internal conformity it modifies its current stance towards the stance reflected by the received modal statement.

Formally the individual modification, i.e. alignment, is defined as a particular adjustment of the internal system of modality thresholds λ_G^a, i.e the

increase/decrease of coordinate related to the modal operator incorporated in the utterance towards the grounding strength associated with the particular topic of this utterance. As noted, the magnitude of the imposed modification depends on the agent's internal conformity γ and the distance between the internal modal threshold and the internal relative grounding strength related to the received utterance. Below, we present the exact formulation of the proposed modification:

Definition 3. *Alignment modification. At a given time point $t_n \in T$ for a given object o, property P, agent a with an internal system of modality thresholds $\lambda^a_G(t_n)$, conformity level γ, and relative grounding strength $\lambda = \lambda^a(t_n, P(o))$ we define alignment modification towards utterance*[3]*:*

Pos(P(o))

$$\lambda^a_{\text{minPos}}(t_{n+1}) = \lambda^a_{\text{minPos}}(t_n) - \gamma * (\lambda^a_{\text{minPos}}(t_n) - \lambda) \quad iif \quad \lambda < \lambda^a_{\text{minPos}}(t_n)$$
$$\lambda^a_{\text{minBel}}(t_{n+1}) = \lambda^a_{\text{minBel}}(t_n) + \gamma * (\lambda^a_{\text{minBel}}(t_n) - \lambda) \quad iif \quad \lambda > \lambda^a_{\text{minBel}}(t_n)$$

Bel(P(o))

$$\lambda^a_{\text{minBel}}(t_{n+1}) = \lambda^a_{\text{minBel}}(t_n) - \gamma * (\lambda^a_{\text{minBel}}(t_n) - \lambda) \quad iif \quad \lambda < \lambda^a_{\text{minBel}}(t_n)$$
$$\lambda^a_{\text{maxBel}}(t_{n+1}) = \lambda^a_{\text{maxBel}}(t_n) + \gamma * (\lambda^a_{\text{maxBel}}(t_n) - \lambda) \quad iif \quad \lambda > \lambda^a_{\text{maxBel}}(t_n)$$

Know(P(o))

$$\lambda^a_{\text{maxBel}}(t_{n+1}) = \lambda^a_{\text{maxBel}}(t_n) - \gamma * (\lambda^a_{\text{maxBel}}(t_n) - \lambda) \quad iif \quad \lambda < \lambda^a_{\text{maxBel}}(t_n)$$

Theorem 1. *On maintaining of the initial boundaries. At a fixed time point t_n, given hearer agent a_H, with a system of modality thresholds $\lambda^{a_H}_G(t_n)$ and conformity level $\gamma \in [0,1]$, and population's system of modality thresholds $\lambda^A_\phi[t_n] = \left[\min_{a\in A}\lambda^a_\phi(t_n), \max_{a\in A}\lambda^a_\phi(t_n)\right]$ where $\phi \in \{minPos, minBel, maxBel\}$. For a random speaker agent a_S uttering one of modal statements $Pos(P(o))$, $Bel(P(o))$, $Know(P(o))$ if $\lambda^{a_S}(t_n, P(o)) = \lambda^{a_H}(t_n, P(o))$, then*

$$\lambda^{a_H}_\phi(t_{n+1}) \in \lambda^A_\phi(t_n).$$

Intuitively the theorem states, that two agents with consistent estimates of the distribution of property exhibition in a given object will not diverge (in terms of modality thresholds) from the general population stance after applying the introduced alignment procedure.

Proof: Since $\lambda^{a_S}(t_n, P(o)) = \lambda^{a_H}(t_n, P(o))$ and boundaries from section 3.2 are assumed, agents a_S, a_H can vary only by one modality level. If for example $KS^{a_H}(t_n) \models_G Pos(P(o))$ and $KS^{a_S}(t_n) \models_G Bel(P(o))$ then

$$\lambda^{a_H}_{\text{minBel}}(t_n) > \lambda^{a_H}(t, P(o)) = \lambda^{a_S}(t, P(o)) > \lambda^{a_S}_{\text{minBel}}(t_n).$$

and $\lambda^{a_H}_{\text{minBel}}(t_{n+1}) \in [\lambda^{a_H}(t, P(o)), \lambda^{a_H}_{\text{minBel}}(t_n)] \subseteq [\lambda^{a_S}_{\text{minBel}}(t_n), \lambda^{a_H}_{\text{minBel}}(t_n)]$. Proof for other pairs follows the same pattern in the end showing that agents' corresponding thresholds can only get closer but cannot overcome each other. ■

[3] Unchanged thresholds are omitted.

4.3 Alignment Process Measures

As already noted, the proposed mechanism of alignment of modality thresholds should lead to the decrease of dissimilarity between agents and lead to the increase of the number of successful communications. In order to formulate the dynamics of the alignment process, we further focus on two basic measures: success rate and the similarity of modality thresholds.

The most obvious characteristics of the communication process is the frequency of successful communications between agents, as it resembles the ability of the system to correctly transfer information from one agent to the other. Whenever the hearer a_H receives a particular utterance G^{a_S} it compares it against its individual belief concerning the state of the mentioned property of the mentioned object. If this belief is consistent with the utterance, i.e. identical modal levels were grounded by both agents, then the communication is consider as a success, otherwise the communication is considered as a failure. In order to keep track of the effectiveness of the communication process itself we introduce the so-called success rate μ_{SR}. In general, the success rate $\mu_{SR(N)}(t_i)$ of order N in time point t_i is the frequency of successful communications in last N interactions ($T_N(t_i)$).

$$\mu_{SR(N)}(t) = \frac{1}{N} \sum_{t_i \in T_N(t)} \mathcal{I}(KS^{a_H(t_i)}(t_i) \vDash_G G^{a_S(t_i)}(t_i)), \tag{1}$$

where $\mathcal{I}$ is the identity function and $G^{a_S(t_i)}(t_i)$ is a message uttered by a speaker agent.

As the frequency of successful communications depends on a particular realisation of the underlying random process, we also incorporate an additional measure that resembles the precise similarity within the entire population. The threshold distance rate μ_{TDR}, defined as the relation of the sum of distances between each pair of agents to the initial sum of distances, reflects the coherence among the individual systems of modality thresholds . In particular, $\mu_{TDR} = 0$ resembles the case of maximum similarity (all individual systems of modality thresholds are equal), $\mu_{TDR} \in (0, 1)$ resembles the relative increase of similarity, whilst $\mu_{TDR} \geq 1$ resembles the relative increase of dissimilarity between the agents.

$$\mu_{TDR}(t) = \frac{1}{2\mu_{TDR}(t_0)} \sum_{a_1 \neq a_2 \in A} d(\lambda_G^{a_1}(t), \lambda_G^{a_2}(t)), \tag{2}$$

where $d(\cdot, \cdot)$ denotes the taxicab metrics, and $\mu_{TDR}(t_0)$ denotes the initial sum of distances between each pair of agents.

5 Simulation

The presented grounding model (see Section 3), enriched with the introduced alignment mechanism (see Section 4) and a population of communicating agents

(see Section 3), was implemented and tested against multiple settings. Due to the limited space we present only the properties of the proposed procedure in case of different population sizes (see Figure 1) and in case of different conformity levels (see Figure 2), followed by a short general analysis of its dynamic behaviour.

Each simulation run is initialised with a fixed and static set of agents with random system of modality thresholds, i.e. uniformly sampled from $(0, 0.5) \times (0.55, 0.8) \times (0.85, 1)$, and a fixed and static set of objects with random characteristics of property exhibition, i.e. uniform over the $(0, 1)$ interval. Presented results represent the general behaviour of the system and are an average over 500 runs with a fixed set of parameter values, i.e. baseline values include a population of 10 agents with a conformity level 0.5, 20 random objects, success rate over 50 consecutive iterations, and overall 5000 iterations.

5.1 Results and Discussion

The increase of the population size results in noticeable increase of iterations required to reach a certain state of alignment (see Figure 1). This intuitive behaviour of the system is a direct consequence of the fact that each iteration influences only a single agent, i.e. the hearer agent, and the more agents are present in the population the lower the chance that a particular agent will be involved in interaction. Additionally, it should be stressed that even a fully cooperative ($\gamma = 1$) and fast learning (from a single interaction) agent would require at least 3 interactions in order to align itself within all assumed three modal levels. Moreover despite the number of agents the system undergoes two stages, early and initial sudden increase of success rate (roughly to $\mu_{SR(50)}(t) = 0.75$ in first 100 iterations), followed by a gradual flattening out towards a slow and nearly stagnant success rate (stabilising at $\mu_{SR(50)}(t) = 0.95$). It should be noted that the variance of presented results is relatively small and as such the presented graphs depict stable behaviour of the alignment procedure.

Similar behaviour can be observed in the dynamics of the threshold distance rate, as the smaller the population size is the faster the decrease of dissimilarity is observed, and opposite, the larger the population is the higher the distance among the individual agents is observed. We can observe that the system undergoes a gradual change of behaviour, similar to the one in case of success rate, from an early sudden decrease of dissimilarity to a nearly flat and linear late alignment change.

The influence of a conformity level γ shown in figure 2 is particularly interesting. A value $\gamma = 0$ represents an agent ignorant of statements uttered by other agents (no learning is present in the system), hence the 'constant' error ($\mu_{SR(50)} \approx 0.67$) and no improvement of a distance ($\mu_{TDR} = 1$). Even by using low non-zero values of γ the aligning procedure results in an increase of success rate and a noticeable decrease of dissimilarity of thresholds.

Increasing values of γ result in a faster growth of success rate (a faster decrease of threshold distance rate, respectively) taking its maximum (minimum) for $\gamma = 1$. A fast increase of the success rate however has its price, as from an interpretational point of view values of γ closing to 1 mean that an agent

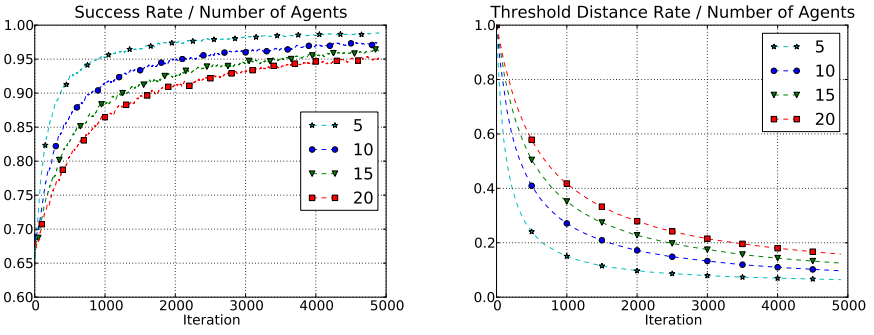


Fig. 1. Success rate (left) and threshold distance rate (right) evolution in number of iterations, each line depicts different population size

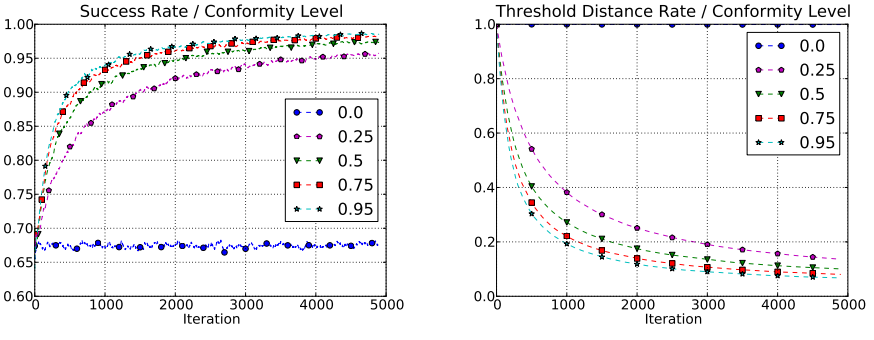


Fig. 2. Success rate (left) and threshold distance rate (right) evolution in number of iterations, each line depicts different conformity level

is highly cooperative and entirely conforms to other agents while disregarding its own stance. A value of $\gamma = 0.5$ results in a behaviour of the system reasonably close to behaviour of population consisting of fully cooperative agents while maintaining an autonomy of individual agents.

6 Summary and Future Work

This paper follows an innovative approach originally developed by Katarzyniak [4,5,6] describing how an individual agent can meaningfully relate its beliefs to its empirical experience. We have extended this model to a case of multiple communicating agents, i.e. capable of exchanging simple and meaningful statements of possibility, belief and knowledge, and proposed a distributed mechanism that allows the society of agents to autonomously develop and adjust their individual

semantics, while maintaining coherent and understandable descriptions. In particular, assuming basic initial constraints and a pair-wise interaction scheme we have shown how a set of agents capable of grounding autoepistemic statements can align itself towards a shared meaning. Based on the presented simulation results we have shown the dynamic behaviour of the system in terms of the frequency of successful communications and relative similarity of the aligned semantics.

This initial research pinpoints a valuable investigation path on how a population of communicating agents is capable of aligning their individual epistemic satisfaction relations towards a coherent description of modal statements of possibility, belief and knowledge. Despite the desired behaviour of the alignment process the proposed approach is only investigated in a limited set of settings and the future research should include investigation on non-homogeneous populations of agents, study the influence of more reliable distributions of property exhibition characteristics and analyse different possible alignment procedures. It should be noted that the early results on the influence of different history limits (internally imposed on the agents) show that it is possible to significantly limit agents demand on memory while maintaining high frequency of successful communications and reaching low dissimilarity rates.

Acknowledgments. This paper was partially supported by Grant no. N N519 444939 funded by Polish Ministry of Science and Higher Education (2010-2013).

References

1. Dennett, D.C.: True believers: The intentional strategy and why it works. In: Stich, Warfield (eds.) Mental Representation: A Reader. Blackwell, Oxford (1994)
2. Eco, U.: La struttura assente (The Absent Structure). Tascabili Bompiani, Milano (1991)
3. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena 42(1-3), 335–346 (1990)
4. Katarzyniak, R.: The language grounding problem and its relation to the internal structure of cognitive agents. Journal of Universal Computer Science 11(2), 357–374 (2005)
5. Katarzyniak, R.: On some properties of grounding simple modalities. Systems Science 31(3), 59–86 (2005)
6. Katarzyniak, R.: On some properties of grounding uniform sets of modal conjunctions. Journal of Intelligent and Fuzzy Systems 17(3), 209–218 (2006)
7. Lorkiewicz, W., Katarzyniak, R.: Recalling the Embodied Meaning of Modal Conjunctions in Artificial Cognitive Agents. In: Nguyen, N.T., Jo, G.-S., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2008. LNCS (LNAI), vol. 4953, pp. 763–772. Springer, Heidelberg (2008)
8. Ogden, C., Richards, I.A.: The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Harcourt Brace Jovanovich, New York (1923)
9. Steels, L.: Language Games for Autonomous Robots. IEEE Intelligent Systems 16(5), 16–22 (2001)
10. Vogt, P.: The physical symbol grounding problem. Cognitive Systems Research 3(3), 429–457 (2002)

Multilateral Negotiations in Distributed, Multi-agent Environment

Piotr Pałka

Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
P.Palka@ia.pw.edu.pl

Abstract. The paper describes the problem of multilateral negotiations, conducted in a distributed environment. In the multilateral negotiation, more than two partners participate simultaneously. Multilateral negotiation processes are important in the case where objects of the negotiation are bounded with by certain constraints (e.g. physical constraints), or if the participants trade with different bundles of commodities. The commodities structure may then force the necessity of multilateral negotiation. The interaction protocol for the multilateral negotiation, based on FIPA's protocol, is presented. A simple, yet representative example is presented to illustrate the problem. The results of computer simulations of the proposed negotiation process are presented.

Keywords: multilateral negotiation, distributed trade, communication protocol, multi-agent system, multi-commodity trade.

1 Introduction

In e-commerce complex markets (e.g. infrastructure markets, where exchange of goods takes place in a complex network structure), there are numerous limitations and other ties between actors – market participants, and between objects of trade – commodities. Frequently occurring constraints arising from a structure of the problem, as well as preferences of participants, cause inability to reach a satisfactory outcome in bilateral negotiations. The solution is either the trade on centralized auctions or exchanges (is there exist one) or the multilaterally negotiated agreement. For certain markets (e.g. for the energy market) the centralized trade exists, where the transmission system operator (TSO), which manages the infrastructure, ensure fulfillment of the associated constraints, and is also the organizer of the trade, exists. However, the predetermined market structure, which allows for the centralized exchange of commodities, often does not exist. Thus, the multilateral negotiations are suitable for exchange of commodities that are not covered by auctions or exchanges.

To implement the negotiation process, the proper interaction protocol should exist. In the literature we can find the simple protocol for multilateral negotiation [3], which is generalization of the protocol for bilateral negotiation [15]. We propose another implementation of the interaction protocol for the multilateral negotiations. We do not assume the simplicity of the proposed protocol,

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 80–89, 2011.

but its effectiveness, correctness and usefulness. The protocol is consistent with FIPA organization specifications [4]. To verify the characteristics presented, we implement the negotiation process in the multi-agent trading platform [7,11], and evaluate it on the simple example of the multilateral trade. Note however, that the simplicity of the example does not preclude the wide application of the protocol in real negotiations.

The contributions of this paper are threefold: (i) development of the interaction protocol for multilateral negotiations, consistent with FIPA's specification (ii), implementation of the multilateral negotiation protocol in the multi-agent platform for multi-commodity exchange, and (iii) evaluation of proposed interaction protocol for multilateral negotiations.

2 Multilateral Negotiations

Participants on the distributed market are directly engaged in exchange of goods, by negotiating the best, from their point of view, contracts. Various participants reach bilateral or multilateral agreements on such markets, usually after complex negotiations. The simplest type of agreement is a bilateral agreement. Note however that in the case of multi-commodity trade (e.g. simultaneous energy and ancillary services trade) the agreements may affect not only many goods, but also a number of suppliers and customers.

Multilateral negotiations are most common in situations where interests of various social groups, companies, organizations, political parties or even states converge, and a resource or resources, which are involved in negotiations, are the common good. An example of the multilateral negotiation concerns establishing limits on greenhouse gases emissions. During the Kyoto conference in 1997 representatives of countries agreed on the need to react on the global warming [10]. In order to ensure the fulfillment of the global constraints, each country declared the level of the CO_2 emission. The problem of water allocation among farmers in Adour basin in southwestern France is described in the work [16]. Since water uptake in the upper reaches of the river affects the flow in its lower course, the negotiation process, which consists in assigning participants to different values of water consumption, and charging them fees, is a process that requires the activity of all participants. In [1] the model for multilateral negotiation, on the example of negotiations between the three entities, is presented. The negotiation concerns the location of a distribution center. Authors apply the non-cooperative multilateral bargaining model developed in [14]. Multilateral negotiations may be also enforced by a simultaneous trade in more than one commodity, so called multi-commodity trade [5]. The multi-commodity trade facilitates compliance with complex preferences of market participants. In the paper [12] authors discuss the problem of multilateral negotiation in an environment with the centralized architecture. The paper considers different mechanisms for communication, however related issues are similar. Author of [3] considers the simple monotonic concession protocol for multilateral negotiation. Author generalizes the bilateral monotonic concession protocol. The monotonic concession protocol was developed in the work [15], it is a generalized version of the Zeuthen Strategy [19].

The same protocol was applied in the work [17], where authors consider the negotiation on the production sequencing in the car factory.

3 Case Study

To illustrate the problem of the multi-commodity trade that requires the multilateral negotiation, we use a fairly simple example. Let us assume the existence of three building plots on the real estate market (see Fig. 1). Entity B is the owner of building plot X, entity C is the owner of parcels Y and Z. Entity B wants to exchange parcel X to Z, because it is consistent with his construction plans. He allows the possibility for selling property X, and purchasing plot Z, but entity C will not agree to sell the Z building plot separately, because he knows that parcel Y is too small. Thus, entity C wants to sell bundle of parcels Y and Z. Entity B does not have enough money to buy parcels Y and Z, besides parcel Y is useless for him. Entity A wants to buy the bundle of X and Y plots, because he is interested in building a house on plot X, and a farm building on parcel Y.

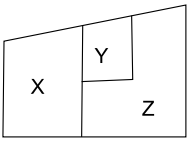


Fig. 1. Plan of building plots, which are subject to negotiations. X, Y, and Z are the names of of individual parcels.

Note, that no pair of entities can achieve acceptable to both parties agreement. However, the agreement between three entities, meets the preferences of all of entities. Thus, in this case the multilateral negotiations are justified by the preferences of all participants.

4 Interaction Protocol for the Multilateral Negotiations

In this section we present the protocol for the multilateral negotiation. Trading processes on different markets, related to negotiation or bidding, are specific for each type of trade. Multi-agent systems [18] are natural framework for automating tasks of the trading processes in the environment, where a number of entities, guided by its own interests, exists, and identify themselves with their own objectives [7]. In the multi-agent system, an agent is a software component which is able to take autonomous actions and conduct negotiations, by submitting offers and exercising sophisticated strategies. However, trading processes can be performed in various circumstances and may take different forms.

4.1 Agents

As we assume that the negotiation is performed using the multi-agent system which models a distributed environment, the information is distributed between particular agents who participate in a trade. The roles structure of the agents considered in the project of multi-agent platform for the multi-commodity exchange [11] is wide. Since the multi-agent platform is a framework, sometimes it appears that the roles structure of the agents should be expanded or clarified. The individual roles are usually closely tied to the application, thus we describe only those roles, which are needed.

Negotiator Agent. The negotiator agent realizes the best contracts from its preferences point of view. Agreement is preceded by (possibly complex) negotiations. Negotiator agent has wide freedom in applying negotiation strategies. It should be able to negotiate multilaterally. A particular negotiator agent, to be able to negotiate, needs either to inform other agents about its negotiation proposals or collect a specific information about other negotiators, their proposals, and particularly about the possibilities for negotiation. We can certainly imagine a case in which each agent broadcasts message about a negotiation proposal to all remaining agents. However, in large systems, such a solution would involve a massive communication effort. Moreover, agents would obtain a huge amount of unnecessary information.

Morris Column Agent. In the centralized system it is possible to create a central repository, which collects all the necessary data about the agents, and existing and planned trade processes. As distributed systems do not have (and often cannot have) any central entity, such repository cannot be built, so there is a need to share the information in a different way. For this purpose we define a special agent called Morris Column agent [5]. The task of this agent is to offer some public location, where other agents may "hang a notice" to report certain information about provided processes of trade. Additionally the Morris Column agent should provide a functionality of searching, filtering and removing the information. In a broader context, agents may leave notices on initiating various types of negotiation processes. Despite apparent compliance with the Morris Column agent, Directory Facilitator agent (provided by FIPA specification) operates in limited way. It accepts queries as a simple strings, interpreted by it as the names of the services registered by other agents, and returns only identifiers of matched agents. The Morris Column agent accepts complex queries, mostly formulated in XPath language (other languages are also allowed, e.g. XQuery), and returns XML formated response. The Morris Column agent provides information not only about agents, but also about other aspects of the trade.

Broker Agent. After the negotiator agents submit to the Morris Column their negotiation proposals, there is a need to associate them. The broker agent associates the matching negotiation proposals, and then notifies the agents (which submitted the proposals) about the possibility of agreement. Broker agent has

certain algorithms for matching proposals. One of it is presented in further parts of the paper. In this paper we assume that the broker agent has limited functionality, nevertheless we do not limit its functionalities to only those activities. In [2] authors consider the negotiation host agent, which has very similar goals as the broker agent. Nevertheless, the negotiation host agent is a central entity in the negotiation process. It creates new negotiator agents and collects proposals. In our proposition, these operations are performed by other agents. Also we assume that more than one broker can exist.

Application of these three agent roles makes it possible to perform multilateral negotiations.

4.2 Communication in Distributed Environment

The Foundation for Intelligent Physical Agents (abbrev. FIPA) [4] developed the interaction protocol for the bilateral negotiations – the *Iterated Contract Net Interaction Protocol* [2]. Unfortunately, this interaction protocol is insufficient for performing the multilateral negotiation. The new interaction protocol for the multilateral negotiation was developed. We based it on the modification of the *Iterated Contract Net Interaction Protocol*, described in [13]. The modification relates to allow the bilateral negotiations in distributed environments. Particular stages of the developed interaction protocol for multilateral negotiations are presented.

Submission of Proposals. A number of negotiating agents submits the negotiation proposals (the *call for proposal* message) to the Morris Column agent (see Fig. 2). We assume the application of the Morris Column agent instead of the Directory Facilitator agent due to the potential complexity of the proposals, which may contain information about many commodities, prices and possible quantities of commodities. Negotiation proposals are written using the XML-based dialect of M^3 model – M3-XML language [11].

Association of Proposals. The broker agent browses familiar Morris Column agents, for the new negotiation proposals (the *query-ref* message – see Fig. 2). After each new proposal received (the *inform* message), it tries to associate them in such a way that agents could potentially reach an agreement. The proper negotiation begins after the number of agents are associated. The simple algorithm for matching offers is presented in section 4.3.

Proper Negotiation. A proper negotiation is started by the current negotiation manager, by sending its proposal to negotiation participants (the *propose* message). If the manager is willing to sell commodities, the proposal includes the value which he wants to receive. If the leader is willing to buy commodities, the proposal includes the value which it is willing to pay. As set forth in [6], we assume that the purchase value is negative.

Every agent, after receiving the proposal, has to decide whether it accepts it. If the agent accepts the received proposal, it will send an *accept-proposal*

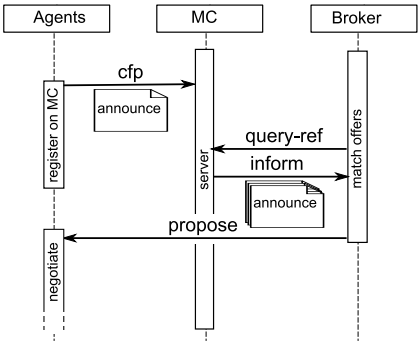


Fig. 2. Preparatory phase of the multilateral negotiation: submission of new proposals and offer matching

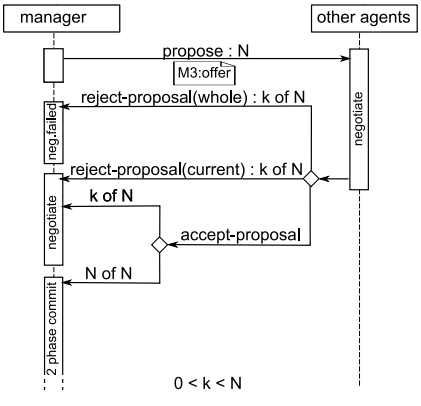


Fig. 3. Multilateral negotiation

message to the manager. If the agent decides that the proposal does not meet its preferences, and it expects a concession from manager, it will send a *reject-proposal(current)* message to the manager. If the agent is no longer interested in the negotiation (e.g. if it reached more profitable agreement in other negotiation process), it will send a *reject-proposal(whole)* message to the manager.

If the manager receives at least one *reject-proposal(whole)* message, it will finish the negotiation, by sending the *inform(finish negotiation)* message to every negotiator. If the leader receives an *accept-proposal* message from all the agents, it will proceed to the two-phase commit protocol, which is the final completion of the agreement. If not, it will inform other agents (*inform(continue negotiation)*) that negotiation should proceed.

Two-Phase Commit. An adaptation of the two-phase commit protocol (2PC), well known in distributed OLTP (On-line Transaction Processing) systems, can

be used as a method for fixing agreements on decentralized markets. It guaranties safe completion of the agreement, ensuring that all the involved parties ultimately accept the agreement. This is especially important in situation where agents are simultaneously involved in the different processes of negotiation, and each of them may reach better agreement in any moment. We assume that every negotiation process is disobliging until the two-phase commit. From the beginning of the two-phase commit protocol agents fix the commodities as contracted. FIPA standards do not directly define this protocol, but it can be quite easily defined on the base of existing communication acts. In [9,13] an appropriate communication schema is proposed for bilateral transactions, nevertheless the 2PC protocol can also be used for multilateral ones.

4.3 Offer Matching Algorithm

The simple offer matching algorithm is presented. This is an internal algorithm of the broker agent. The algorithm creates empty collection of matrices. Each matrix is indexed by the names (unique identifiers) of commodities and participants. If the broker agent finds a new proposal on the Morris Column, the algorithm will add the new matrix to the collection. Also all the matrices, in which there is at least one commodity contained in the newly added offer, are duplicated. The algorithm adds to every newly created or duplicated matrix the information about notifier's preferences – for each notifier the new column is created, where for each commodity +1 is added if the notifier is willing to sell this commodity, or -1 is added if the notifier is willing to buy this commodity (see Fig. 4).

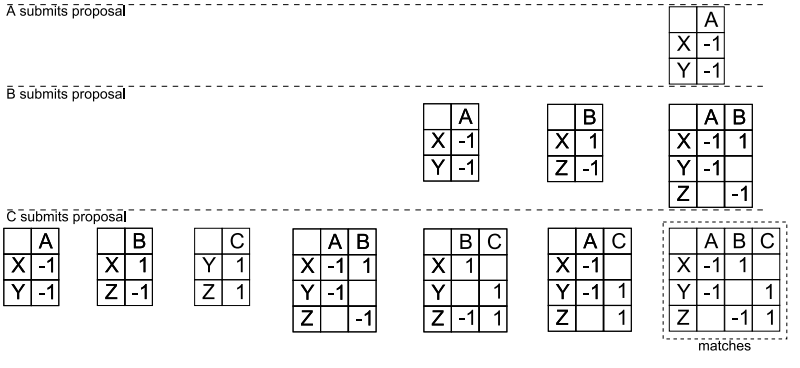


Fig. 4. Offer associating algorithm: accepting new offers

If the algorithm detects that it is possible to make an agreement (i.e. for each commodity, the sum over all participants is equal to zero), the broker agent notifies the corresponding agents. The algorithm does not include the price bids, as it treats them as the subject of further negotiations.

5 Evaluation

The multi-agent platform is implemented in Java. To obtain implementation of FIPA's communication standards it uses the JADE framework. XML-based dialect of multi-commodity market data model M^3 (M3-XML) is used to notate the content of exchanged messages. To facilitate the use of XML based dialect, the JAXB framework, which enables translation of XML objects to Java classes, is applied [8]. The platform is described in details in paper [11]. Using the platform we present the implementation of interaction protocol for the multilateral negotiation, implement the negotiation scenario, and perform tests.

The simulation of the negotiation process was performed. As the example, we use the case study described in section 3. At the beginning of the negotiation, participant A argues that the plots X and Y are worth for him 1 mln $. In fact, he allows to pay for them up to 1.6 mln. $. Participant B wants to receive 200 thous. $ for the exchange the property X on Z. In fact, he allows to pay 100 thous. $ for this transaction. Finally, the participant C, who claimed that he wants to cash in on land sales Y and Z 2.8 mln. $, allows the sell them at 1.5 mln. $. Note, that the transaction can not be completed, taking into account the initial proposals. Thus, each participant concedes its pricing (at each round) of respectively: 100, 50 and 250 thous. $.

Negotiation finished after 18 iterations. During the negotiation process 116 messages have been exchanged. Every negotiating agent sends 37 messages: one message with submission to the Morris Column (*cfp*), 12 messages with proposition (*propose*), 12 messages informing about the result of negotiation process (*inform*), 11 messages rejecting current proposal (*reject-proposal*), and one accepting message (*accept-proposal*). The Morris Column agent sends one message (*inform*) to broker agent, broker agent sends one message to the Morris Column (*query-ref*), and three messages (*propose*) to each negotiating agent.

Agents involved in the negotiation have reached an agreement allowing for the finalization of the transaction. As a result of the agreement participants A and B respectively have agreed to pay 1.5 mln. and 50 thous. $, participant C sold the plots for amount of 1.55 mln. $.

To analyze duration of negotiation process, ten simulations were performed on the computer with Intel Core i7-920 (2.67 GHz) processor, and 8 GB RAM, under the Debian 5.0.3 OS. Duration of the whole negotiation process, taking into account the time from publishing the negotiation proposals to successful finishing the negotiation, and the duration of the relevant negotiation (from the first notification of the proposal) has been measured. The time for single iteration of the negotiation also has been measured (see Table 1).

Simulation has shown, that the process for the multilateral negotiation proceeds fast, since the execution time of single iteration of negotiations associated with the formation of messages, their transmission, analysis and coordination are the order of fractions of a second. In normal commercial practice this delay is of no significance. The main objective of the simulation was to examine the technical issues: ease of implementation, and to verify and confirm that the solutions do not cause performance problems. The effectiveness of the adopted

Table 1. Average, maximal, minimal and standard deviation time of duration of 10 simulations

	avg.	max.	min.	std. dev.
Negotiation process [ms]	3 917.6	4 225.0	3 726.0	149.4
Relevant negotiation process [ms]	933.6	1 223.0	754.0	138.6
Single iteration [ms]	50.8	133.0	30	16.2

multilateral negotiations scenario in the concerned case of a simple but realistic example of the problem of trade, has been confirmed. The correctness of the protocol also has been confirmed on the example. The implementation process of multilateral negotiations in the proposed architecture and using the selected tools was relatively simple. The use of M3-XML dialect allows to easily formulate subsequent negotiating proposals.

6 Summary

The method for reaching agreements using the multilateral negotiations in a distributed environment, using software agents, was proposed. As an interesting trade problems that require multilateral transactions, may affect the multi-commodity trade, the application of M^3 model was also suggested. The paper presents a simple, but realistic case study of a problem whose solution requires the use of the multi-commodity trading mechanism and the multilateral negotiations. Also, results of simulation of the negotiations were presented and analyzed. Of course, the case study described herein relate to a simple example, but it helped to determine the usefulness of the proposed approach: multilateral negotiations in electronic, multi-commodity trade. Nonetheless, despite the simplicity of the example, such situations may occur in the actual trading. In the real multilateral negotiation, the participants' concession strategies, as set forth in [3], should be taken into account. Thus our further work concerns the influence of concession strategies on the properties of the protocol for the multilateral negotiation. Also we will compare our negotiation protocol with the protocols existing in the literature.

Acknowledgments. The research was supported by the Polish National Budget Funds 2009-2011 for science under the grant N N516 3757 36.

References

1. Adams, G., Rausser, G., Simon, L.: Modelling multilateral negotiations: An application to california water policy. Journal of Economic Behavior & Organization 30(1), 97–111 (1996)
2. Bartolini, C., Preist, C., Jennings, N.: A software framework for automated negotiation. In: Choren, R., Garcia, A., Lucena, C., Romanovsky, A. (eds.) SELMAS 2004. LNCS, vol. 3390, pp. 213–235. Springer, Heidelberg (2005)

3. Endriss, U.: Monotonic concession protocols for multilateral negotiation. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2006, pp. 392–399. ACM, New York (2006)
4. Foundation for Intelligent Physical Agents, http://fipa.org/
5. Kacprzak, P., Kaleta, M., Pałka, P., Smolira, K., Toczyłowski, E., Traczyk, T.: Modeling distributed multilateral markets using Multi-commodity Market Model. In: Information Systems Architecture and Technology: Decision Making Models, pp. 15–22. OWPW, Wrocław (2007)
6. Kacprzak, P., Kaleta, M., Pałka, P., Smolira, K., Toczyłowski, E., Traczyk, T.: Communication model for M^3 – Open Multi-commodity Market Data Model. In: 2nd National Scientific Conference on Data Processing Technologies KKNTPD 2007, pp. 139–150 (2007)
7. Kaleta, M., Pałka, P., Toczyłowski, E., Traczyk, T.: Electronic trading on electricity markets within a multi-agent framework. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 788–799. Springer, Heidelberg (2009)
8. McLaughlin, B.: Java and XML data binding. O'Reilly & Associates Inc., Sebastopol (2002)
9. Nimis, J., Lockemann, P.: Robust Multi-Agent Systems The Transactional Conversation Approach. In: SASEMAS 2004 – Third International Conference on Autonomous Agents and Multi-Agent Systems, New York (2004)
10. Oberthur, S., Ott, H.: The Kyoto Protocol: international climate policy for the 21st century. Springer, Heidelberg (1999)
11. Pałka, P., Całka, M., Kaleta, M., Toczyłowski, E., Traczyk, T.: Design and java implementation of the multi-agent platform for multi-commodity exchange. In: 3rd National Scientific Conference on Data Processing Technologies KKNTPD 2010, pp. 184–196. OWPW (2010)
12. Pałka, P., Traczyk, T., Wilk, R.: Centralized multilateral negotiations in multi-commodity trade using the M^3 and ebXML standards (in polish). In: Studia Informatica, vol. 32, pp. 455–466. Wydawnictwo Politechniki Śląskiej, Gliwice (2011)
13. Pałka, P., Kaleta, M., Toczyłowski, E., Traczyk, T.: Use of the FIPA standard for M^3 – open multi-commodity market model (in polish). In: Studia Informatica, vol. 30, pp. 127–140. Wydawnictwo Politechniki Śląskiej, Gliwice (2009)
14. Rausser, G.C., Simon, L.K.: A noncooperative model of collective decision making: A multilateral bargaining approach. Tech. rep., Department of Agricultural & Resource Economics, UC Berkeley (August 1992)
15. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter. MIT Press, Cambridge (1994)
16. Simon, L., Goodhue, R.E., Rausser, G., Thoyer, S., Morardet, S., Rio, P.: Structure and Power in Multilateral Negotiations: An Application to French Water Policy. No. 37630, University of California, Davis, Giannini Foundation (2007)
17. Wooldridge, M., Bussmann, S., Klosterberg, M.: Production sequencing as negotiation. In: Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 1996), pp. 709–726 (1996)
18. Woolridge, M.: Introduction to multiagent systems. John Wiley & Sons, Chichester (2001)
19. Zeuthen, F.: Problems of monopoly and economic warfare. G. Routledge & sons, ltd (1930)

Route Guidance System Based on Self Adaptive Multiagent Algorithm

Mortaza Zolfpour Arokhlo, Ali Selamat, Siti Zaiton Mohd Hashim, and Md Hafiz Selamat

Faculty of Computer Science & Information Systems,
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
Tel.: +6-07-5538009; Fax: +6-07-5565044
zolfpour@gmail.com, {aselamat, sitizaiton, mhafiz}@utm.my

Abstract. Nowadays, self-adaptive multi-agent systems are applied in a variety of areas, including transportation, telecommunications, etc. The main challenge in route guidance system is to direct vehicles to their destination in a dynamic traffic situation, with the aim of reducing the traveling time and to ensure and efficient use of available road networks capacity. In this paper we propose a self-adaptive multi-agent algorithm for managing the shortest path routes that will improve the acceptability of the costs between the origin and destination nodes. The proposed algorithms have been compared with Dijkstra algorithm in order to find the best and shortest paths using a sample of Tehran road network map. Two cases have been tested on the simulation using the proposed algorithm. The experimental results demonstrate that the proposed algorithm could reduce the cost of vehicle routing problem.

Keywords: Multi-agent system (MAS), route guidance system (RGS), Dijkstra algorithm, self-adaptation, self-adaptive multi-agent system (SAMAS).

1 Introduction

Due to the variety of road network environments the corresponding route guidance systems are becoming complex and present many new characteristics such as traffic congestion, time to travel, etc. Therefore the vehicle driver needs to use a route guidance system in order to reduce the traffic delays. Any changes in the traffic conditions can be detected by a self-adaptive multi-agent system (SAMAS) because it has the ability to change its structure and behavior based on the environments where the agents reside [1], [2]. From the literatures, multi-agent system is considered as a problem solving approach that consists of specific and nearly modular agents specialized at solving a particular problem in different type of environments. In a multi-agent system, agents can represent as the cells in a lattice which show the location of the place on the earth or it can represent human behaviors [3].

Nowadays, many complex systems have been studied, solved, managed and controlled by the multi-agent systems [4]. The main challenge in the route guidance systems is to find ways of directing the vehicles to their destination with

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 90–99, 2011.

the aim of reducing traveling time and efficiently use of the available network capacity. In order to avoid traffic congestion and make traveling easier, drivers need traffic direction services. Generally, to solve the problems, the use of self-adaptive multi-agent system seems to be an effective approach to conduct correct path and improve the road network traffic management [5]. Self-adaptive multi-agent systems have successfully been used in the following areas: flood forecast [6], maritime [7], preliminary aircraft design [8] and traffic information and traffic coordination [5]. The shortest paths to the destinations are calculated using the self-adaptive multi-agent system that will suggest to the vehicle drivers in each intersection (or node). When an agent report on the current condition of road traffic on a specific route that has been identified earlier, the vehicle driver moves to the next shortest path priority based on the advised given by SAMAS. In this study, we have not taken into consideration many real factors such as vehicle accidents, special events, weather events, etc. The advantages of using the proposed method is that the vehicle driver is able to dynamically adapt and find the best route with the minimum costs. The proposed algorithm performance is evaluated adaptively by the given routing on Teheran street maps.

The rest of the paper is organized as follows. In section 2, we describe the formulation of self-adaptive multi-agent system. In section 3, the empirical study is presented. The simulation results and experimental comparison are described in Section 4. Finally, the conclusion is discussed in Section 5.

2 Self-Adaptive Multi-Agent System

In this section we discuss a self-adaptive agent system for route guidance system. The formulation of the self-adaptive multi-agent system (SAMAS) is described as follows:

2.1 Problem Formulation

A road directed graph presented by G=(V,A) is a directed dynamic route guidance system based on an electronic map, a set of *'N'* nodes (V) and *'M'* directed edges (A). Each $R_{(s,d)}$ edge is a nonnegative number which stands for the cost while *'s'* is a start node and *'d'* is a finish node connected to the $R_{(s,d)}$. Consider a directed graph, G, which is composed of a set of *'s'* nodes and a set of *'r'* directed edges. Set a number of edges as cost(C) table. Also, G file is filled in data file. If $S = \{s_1, s_2, ..., s_n\}$, $D = \{d_1, d_2, ..., d_n\}$ then $R_{(s,d)} = R_{(s_1,d_1)} + R_{(s_2,d_2)} + ... + R_{(s_n,d_n)}$. $R_{(s,d)}$ consists of the sum of all edge distances (costs) participating in the network path. Therefore, according to the trip origin (s) node and destination (d) node, this issue can be solved as the optimization problem based on real transportation network that is defined as follows:

$$R_{(s,d)} = min \sum_{i=0}^{d} \sum_{j=0}^{d} \alpha R_{(s_i,d_j)} \begin{cases} \alpha = 1 & a\ link\ exists\ between\ i\ and\ j \\ \alpha = 0 & otherwise \end{cases} \quad (1)$$

where:

- i and j are the number of current states stored in the list.
- $R_{(s,d)}$ is the shortest path from a origin node, *'s'* to a last node, *'d'*.
- α is binary digits (0 or 1).
- Links are independent of each other.

Fig. 1. A part of Tehran city road network map

2.2 The Model of Self-Adaptive Multi-Agent System

The model of self-adaptive multi-agent system (SAMAS) is calculated based on Figure 1. The figure shows the road network map of a part of Tehran, the capital city of Iran. A grid network of Figure 1 has been drawn in Figure 2. The Dijkstra algorithm can also be used for finding the shortest path from the origin node to the destination node in a road network graph. The input of this algorithm consists of distances directed between nodes as r and a source data from origin and destination paths as s, d in the network graph. If the distance between the two nodes can be defined as path cost, the total cost of a network graph is the sum of all route distances between nodes in a graph. The proposed algorithm is expressed based on node weight. In this study, a "node weight" is defined as the shortest distance from each node to last network node. If *"n"* is the last node number, *"n-1"* and *"k"* are two node numbers connected to node *"n"* in the network. Also, regarding to equation (1) and node weight definition, the SACCA algorithm reports the procedure is presented in Algorithm 1.

The procedure of Algorithm 1 is calculated in three steps as follows:

- **Step 1.**
  The distance between node *n-1* and node n is equal to the weight of node *n-1*. Also the distance between node k and node n is equal to the weight of node k.

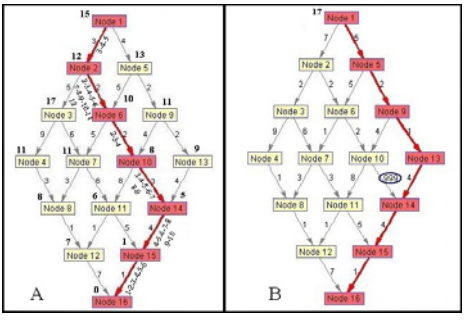


Fig. 2. G (16, 24), the optimal path in ideal (A) and blocked (B) routes status in Case 1

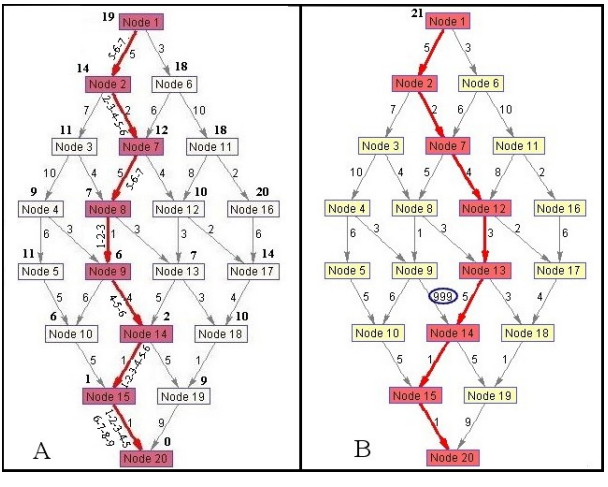


Fig. 3. G (20, 31), the optimal path in ideal (A) and blocked (B) routes status in Case 2

- **Step 2.**
  Similarly, the procedure in Step 1 is continued to calculate the weights of all next nodes until the first node weights is calculated. Finally, the weights of the first node will be used as the minimum distance of paths between the first nodes to last nodes in the network.
- **Step 3.**
  Algorithm 1 calculates the domain of route costs limit for all of the routes that are introduced as the shortest path by using the Dijkstra algorithm. The self-adaptive route information provided by the Dijkstra algorithm will be used in recommending the shortest path of the graph. The suggested costs will be shown to the vehicle driver's so that the shortest path will be used to reach the final destination in a route guidance system.

Algorithm 1. Self-adaptive Cost Computation Algorithm (SACCA)

```
1.  DEFINITIONS:
2.    R(s,d) , a nonnegative number stands for the cost where "s" is start node and "d" is last node.
3.    i, j , k; loop index, G( 1, i) is array of vertexes source; G( 2, i) is array of vertexes destination.
4.    G( 3, i) is array of edge distance(or cost).
5.    for each node and P(i) is array of shortest path from the origin to final node.
6.  INPUT:
7.    G( V, E) is a data file which composed a set of V nodes and set of E directed edges.
8.  OUPUT:
9.    W(i) is a cost data table for all nodes. P(k), a shortest path data table for a graph.
10.   S(j), cost limit s of shortest path routes.
11. INITIALIZATION:
12.   // All nodes from last to first nodes are examined for the routes connected nodes.
13.   // For each edge do operation in two steps as follows:
14.    set W[1... n-1] = 999 , W(n) = 0 , P(i) = 0 ;
15. BEGIN // step 1: Node weight computation
16.     for all nodes // for each node and edge pick the costs in W(k).
17.           for j = first to last edges //j is set to the destination node of edges.
18.                 if (G( 2 , j) = i ) // k is set to the source node of edges.
19.                       W(k) = W(i) + G( 3 , j);
20.                 end if
21.           end for
22.     end for
23.   // step 2: Shortest Path computation
24.     for i = first to last edges
25.           while (the origin (k) is the same in graph, G )
26.                 if (G( 3 , i) = W(k) - W(j)) // j is set to the destination node of edges.
27.                       P(k) = G( 2 , i);
28.                 else
29.                       i = i + 1;
30.                       k = G( 1 , i);
31.                 end if
32.           end while
33.     end for
34.   // step 3: Cost limit computation for self-adaptive
35.     for j = first to last edges // j is set to the destination node of edges.
36.           while (edge belong to p(k) & defined shortest path is true )
37.                       C(j) = G( 3 , j) + 1; // C(j) is cost of j edge.
38.                       S(j) = { G( 3 , j), C(j) }; cost limit s of shortest path routes.
39.           end while
40.     end for
41. END
```

The vehicles drivers need to follow the recommended cost limit by keeping to the mentioned shortest path routes. For example, referring to Figure 3, the cost limit for the route of node 1 to node 2, *5, 6, 7* is based on the shown paths as follows: *Node 1→2 → Node 7 → Node 8 → Node 9 → Node 14 → Node 15 → Node 20.*

3 Empirical Study

In this section, we summarize the experimental results obtained using self-adaptive multi-agent algorithm in Algorithm 1. The information is obtained from the installed agent in each of the nodes in order to acquire the information on the next paths or routes status as shown in Figure 1. Hence, one of the requirements of a dynamic route guidance system is that the real and current information on traveling time for the paths used by the multi-agent system are detected by the magnet sensors, video cameras, GPS, GSM and also other network traffic sensors installed on the routes [9]. In this paper, the traveling cost

and edges distance are assumed as random variables. We have applied the roles of self-adaptive agents in the route guidance system as follows:

- The process of transferring information acquired by the agents through the sensors on the vehicle;
- The responds by agents that are done dynamically to changes in its environment and user requirements.
- The process of monitoring and adapting existing resource to the defined requirement of route guidance system.
- The process of receiving the route requests from the vehicles through the sensor;
- The shortest path information will be sent to the drivers via sensor to other agents;
- The process of responding and monitoring other collaborative agents on the route and the status to alert the incoming vehicles.

3.1 Case 1

Figure 2 presents a transportation network that has 24 routes that create 16 intersections (nodes). Intersections are numbered 1 to 16 so that node 1 is considered as origin and node 16 is road network destination and the values of distance between each of two intersections are generated randomly by numbers between 1 and 10. Also, the distance between node 1 and node 16 (the last node) is 15 that is the weight of node 1. Figure 2 presents the optimal routes in ideal route status as follows: *Node 1 → Node 2 → Node 6 → Node 10 → Node 14 → Node 15 → Node 16.* Figure 2A shows the self-adaptive agent model performance measures of the shortest path routes. Self-adaptive agent system will provide specifications as to what costs exist for a particular route within a given time frame, after which an alternative route must be taken. The SAMAS model uses the received traffic congestion information of shortest path routes from the installed agent in each intersection with acceptable route critical costs suggested by the proposed algorithm in Algorithm 1 (SACCA). For example in first route of Figure 2A, the costs limit of node 1 to node 2, *3,4,5* show that the costs limit encountered that vehicle driver can use the costs of 3,4 and 5 for taking this shortest path. The out of *3,4,5* range which causes are the change in the recommended shortest path. Beyond the cost of 5 the recommended shortest path will be changed. Figure 2B shows the route between node 10 and node 14 which is blocked (let cost = ∞) because of accident or traffic congestion; therefore, some connected routes are removed automatically in the new optimal route computation and the next route which is connected to node 1 is selected as the first choice route in this figure. Thus, considering the above mentioned constraints, the optimal path in the blocked route status (Figure 2B) is: *Node 1 → Node 5 → Node 9 → Node 13 → Node 14 → Node 15 → Node 16.* Lastly, by considering Figure 2B (receiving blocked message from installed agent), the distance between nodes 10→14 tends to ∞ (this road is blocked). Therefore, the previous shortest path in Figure 2B (minimum distance between Node 1 and Node 16) is changed to 17.

3.2 Case 2

Figure 3 presents a transportation network that has 31 routes that create 20 intersections (nodes). The distance between Node 1 and node 20 (last node) is 19 that is the weight of Node 1. Figure 3 presents the optimal routes in ideal route status as follows: *Node 1 → Node 2 → Node 7 → Node 8 → Node 9 → Node 14 → Node 15 → Node 20.* Figure 3A shows the self-adaptive agent performance measures of the shortest path routes. Self-adaptive agent system will provide specifications as to what costs exist for a particular route within a given time frame, after which an alternative route must be taken. The SAMAS model uses the received traffic congestion information of shortest path routes from the installed agent in each intersection with acceptable route critical costs suggested by the proposed algorithm in Algorithm 1 (SACCA). For example, the first route on Figure 3A represents the costs limit of node 1 to node 2, *5,6,7* show that the costs limit encountered that vehicle driver can use the costs of 5,6 and 7 for taking this shortest path. Then, out of *5,6,7* range which causes are the change in the recommended shortest path. Beyond the cost of 7, the recommended shortest path will be changed. Figure 3B shows the distance between nodes $9 \rightarrow 14$ tend to ∞ (this road is blocked). By considering the Figure 3B (receiving blocked route message from installed agent) the minimum cost (or distance) between node 1 and node 20 are changed from 19 to 21. Therefore, regarding to the above mentioned constraints, the optimal path in the blocked route status is: *Node 1 → Node 2 → Node 7 → Node 12 → Node 13 → Node 14 → Node 15 → Node 20.*

4 Simulation Results

In this section, the simulation experiments have been carried out on different network topologies for road networks consisting of 4 to 100 nodes with different edges (between 4 to 200 edges). For example, the results of the proposed algorithm are evaluated by the SACCA algorithm using several cases (Case 1 with 16 nodes and 24 routes, and Case 2 with 100 nodes and 217 routes). Besides, for the current study, we assumed that there are 150 vehicles in the experimental simulation. These vehicles were equipped with Dijkstra algorithm so that the shortest path can be found. The vehicles were divided into two groups, guided and not-guided groups. Fifty of them were the equipped vehicles that were guided through the self-adaptive multi-agent system technique while the remaining 100 were not guided by the self-adaptive multi-agent system technique. Guided equipped vehicles started to travel from the departure intersection and finally exited the network transportation at the destination. They also used the self-adaptive multi-agent systems during the joining on the road network. This group of vehicles was provided with path information from the specified origin to the target destination at any time through the equipment and obeyed installed agent information to follow the route based on self-adaptive algorithm policy [12]. In contrast, the equipped vehicles which belong to the other group (not-guided) only used the route guidance system for finding the shortest path

and did not obey any self-adaptive multi-agent system technique information to be use during their movement from the origin to the destination in the road network. If they wish to change their route direction in any intersection, they only treat it based on self-adaptive algorithm strategy. Table 2 shows the simulation results for the two groups those are assessed by SACCA algorithm using several weighted directed graphs on sparse graphs. However, this problem can be generally more accurate and its development should be studied. The suggested proposed solution will be b riefly discussed in the next Section.

4.1 Evaluations

This section presents the results obtained by using the self-adaptive multi-agent system model in the experimental case comparisons in Table 2.

Case 1: As depicted in Section 3.1, the transportation network graph has 24 routes with 16 nodes (intersections), then the model is simulated for 50 vehicles as group 1 and 100 vehicles as group 2 respectively. Considering Table 2 , the total traveling distance (TD1) of group1 (50 vehicles) is 750km at average travel time (AveTT1) of 1277s. And also group 2 (the 100 remaining vehicles) that did not use the self-adaptive multi-agent system messages, the total traveling distance (TD2) is 1600km at average travel time (AveTT2) of 2133 seconds. Therefore, the second group obtain 865s as the average delay time (or AveDT= 865s) and the saving time is 40.13% (or SveTime = 40.13%).

Table 1. Performance measure for using SAMAS model on road network graphs

Criterion	Description
Group 1	The group which uses both SACCA and Dijkstra algorithms to find the shortest path in the network routing.
Group 2	The group which uses only Dijkstra algorithm to find the shortest path through the network routing.
TD	Traveling distance in kilometers(km).
NV	The number of vehicles that NV_1 is the number of participated vehicles in group 1.
AveSp	Average speed in kilometer per hour (km/h).
AveDT	Average delay time in seconds. $AveDT = AveTT_2 - AveTT_1$
SveTime	Difference between the average traveling time of group 1,$AveTT_1$ and the traveling time of group 2, $AveTT_2$. $SveTime = (AveTT_2 - AveTT_1) / AveTT_2 * 100$
AveTT	Average traveling time in seconds(s).

Table 2. The comparison results of two groups results

(V , E)	Group 1 (Using Dijkstra and SACCA)				Group 2(Using only Dijkstra)				AveDT	SveTime
	TD_1	$AveTT_1$	$AveSp_1$	NV_1	TD_2	$AveTT_2$	$AveSp_2$	NV_2		
G(16,24)	750	1277	42.28	50	1600	2133	27.00	100	856	40.13%
G(20,31)	950	1353	50.55	50	2100	1738	43.50	100	385	22.14%
G(100,197)	2000	1784	80.70	50	7400	3886	68.55	100	2102	54.08%
G(100,217)	2100	1512	100.00	50	5200	3714	50.40	100	2202	59.29%

Case 2: As depicted in Section 3.2, the transportation network graph has 31 routes with 20 nodes (intersections), then the model is simulated for 50 vehicles as group 1 and 100 vehicles as group 2 respectively. Considering Table 2 , the total

traveling distance (TD1) of group 1 (50 vehicles) is 950km at average travel time (AveTT1) of 1353s. And also group 2 (the 100 remaining vehicles) that did not use the self-adaptive multi-agent system messages, the total traveling distance (TD2) is 2100km at average travel time (AveTT2) of 1738 seconds. Therefore, the second group obtain 385s as the average delay time (or AveDT= 385s) and the saving time is 22.14% (or SveTime = 22.14%).

5 Conclusions

This paper attempts to offer a new self-adaptive algorithm based on multi-agent system for finding the critical route costs and it will dynamically respond to the environmental conditions for the stated path. It may change its behavior in the route guidance system based on the given by the algorithm. The performance of the proposed algorithm 1 is evaluated by using the adaptive routing agents where it will dynamically and adaptively find the minimum delay and maximize the saving of time between the routes in the route guidance system (see Table 2). In the simulation results, the SACCA algorithm has been used for managing the shortest path routes for many network graphs with a number of edges, from 4 to 200 edges (like Figure 2 and 3). In each node (intersection), the shortest path (or lowest cost) is suggested by the Dijkstra algorithm. When the agent reports on the traffic congestion in a specific route of the previous shortest path routes to the vehicle drivers, the vehicle drivers will go to the next shortest path.

Acknowledgments. The authors thank Ministry of Higher Education Malaysia (MOHE) under Fundamental Research Grant Scheme (FRGS) Vot 4F031 and Universiti Teknologi Malaysia under the Research University Funding Scheme (Q.J130000.7110.02H47) for supporting the related research.

References

1. Che, K., Li, L.-L., Niu, X.-T., Xing, S.-T.: Research of software development methodology based on self-adaptive multi-agent systems. In: 2009 IEEE International Symposium on IT in Medicine & Education (ITME 2009), vol. 1, pp. 235–240 (2009)
2. Sloman, M., Lupu, E.: Engineering policy-based ubiquitous systems. The Computer Journal, 1113–1127 (2010)
3. Ligtenberg, A.: Exploring the use of multi-agent systems for interactive multi-actor spatial planning, Ph.D. thesis, Wageningen University (2006)
4. Jiang, L., Liu, D.-A.: Advances in multi-agent coordination. In: International Conference on Machine Learning and Cybernetics, pp. 1–7 (2006)
5. Marti, I., Tomas, V.R., Saez, A., Martinez, J.J.: A rule-based multi-agent system for road traffic management. In: International Joint Conference on Web Intelligence and Intelligent Agent Technology, vol. (3) (2009)
6. Georgé, J.-P., Peyruqueou, S., Régis, C., Glize, P.: Experiencing self-adaptive MAS for real-time decision support systems. In: Demazeau, Y., Pavón, J., Corchado, J.M., Bajo, J. (eds.) PAAMS 2009. Advances in Intelligent and Soft Computing, vol. 55, pp. 302–309. Springer, Heidelberg (2009)

7. George, J., Mano, J., Gleizes, M., Morel, M., Bonnot, A., Carreras, D.: Emergent maritime multi-sensor surveillance using an adaptive multi-agent system. In: Cognitive systems with Interactive Sensors (COGIS), Paris (2009)
8. Welcomme, J., Pierre Gleizes, M., Redon, R., Druot, T.: Self-regulating multi-agent system for multidisciplinary optimisation process, In: EUMAS 2006 (2006)
9. Zhang, Z., Xu, J.-M.A: Dynamic route guidance arithmetic based on reinforcement learning. In: International Conference on Machine Learning and Cybernetics, China, pp. 3607–3611 (2005)
10. Zou, L., Xu, J., Zhu, L.: Application of genetic algorithm in dynamic route guidance system. Transportation Systems Engineering and Information Technology, China (7), 45–48 (2007)
11. Shi, X., Xu, J., Xu, Y., Song, J.: A simulation study on agent-network based route guidance system. In: 8th IEEE Intelligent Transportation Systems Conference (ITSC), Vienna, Austria, pp. 263–268 (2005)
12. Khakpour, N., Jalili, S., Talcott, C., Sirjani, M., Mousavi, M.R.: PobSAM: Policy-based managing of actors in self-adaptive systems. Electronic Notes in Theoretical Computer Science, pp. 129–143. Elsevier, Amsterdam (2010)

Agent-Based System with Learning Capabilities for Transport Problems

Bartłomiej Śnieżyński and Jarosław Koźlak

AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics,
Department of Computer Science
Al. Mickiewicza 30, 30-059 Kraków, Poland

Abstract. In this paper we propose an agent architecture with learning capabilities and its application to a transportation problem. The agent consists of the several modules (control, execution, communication, task evaluation, planning and social) and knowledge bases to store information and learned knowledge. The proposed solution is tested on the PDPTW. Agents using supervised and reinforcement learning algorithms generate knowledge to evaluate arriving requests. Experimental results show that learning increases agent performance.

Keywords: agent-based system, machine learning, transport problem.

1 Introduction

The multi–agent and machine learning approaches are useful methods for finding efficient solutions for transportation problems. Transportation problems are based on serving a set of transportation requests using vehicles with the lowest possible costs. The theoretical problem (for instance, VRPTW or PDPTW) assume constant velocities and travel times between locations, but, in practice, transportation problems are characterized by a high degree of dynamism and uncertainty – new requests may come while vehicles are on the move and the delays may occur because of the traffic on the roads.

As a result of the distributed character of the problem, applying the multi–agent approach, where individual agents have a high degree of autonomy in decision making but cooperate in the realization of the requests, may be very useful and should be analyzed. Events may take place according to some regularities, not known to the designer of the decision system a priori. Therefore, the application of machine learning could be useful here.

In this work, we are focusing on solving Pickup and Delivery Problem with Time Windows (PDPTW) [10]. In this problem, the requests are described by the location of pickups and deliveries, time windows (time periods when the loading and unloading of the cargo may be performed) and freight capacity. The vehicles are described by their maximum capacity. Every vehicle has the same velocity and travel times between locations are constant. We can distinguish a static version of the problem, where all requests are known in advance, from a

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 100–109, 2011.

dynamic, where the requests are incoming while the system is functioning and alongside solving the optimization problem, the modeling of vehicle movements and operations is necessary.

The PDPTW may have numerous practical applications such as transport on demand services, transportation of elderly and handicapped people, scheduling for minibus companies, school bus routing and scheduling or shared taxis.

The goal of the presented work is to design and test a hybrid learning agent, applied to the domain of transportation planning and scheduling and an environment where such agents may function. The agent consists of several modules which use different forms of knowledge representation and apply different decision and machine learning algorithms. The current possibilities of the agent will be verified with the use of selected experimental scenarios concerning decisions about accepting requests.

2 Related Research

Because of the high computational complexity of the PDPTW problem (which is an extension of the travelling salesman problem) heuristic approaches are the most widely used methods. Different methods are used: tabu search, evolutionary algorithms, simulated annealing or ant colony approaches.

Apart from the various heuristic approaches, especially important from the point of view of research described in this paper is the application of the multi-agent approaches. The examples of such systems are MARS [5] and Teletruck, which use algorithms and methods such as Contract Net Protocol [14] and Simulated Trading [2], In some cases the machine learning algorithms are applied, some of these solutions are presented below.

The most popular learning technique in multi-agent systems is reinforcement learning, which allows agent to learn its strategy: what action should be executed in a given situation. Other techniques can also be applied: neural networks, models coming from game theory as well as optimization techniques (like the evolutionary approach, tabu search etc.). Good survey of learning in multi-agent systems working in various domains can be found in [12]. Only several examples are described here.

In [17], agents learn coordination rules, which are used in coordination planning. If there is not enough information during learning, agents can communicate additional data during learning. Airiau [1] adds learning capabilities into the BDI model. Decision tree learning is used to support plan applicability testing. Nowaczyk and Malec are also using learning for evaluating plans. Inductive Logic Programming is used to generate knowledge for choosing the best partial plan for an agent [11].

There are also some works carried out in the transport domain. For example, in [7] machine learning was applied to evaluate travel times between locations for solving dynamic transportation problems with uncertain travel times.

In [9] the agents solving the problem VRP used two techniques: a reinforcement learning for a choice of the best sequences of optimization operators and mimetic learning based on exchange of learned solutions between agents.

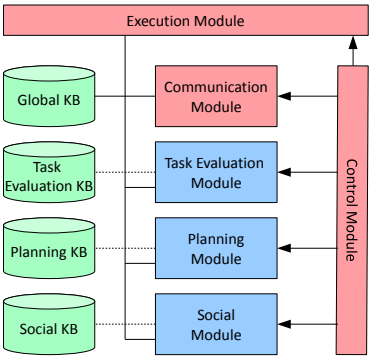


Fig. 1. Agent model

In [6] rule induction is used in a multi-agent solution for the vehicle routing problem. However; in this work learning is done off-line. First, rules are generated by AQ algorithm (the same as used in this work) from traffic data. Next, agents use these rules to predict traffic. Continuation of this work is [16], where on-line learning is applied to generate a routing strategy. Agents use the hybrid learning approach: Q-learning is combined with rule induction. Rules are used to predict traffic on the roads and allow the decrease of space for Q-learning.

3 Model of the System

The model of the system consists of an environment which represents a road network, an agent Dispatcher, responsible for providing transportation requests and a set of agents which represents individual vehicles which manage the travel routes and models the movement of vehicles.

This paper focuses on the presentation of the model of the agent which represents the vehicle. There are several agents of this type in the system and they should cooperate to serve the requests. The model is presented in Fig. 1. The agent consists of modules and knowledge bases. Control module is responsible for managing all the other modules. Communication module exchanges information with other agents. Task evaluation module is used to accept or reject a given transport request. Planning module generates a plan to serve accepted requests. Social module is responsible for modeling the other agents. All modules, except the Control module, have access to the Global knowledge base, where they can store information. The last three modules have their own knowledge bases, where they store knowledge specific to their tasks. The description below concentrates on the following elements: knowledge, performed actions, behavior's and learning processes.

Knowledge. The knowledge of the agent is stored in knowledge sources: global (K_M) and locals. Local knowledge is accessible for processing by individual modules, but global knowledge may be accessible from every module.

The global knowledge base, K_M, contains information from the following domains: State, Map, Requests and Plans.

The *State* contains the following knowledge: current location of the vehicle (P), current travel goal (NT – next target node), current size of the freight (CL – current load), maximal allowed capacity (MC – maximum capacity), information about other vehicles (OV) for example their locations and current travel paths.

Map contains current information about the state of the road network, which is described by a directed graph with weights: N – set of nodes (locations and intersections N_i), E – set of edges representing roads E_i (described by the distance d_i and current travel time t_i).

Requests (*Rec*) is a set of accepted requests Rec_j described by locations of pickup and delivery (Rec_j^{PickUp} and Rec_j^{Deliv}), time windows of pickups (e_j^{PickUp}, l_j^{PickUp}) and delivery (e_j^{Deliv}, l_j^{Deliv}) and needed capacity c_j;

Plan (called *Route*) is a sequence of operations of movements between nodes $Mov(N_{start}, N_{end})$, loadings ($Pickup(req)$) and unloadings ($Delivery(req)$).

One can distinguish the following local knowledge bases: Task Evaluation KB, Planning KB, Social KB.

Task Evaluation KB (K_{TE}) contains a knowledge concerning request acceptation: $(K_M, K_{TE}) \rightarrow \{accept, reject\}$.

Planning KB (K_P) contains knowledge about the constructed routes of travel and loading/unloading operations TR and their evaluations (travel cost TC, travel distance TD, travel time TT, waiting time WT), also specific knowledge about planning (how to check if a plan is good) can be stored here.

K_B also contains particular models of road network Map with information which may be taken into consideration during the planning process: graphs describing the states of the map in subsequent time periods $(t, t+\tau)$: $Map^{(t)} = (N^{(t)}, E^{(t)})$, where for given nodes, information is associated about the distribution of transportation request location in given time intervals $r_i^{(t)}$, and with edges – an average travel times $t_i^{(t)}$ in that interval. Using such models of a network, it is possible to define criteria $crit_j$ which describe the travel which may be assumed in the planning with expected confidence levels:

$$Eval(\{Map^{(j)} : j = \tau \ldots t_{max}\}, crit_j).$$

Another part of planning KB are sets of identified patterns *pat*, which describe dependencies between the states of the traffic on different roads/edges and the changes that occur.

Social KB (K_S) stores information about strategies of communication (frequency of updating of information concerning given subjects j $IU(j)$ and relations between agents (conviction/trust $TR(V_j, Op_k)$ that a given agent V_j will be able to execute a given operation Op_k.

Actions. Below the actions performed by each of the distinguished modules of the agent are described. Actions are described by knowledge which is used and agent state parameters which are changed as a result of their execution.

Task Evaluation Module has only one operation – task evaluation:

$$EvalRequest: (K_{TE}, Route, Req_i, StrongC, WeakC) \rightarrow \{accept(Req_i), reject(Req_i)\},$$

where $StrongC$ represents strong constraints, in our case time window constraints, and $WeakC$ are weak constraints based on the following information:

- change of the route length if Req_i is accepted,
- sum of accepted requests capacities $\sum_{j \in Rec} c_j$,
- current distance to the pick up point (Req_i^{PickUp}),
- smallest distance of all vehicles to the pick up point (Req_i^{PickUp}),
- smallest change of the route length if Req_i is accepted by some vehicle,
- average sum of accepted requests capacities.

Last three are known because of communication with other agents.
Planning Module's operations are:

- plan creation: $CP : (K_M, K_P) \rightarrow Route$;
- request adding: $AR(R_j) : (Route, Reg_j) \rightarrow (Route)$;
- request removal: $RR(R_j) : (Route, Reg_j) \rightarrow (Route)$;
- update plan $UP : (Route, conditions) \rightarrow (Route)$.

Social Module's actions are:

- $SetVehConf$ – describe the confidence to agent vehicle k considering execution of action Op_j: $SetVehConf(value, V_k, Op_j)$: $value \rightarrow TR(V_k, Op_j)$;
- $SetComFreq$ – determine a frequency of communication with subject sub for a given state of the network: $SetComFreq(sub, value) : value \rightarrow IU(sub)$.

Communication Module is a module responsible for an exchange of messages between agents, the messages concern:

- Conditional offer of request: $OR(Request, condition)$. The *condition* identifies other action(s) which the agent is willing to has assigned instead;
- Conditionally accept request if specified conditions are fulfilled: $AR(Request, Condition)$. *Condition* is specified by an agent which declares a conditional willingness of realization of the request. It may concern an acceptation of some requests previously assigned to this agent by other agents or different reasons (for example neither of other agent is able to serve a considered request). In both mentioned actions (OR, AR), the condition may be empty, which means the unconditional offer/acceptance of request
- Send Info ($SI(Info)$) – sending information about state of the world.

Execution Module is responsible for the information concerning physical operations of the vehicle:

- freight loading: $Pickup(Req_i) : (CL) \rightarrow (CL + c_i)$;
- freight unloading: $Delivery(Req_i)$: $(CL) \rightarrow (CL - c_i)$;
- vehicle movement from *old* to *new* location: $Move(N_i^{new}, N_j^{new})$: $(P_i^{old}, NT_i^{old}) \rightarrow (P_j^{new}, NT_j^{new})$.

Behaviors. The actions are executed in the contexts of parallel behaviors:

- evaluation of request (assigning request or conditional acceptance of the request) – negotiations use Contract Net Protocol [14], the request is assigned to the vehicle which is able to realize it with the lowest costs;
- exchange of the requests (the vehicle is trying to get rid of a request assigned to it previously – negotiations may be performed with the use of simulated trading algorithm [2];
- modification of plan (in response to the changing conditions – a change of the state of the agent or change of its model of the world, the agent may change the previous plan. It may be done by activities Conditionally Offer Request (OR) or Conditionally Accept Request (AR);
- learning – the learning may concern several representation elements or decision schemes of the agent, the versions of learning will be described in the subsequent section.

Learning. The learning process concerns the modification of the agent model of the world. The model of the agent considers the following kinds of learning: evaluation of how profitable an acceptance of the given requests is ($L1$), qualification of the points on the map which are worth visiting ($L2$), learning of the characteristics of the road network ($L3$), learning the best frequencies of the propagation of the information about the environment ($L4$) and learning of the behavior of other vehicles ($L5$).

The main goal of this research is to test $L1$ learning. Two learning strategies are tested here: supervised learning and reinforcement learning. In the former, the agent learns to classify decisions as *good* or *bad*. To learn the classifier, it needs labeled examples. Every example consists of the attributes representing $WeakC$ and the decision (*accept*, or *reject*). Example related to Req_i is stored after delivery of the request or if it is too late for delivery. It is labeled as *good* if the request was accepted, delivered and if other requests appear between accepting the Req_i and delivery of Req_i were not rejected, or if Req_i was rejected and accepted by some other agent. Example is labeled as *bad* if the request was delivered but at least two other requests were rejected or Req_i was not delivered by any agent and the agent could accept it ($StrongC$ were fulfilled). In the case of reinforcement learning, the state is defined by variables representing $WeakC$. If the agent accepts the request and delivers it on time, it gets a reward equal to 1. In all other cases the reward is equal to 0.

$L2$. On the basis of the gathered information about the spacial and temporal distribution of request locations $Map^{(t)}$, some conditions *conditions* necessary to be fulfilled by the routes are identified (set by action $UP(route, condition)$, for example points are identified where the road should preferably pass. These are the points characterized by a high density of locations of request points. Some works concerning the modification of vehicle routes to achieve this goal were presented in [8].

$L3$ takes into consideration the history of its travels and of travels of other agents ($Map^{(t)}$). The agents may create patterns *pat* describing changes of the

network characteristic, for examples, dependencies between the changes of travel times between different roads. A model of learning of the current travel times was described in [7].

$L4$. Getting information about the changes of the road network characteristics may be executed at higher or lower frequencies. As a result of learning, the agent gets the preferred frequency of the model of the world update for the given network configuration. This is based on the evaluation of the results obtained for different state of the traffic $Map^{(t)}$ and different $IU(j)$ and how the increase of IU changes the average values from Task Evaluation KB. Some preliminary tests concerning this kind of learning were described in [7].

$L5$ relates to learning of the preferences of other agents/vehicles concerning acceptance of different kinds of requests (for example, depending on the given operation locations or size of the freight), on the basis of it, the agent may determine chances of delegation of a request to other given vehicles.

4 Testing Environment and Performed Experiments

To test the performance of the learning agent, we built a multi-agent system and performed several experiments. The environment is implemented in Java language and makes it possible to solve the PDPTW problem in static and dynamic versions. We started with Acceptance Module learning. Other modules are fixed. The system and results are described below.

Software Used in the Experiments. For the purpose of this research we used a multi-agent system Environment for Learning Agents (ELA), build for testing learning agents in various environments. This system uses Weka [19] for supervised learning, and PIQLE library [4] for reinforcement learning. It provides a general framework, communication facilities and basic classes (such as agent, environment etc.) which can be adapted to create a system for a chosen domain. So far it was used to perform experiments in the Farmer-Pest problem [15], Predator-Prey domain, and PDPTW, described here.

Results of the Experiments. As it was mentioned, the first set of experiments was prepared to test Task evaluation module learning ($L1$). The following supervised learning algorithms were applied: Naïve Bayes classifier, C4.5 [13] (its Weka implementation is called J48), RIPPER [3] (JRip implementation). Also Watkins's $Q(\lambda)$ [18] reinforcement learning algorithm was used (shorter name "Watkins" will be used). It is a modification of well known Q-Learning algorithm [18]. As a reference, greedy algorithm was used. In this setting the first agent for which $StrongC$ are satisfied accepts the request.

Every experiment consists of 100 sequences of 10 (for the first experiment) or 20 simulations. Agents' knowledge was kept between simulations, but it was cleared between sequences. All vehicle agents used the same algorithm. There were as many repetitions as many algorithms tested. Figures present the averages from sequences. Supervised learning algorithms were executed between

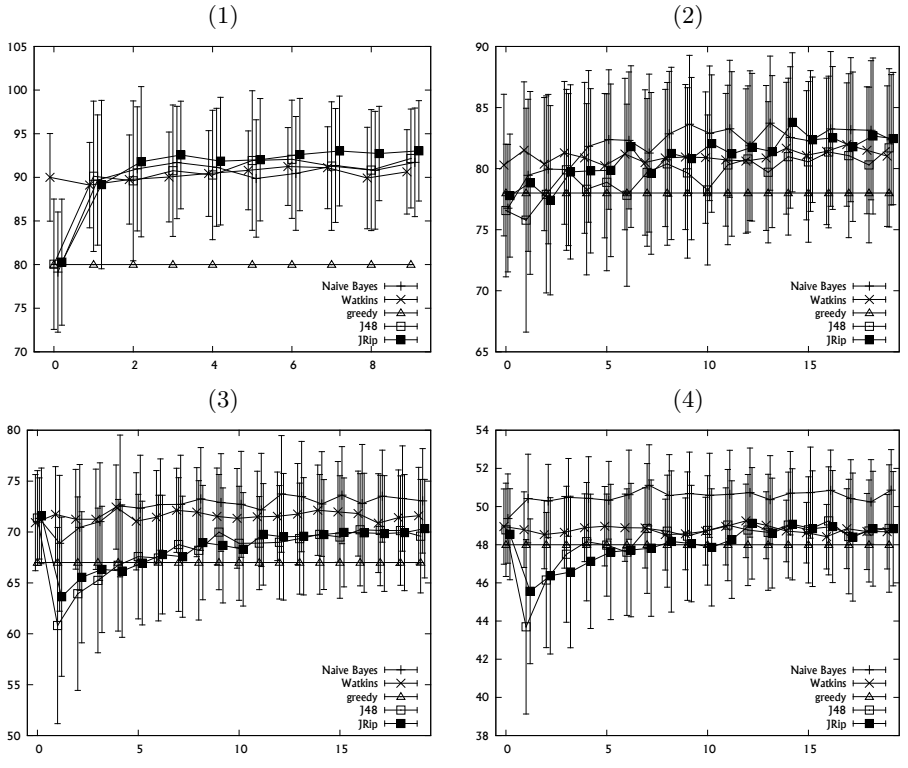


Fig. 2. Performance of agents using various task evaluation learning algorithms measured as a number of requests served (axis Y) in the sequence of simulations (axis X) for experiments 1-4

simulations. Acceptance actions were chosen according to the Boltzmann strategy [18] with $\tau = 0.2$ to provide appropriate exploration of possibilities. Default parameters of the algorithms were used. Watkins algorithm was also executed between simulations with $\lambda = 0.4$, $\gamma = 0.9$, and $\alpha = 0.8$. Exploration strategy was ϵ-greedy with constant $\epsilon = 0.1$ (which gave better results than Boltzmann strategy in initial experiments).

We tested the mentioned learning algorithms in several situations, with various numbers of vehicles, various capacities and speeds, various numbers, frequencies and space distributions of requests. In the first experiment, there are 2 vehicles with capacity 1, and speed 10 which should distribute 90 packages. There were two clusters of places on the ma, called A and B. Source and destination places were in the same folder. There were 80 requests in a cluster A, appearing every 5 time steps and 10 requests in cluster B appearing every 15 steps. Time windows in B were 5 times larger than in A. Results representing the number of requests served are shown in Fig. 2-(1). In the second experiment there were 4 vehicles with speeds 12 and capacity 1. Requests were appearing in two clusters every 5 turns, there were 100 steps for pickup and 500 for delivery. The learning

process is slower, therefore there are 20 simulations in every sequence. Results are presented in Fig. 2-(2). Configuration of the third experiment was almost the same as above, the only difference was that the vehicle speed limit decreased to 10. Results are presented in Fig. 2-(3). The last, fourth experiment, there were 9 vehicles with a capacity of 2 and speed 15. There were 90 requests of size 1 appearing in every one of the three clusters, every step. Times for the pickup and the delivery were 50 and 200. Results are presented in Fig. 2-(4).

Discussion of Results. For all experiments, we can observe that the learning algorithm has better results than greedy. t-Student at the final simulation confirms that they are significantly better at $p<0.05$.

All agents using supervised learning algorithms improved throughout time. In experiments 3 and 4 we can observe initial fall of J48 and JRip performances, because not enough experience is initially collected. Next, the performance increases rapidly at the beginning because a lot of significant examples are stored, which results in the generation of new knowledge. After several simulations, performance becomes stable because examples do not contain any new information.

Watkins algorithm behaves more stable, it does not improve. Probably a better parameter setting should be used here. Also in the first experiment it produces very good results at the beginning. The reason may be in another exploration strategy. However, this needs further investigation.

In the first experiment the best results are archived by agents using JRip. However, the difference is not statistically significant. In experiments 3 and 4 the best is Naïve Bayes and the difference is significant at $p<0.05$. These environments are difficult and their stochastic character seems to be responsible for good performance of Naïve Bayes.

5 Conclusion

As a result of the work the model of learning agent for solving transport problems was designed and applied. The environment for modeling transportation problems was implemented together with the modules necessary for applying different machine learning techniques. Experimental results show that machine learning algorithms allow the agents to update and correct their knowledge which results in the development of better plans in the partially unknown environment.

Future works will mainly concern the integration and evaluation of all learning agent modules. Also performance of the system in environments with changing characteristics should be evaluated.

In this paper, the solving of classical PDPTW problem is presented; however, the presented infrastructure is designed for solving in the future more complex problems, related to: solving soft time windows and variable travel times between locations.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education grant no. N N516 366236. The authors thank students of Computer Science from AGH–UST, especially M. Mlostek and M. Pulchny, for

the participation in the development of simulation environment and realization of the experiments.

References

1. Airiau, S., Padham, L., Sardina, S., Sen, S.: Incorporating learning in bdi agents. In: Proceedings of the ALAMAS+ALAg Workshop (May 2008)
2. Bachem, A., Hochstättler, W., Malich, M.: The simulated trading heuristic for solving vehicle routing problems. Discrete Appl. Math. 65, 47–72 (1996)
3. Cohen, W.: Fast effective rule induction. In: Machine Learning: Proc. of the Twelfth International Conference. Morgan Kaufmann, Lake Tahoe (1995)
4. Comité, F.D.: A java platform for reinforcement learning experiments (2005)
5. Fischer, K., Muller, J., Pischel, M.: Cooperative Transportation Scheduling: an Application Domain for DAI. Applied Artificial Intelligence, 1–33 (1996)
6. Gehrke, J.D., Wojtusiak, J.: Traffic prediction for agent route planning. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part III. LNCS, vol. 5103, pp. 692–701. Springer, Heidelberg (2008)
7. Koźlak, J.: Learning in cooperating agents environment as a method of solving transport problems and limiting the effects of crisis situations. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4488, pp. 872–879. Springer, Heidelberg (2007)
8. Koźlak, J., Créput, J.C., Hilaire, V., Koukam, A.: Multi-agent approach to dynamic pick-up and delivery problem with uncertain knowledge about future transport demands. Fundam. Inform.: Annales Societatis Mathematicae Polonae 71(1) (2006)
9. Meignan, D., Creeput, J.C., Koukam, A.: A coalition-based metaheuristic for the vehicle routing problem. In: Proceedings of IEEE Congress on Evolutionary Computation, Hongkong, pp. 1176–1182 (2008)
10. Mitrowic-Minic, S.: Pickup and Delivery Problem with Time Windows: A Survey. Tech. Rep. 1998-12, SFU CMPT TR (1998)
11. Nowaczyk, S., Malec, J.: Learning to evaluate conditional partial plans. In: ICMLA 2007: Proceedings of the Sixth International Conference on Machine Learning and Applications, pp. 235–240. IEEE Computer Society, Washington, DC, USA (2007)
12. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems 11 (2005)
13. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
14. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computer (December 1980)
15. Śnieżyński, B., Dajda, J., Mlostek, M., Pulchny, M.: Farmer-pest problem: A multidimensional problem domain for comparison of agent learning methods. In: Proc. of the ICCS 2011. Elsevier, Amsterdam (2011)
16. Śnieżyński, B., Wojcik, W., Gehrke, J.D., Wojtusiak, J.: Combining rule induction and reinforcement learning: An agent-based vehicle routing. In: Proc. of the ICMLA 2010, Washington D.C, pp. 851–856 (2010)
17. Sugawara, T., Lesser, V.: On-line learning of coordination plans. In: Proceedings of the 12th International Workshop on Distributed Artificial Intelligence (1993)
18. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (March 1998)
19. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (1999)

Modelling of Agents Cooperation and Negotiation

František Čapkovič

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovak Republic
Frantisek.Capkovic@savba.sk
http://www.ui.sav.sk/home/capkovic/capkhome.htm

Abstract. Three possibilities of modelling the agents cooperation and negotiation by means of place/transition Petri nets (P/T PN) are introduced and analyzed in this paper. Namely, the cooperation and negotiation by means of P/T PN transitions(modelling the occurrence of discrete events), by means of P/T PN places (modelling the states of the interface which is often represented by a supervisor) and by means of their combination are concerned. The particular autonomous agents are represented by modules modelled by means of P/T PN subnets. The first kind of cooperation and negotiation is realized by the mutual communication of agents on way of the information exchange between the particular states of the communicating agents through the interface realized by Petri net (PN) transitions. The second kind is realized by the mutual communication of agents on way of the mediate states of the interface represented by PN places. The combined approach uses the communication of the agents through PN transitions and places.

Keywords: Agents, communication, cooperation, discrete-event systems, negotiation, place/transition Petri nets, supervisor.

1 Introduction and Preliminaries

The P/T PN [11] are used here in the process of modelling the agent cooperation and negotiation. On that way the earlier results concerning the agents cooperation and negotiation [1,2,3,4,5,6,12,13] can be utilized as well as the theory of supervision [8,9,10]. As to the structure P/T PN are bipartite directed graphs $< P, T, F, G >$ with P, T, F, G being, respectively, the set of places, the set of transitions, the set of directed arcs from places to transitions and the set of directed arcs from transitions to places. Here, $P \cap T = \emptyset$, $F \cap G = \emptyset$. Moreover, P/T PN have their dynamics $< X, U, \delta, \mathbf{x}_0 >$ with X, U, δ, $\mathbf{x}_0$ being, respectively, the set of states (marking the places), the set of discrete events (states of transitions), the transition function and the initial state vector. Here, $X \cap U = \emptyset$, $\delta : X \times U \rightarrow X$. The formal expression of δ can be rewritten into the system form as follows

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}.\mathbf{u}_k \quad , \quad k = 0, ..., N \tag{1}$$

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 110–119, 2011.

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F} \tag{2}$$

$$\mathbf{F}.\mathbf{u}_k \leq \mathbf{x}_k \tag{3}$$

where k is the discrete step of the dynamics development; $\mathbf{x}_k = (\sigma^k_{p_1}, ..., \sigma^k_{p_n})^T$ is the n-dimensional state vector; $\sigma^k_{p_i} \in \{0, 1, ..., c_{p_i}\}$, $i = 1, ..., n$ express the states of atomic activities by 0 (passivity) or by $0 < \sigma_{p_i} \leq c_{p_i}$ (activity); c_{p_i} is the capacity of p_i; $\mathbf{u}_k = (\gamma^k_{t_1}, ..., \gamma^k_{t_m})^T$ is the m-dimensional control vector; its components $\gamma^k_{t_j} \in \{0, 1\}$, $j = 1, ..., m$ represent occurring of elementary discrete events (e.g. starting or ending the activities, failures, etc.) by 1 (presence of the corresponding discrete event) or by 0 (absence of the event); $\mathbf{B}, \mathbf{F}, \mathbf{G}$ are matrices of integers; $\mathbf{F} = \{f_{ij}\}_{n \times m}$, $f_{ij} \in \{0, M_{f_{ij}}\}$, expresses the causal relations among the states (as causes) and the discrete events occurring during the DES (discrete-event systems) operation (as consequences) by 0 (nonexistence of the relation) or by $M_{f_{ij}} > 0$ (existence and multiplicity of the relation); $\mathbf{G} = \{g_{ij}\}_{m \times n}$, $g_{ij} \in \{0, M_{g_{ij}}\}$, expresses analogically the causal relations among the discrete events (as causes) and the DES states (as consequences); $\mathbf{B}$ is given according to (2); $(.)^T$ symbolizes the matrix or vector transposition. Just such an exact mathematical expression of P/T PN, in contrast to high-level PN, yield the possibility to deal with the PN models in analytical terms.

2 Modular Structure of PN Models of Agents

Having modules (PN models of particular agents) we can think about building a suitable global structure from such modules. In case of N_A autonomous agents the structural matrices of the global model can have the following form

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{F}_{c_1} \\ \mathbf{0} & \mathbf{F}_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{F}_{c_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F}_{N_A-1} & \mathbf{0} & \mathbf{F}_{c_{N_A-1}} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{F}_{N_A} & \mathbf{F}_{c_{N_A}} \end{pmatrix}; \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{G}_{N_A-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{G}_{N_A} \\ \hline \mathbf{G}_{c_1} & \mathbf{G}_{c_2} & \dots & \mathbf{G}_{c_{N_A-1}} & \mathbf{G}_{c_{N_A}} \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{c_1} \\ \mathbf{0} & \mathbf{B}_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{c_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_{N_A-1} & \mathbf{0} & \mathbf{B}_{c_{N_A-1}} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}_{N_A} & \mathbf{B}_{c_{N_A}} \end{pmatrix} = \left(\text{blockdiag}(\mathbf{B}_i)_{i=1,N_A} \mid \mathbf{B}_c \right)$$

where $\mathbf{B}_i = \mathbf{G}_i^T - \mathbf{F}_i$; $\mathbf{B}_{c_i} = \mathbf{G}_{c_i}^T - \mathbf{F}_{c_i}$; $i = 1, ..., N_A$; $\mathbf{F}_c = (\mathbf{F}_{c_1}^T, \mathbf{F}_{c_2}^T, ..., \mathbf{F}_{c_{N_A}}^T)^T$ $\mathbf{G}_c = (\mathbf{G}_{c_1}, \mathbf{G}_{c_2}, ..., \mathbf{G}_{c_{N_A}})$; $\mathbf{B}_c = (\mathbf{B}_{c_1}^T, \mathbf{B}_{c_2}^T, ..., \mathbf{B}_{c_{N_A}}^T)^T$. Here, $\mathbf{F}_i$, $\mathbf{G}_i$, $\mathbf{B}_i$ represent the parameters of the PN-based model of the agent A_i. $\mathbf{F}_c$, $\mathbf{G}_c$, $\mathbf{B}_c$ represent the structure of the interface between the cooperating agents. It means that

it can have a form of a PN subnet too. This interface consists (exclusively) of additional PN transitions, i.e. it represent the first kind of the agents cooperation. Analogically, in case of the interface consisting (exclusively) of the additional PN places (the second kind of the agents cooperation) the structural matrices are as follows

$$\mathbf{F} = \begin{pmatrix} \text{blockdiag}(\mathbf{F}_i)_{i=1,N_A} \\ \hline \mathbf{F}_d \end{pmatrix};\ \mathbf{G} = \left(\text{blockdiag}(\mathbf{G}_i)_{i=1,N_A} \mid \mathbf{G}_d \right)$$

$$\mathbf{B} = \begin{pmatrix} \text{blockdiag}(\mathbf{B}_i)_{i=1,N_A} \\ \hline \mathbf{B}_d \end{pmatrix}$$

where $\mathbf{B}_i = \mathbf{G}_i^T - \mathbf{F}_i$; $\mathbf{B}_{d_i} = \mathbf{G}_{d_i}^T - \mathbf{F}_{d_i}$; $i = 1, ..., N_A$; $\mathbf{F}_d = (\mathbf{F}_{d_1}, \mathbf{F}_{d_2}, ..., \mathbf{F}_{d_{N_A}})$; $\mathbf{G}_d = (\mathbf{G}_{d_1}^T, \mathbf{G}_{d_2}^T, ..., \mathbf{G}_{d_{N_A}}^T)^T$; $\mathbf{B}_d = (\mathbf{B}_{d_1}, \mathbf{B}_{d_2}, ..., \mathbf{B}_{d_{N_A}})$. $\mathbf{F}_i$, $\mathbf{G}_i$, $\mathbf{B}_i$, $i = 1, ..., N_A$, represent the parameters of the PN-based model of the agent A_i, and $\mathbf{F}_d$, $\mathbf{G}_d$, $\mathbf{B}_d$ represent the structure of the interface between the cooperating agents which can also be a PN subnet.

Finally, combining both previous models we can obtain the kernel of interface in the form of the PN subnet (another agent or even an agent system) containing n_d additional places and m_c additional transitions. Its structure is given by the $(n_d \times m_c)$-dimensional matrix $\mathbf{F}_{d\leftrightarrow c}$ and $(m_c \times n_d)$-dimensional matrix $\mathbf{G}_{c\leftrightarrow d}$. The row and the column consisting of corresponding blocks model the contacts of the kernel with the autonomous agents. Hence,

$$\mathbf{F} = \begin{pmatrix} \text{blockdiag}(\mathbf{F}_i)_{i=1,N_A} & \mid & \mathbf{F}_c \\ \hline \mathbf{F}_d & \mid & \hline \mathbf{F}_{d\leftrightarrow c} \end{pmatrix};\ \mathbf{G} = \begin{pmatrix} \text{blockdiag}(\mathbf{G}_i)_{i=1,N_A} & \mid & \mathbf{G}_d \\ \hline \mathbf{G}_c & \mid & \hline \mathbf{G}_{c\leftrightarrow d} \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} \text{blockdiag}(\mathbf{B}_i)_{i=1,N_A} & \mid & \mathbf{B}_c \\ \hline \mathbf{B}_d & \mid & \hline \mathbf{B}_{d\leftrightarrow c} \end{pmatrix}$$

where $\mathbf{B}_i = \mathbf{G}_i^T - \mathbf{F}_i$; $\mathbf{B}_{d_i} = \mathbf{G}_{d_i}^T - \mathbf{F}_{d_i}$; $\mathbf{B}_{c_i} = \mathbf{G}_{c_i}^T - \mathbf{F}_{c_i}$; $i = 1, ..., N_A$; $\mathbf{B}_{d\leftrightarrow c} = \mathbf{G}_{c\leftrightarrow d}^T - \mathbf{F}_{d\leftrightarrow c}$. $\mathbf{F}$, $\mathbf{G}$, $\mathbf{B}$ acquire a special structure. Each of them has the big diagonal block describing the structure of autonomous agents and the specific part in the form of the letter L turned over the vertical axe. $\mathbf{F}_{d\leftrightarrow c}$, $\mathbf{G}_{c\leftrightarrow d}$, $\mathbf{B}_{d\leftrightarrow c}$ are situated, respectively, on their diagonals just in the breakage of the turned L.

3 Three Forms of Modelling the Agents Communication

There can be defined three forms of modelling the communication as follows: (i) the communication PN modules through transitions; (ii) the communication

PN modules through places; (iii) the communication PN modules through PN transitions and places simultaneously. To illustrate the particular paradigms of building the structures of agents communication mentioned above, let us introduce the following examples.

3.1 Communication of Agent Models through the PN Transitions

Let us analyze the first form of the agent communication on the concrete example when three autonomouos agents communicate through three communication channels.

Example 1. Consider three agents (e.g. intelligent robots) A_1, A_2, A_3 given in Fig. 1. Each of them has the same structure. The sets of their PN models places are $P_{A_1} = \{p_1, p_2, p_3\}$, $P_{A_2} = \{p_4, p_5, p_6\}$, $P_{A_3} = \{p_7, p_8, p_9\}$, while the sets of their PN models transitions are $T_{A_1} = \{t_1, t_2, t_3, t_4\}$, $T_{A_2} = \{t_5, t_6, t_7, t_8\}$, $T_{A_3} = \{t_9, t_{10}, t_{11}, t_{12}\}$. The places represents three basic states of the agents - the particular agent is either available (p_2, p_5, p_8) or it wants to communicate (p_3, p_6, p_9) or it does not want to communicate (p_1, p_4, p_7). Also the communication channels between the corresponding two agents have the same structure - the PN model of the channel Ch_1 between A_1 and A_2 consists of $\{p_{10}, p_{11}\}$, $\{t_{13}, t_{14}, t_{15}, t_{16}\}$, the channel Ch_2 between A_1 and A_3 consists of $\{p_{12}, p_{13}\}$, $\{t_{17}, t_{18}, t_{19}, t_{20}\}$, and the channel Ch_3 between A_2 and A_3 consists of $\{p_{14}, p_{15}\}$, $\{t_{21}, t_{22}, t_{23}, t_{24}\}$. The states of the channels are: available (p_{11}, p_{13}, p_{15}) and realizing the communication of corresponding agents (p_{10}, p_{12}, p_{14}). The mutual communication of the agents through the channels is realized by means of the interface transitions. As we can see, the channels create the interface between the communicating agents. Moreover, they can also be understood to be the agents. The autonomous agents have the same structure given as follows

$$\mathbf{F}_{A_i} = \begin{pmatrix} 0\,1\,0\,0 \\ 1\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix};\ \mathbf{G}^T_{A_i} = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix};\ i = 1, 2, 3$$

It means

$$\mathbf{F}_A = \begin{pmatrix} \mathbf{F}_{A_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{A_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_3} \end{pmatrix};\ \mathbf{G}^T_A = \begin{pmatrix} \mathbf{G}^T_{A_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T_{A_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}^T_{A_3} \end{pmatrix}$$

The communication channels between the particular agents also have the same structure

$$\mathbf{F}_{Ch_i} = \begin{pmatrix} 0\,0\,1\,0 \\ 1\,1\,0\,1 \end{pmatrix};\ \mathbf{G}^T_{Ch_i} = \begin{pmatrix} 1\,1\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix};\ i = 1, 2, 3$$

It means

$$\mathbf{F}_{Ch} = \begin{pmatrix} \mathbf{F}_{Ch_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{Ch_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{Ch_3} \end{pmatrix};\ \mathbf{G}^T_{Ch} = \begin{pmatrix} \mathbf{G}^T_{Ch_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T_{Ch_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}^T_{Ch_3} \end{pmatrix}$$

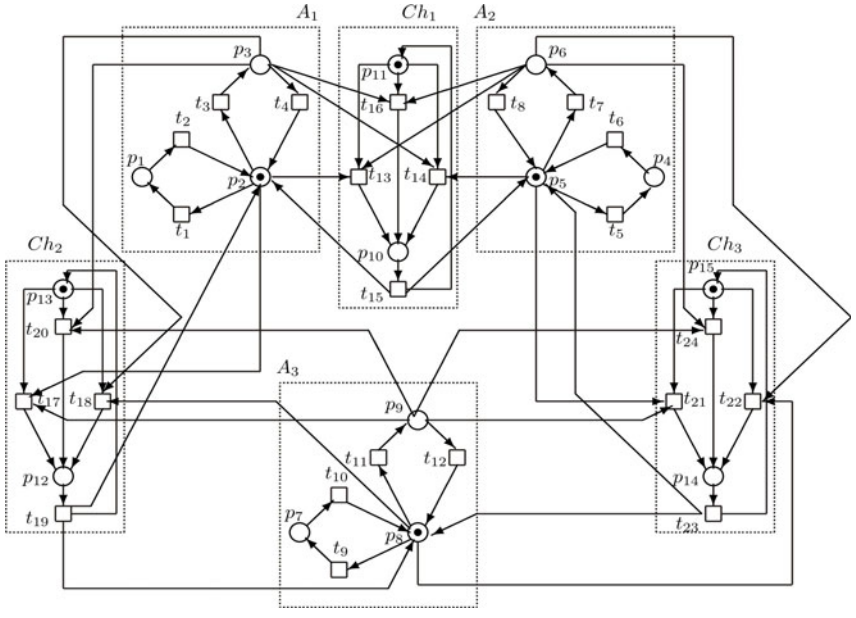


Fig. 1. The communication of three agents A_1, A_2, A_3

The agent interconnections are expressed as follows

$$\mathbf{F}_c = \left(\begin{array}{c|c|c} 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 1\,0\,0\,0 & 1\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,1\,0\,1 & 0\,1\,0\,1 & 0\,0\,0\,0 \\ \hline 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,1\,0\,0 & 0\,0\,0\,0 & 1\,0\,0\,0 \\ 1\,0\,0\,1 & 0\,0\,0\,0 & 0\,1\,0\,1 \\ \hline 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,0\,0\,0 & 0\,1\,0\,0 & 0\,1\,0\,0 \\ 0\,0\,0\,0 & 1\,0\,0\,1 & 1\,0\,0\,1 \end{array}\right); \quad \mathbf{G}_c^T = \left(\begin{array}{c|c|c} 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,0\,1\,0 & 0\,0\,1\,0 & 0\,0\,0\,0 \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ \hline 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,0\,1\,0 & 0\,0\,0\,0 & 0\,0\,1\,0 \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ \hline 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,0\,0\,0 & 0\,0\,1\,0 & 0\,0\,1\,0 \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \end{array}\right)$$

The supervised system is described by the following structural matrices

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_A & \mathbf{F}_c \\ \mathbf{0} & \mathbf{F}_{Ch} \end{pmatrix}; \; \mathbf{G}^T = \begin{pmatrix} \mathbf{G}_A^T & \mathbf{G}_c^T \\ \mathbf{0} & \mathbf{G}_{Ch}^T \end{pmatrix}$$

Starting from the initial state $\mathbf{x}_0 = (\mathbf{x}_{A_1}^T 0, \mathbf{x}_{A_2}^T 0, \mathbf{x}_{A_3}^T 0, \mathbf{x}_{Ch_1}^T 0, \mathbf{x}_{Ch_2}^T 0, \mathbf{x}_{Ch_3}^T 0)^T$, where $\mathbf{x}_{A_i}^T 0 = (0,\, 1,\, 0)^T$, $\mathbf{x}_{Ch_i}^T 0 = (0,\, 1)^T$, $i = 1,\, 2,\, 3$, we obtain the reachability

graph with 36 nodes. It represents the space of feasible states reachable form the initial state $\mathbf{x}_0$. These states are given as the columns of the matrix

$$\mathbf{X}_{reach} = \begin{pmatrix}
0&1&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
1&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&1&0&0\\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&1&0&0&0&0&0&0\\
1&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&1&0&0&0&1&1&0&0&0&0&1&1&1&0&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&1&0&0&1&0&1&0&0&1&0&0&1&0&0&0&1&0\\
1&1&1&1&1&0&0&1&1&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&1&0&0&1&0&1&0&0&1&0&0&1&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&0&1&1&1&1&1&1&0&1&1&0&1
\end{pmatrix}$$

3.2 Communication of Agent Models through the PN Places

Here, let us analyze the second form of the agent communication on the concrete example when five autonomous robots communicate through PN places.

Example 2. Consider five autonomous agents A_i, $i = 1, \ldots, 5$ (e.g. intelligent robots) with the same structure. The structure of the single agent having two states - 'idle' and 'working' - is given in Fig. 2. In the group of five agents the states 'idle' are modelled by p_1, p_3, p_5, p_7, p_9 and the states 'working' are modelled by p_2, p_4, p_6, p_8, p_{10}. It is an analogy with the problem of 5 dining philosophers [7]. The robots are situated in a circle. Between two robots a needful identical device is situated. All of the agents need for their work two such devices. However, the number of these devices is also only 5. Each agent has own device but the second device it have to obtain from the left or right neighbour. It means that the agents have to share the devices. Formally, the availabilities of the devices are expressed by means of the PN places p_{11}, p_{12}, p_{13}, p_{14}, p_{15} - see Fig. 3, apart from their relations with the robots. Any robot can take devices only from its neighbours. The problem how to synthesized the cooperation of agents and devices can be solved by the supervisor synthesis [2,3,4,5,6] based on DES control theory results [8,9,10]. The incidence matrices and initial states of the PN models of the robots A_i, $i = 1$, ..., 5 are

$$\mathbf{F}_i = \begin{pmatrix} 1&0\\0&1 \end{pmatrix};\ \mathbf{G}_i^T = \begin{pmatrix} 0&1\\1&0 \end{pmatrix};\ \mathbf{B}_i = \begin{pmatrix} -1&1\\1&-1 \end{pmatrix};\ {}^i\mathbf{x}_0 = (1,0)^T;\ \ i = 1, ..., 5$$

The parameters of the PN model of the group of autonomous agents can be expressed as follows

$$\mathbf{F} = \text{blockdiag}(\mathbf{F}_i)_{i=1,5};\ \mathbf{G} = \text{blockdiag}(\mathbf{G}_i)_{i=1,5}$$

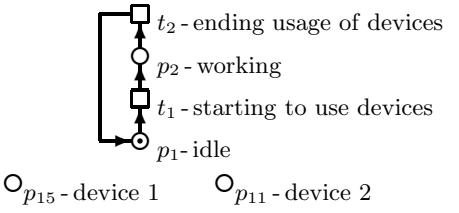


Fig. 2. The PN-based model of the single robot activities

$$\mathbf{x}_0 = ({}^1\mathbf{x}_0^T, {}^2\mathbf{x}_0^T, {}^3\mathbf{x}_0^T, {}^4\mathbf{x}_0^T, {}^5\mathbf{x}_0^T)^T$$

The conditions imposed on the autonomous agents are

$$\begin{aligned} \sigma_{p_2} + \sigma_{p_4} &\le 1 \\ \sigma_{p_4} + \sigma_{p_6} &\le 1 \\ \sigma_{p_6} + \sigma_{p_8} &\le 1 \\ \sigma_{p_8} + \sigma_{p_{10}} &\le 1 \\ \sigma_{p_{10}} + \sigma_{p_2} &\le 1 \end{aligned}$$

Verbally it means that two adjacent robots (neighbours) must not work simultaneously. These conditions yield [2,3,4,5,6] the matrix $\mathbf{L}$ and the vector $\mathbf{b}$ as follows

$$\mathbf{L} = \begin{pmatrix} 0\,1\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix};\ \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix};\ \mathbf{B}_s = -\mathbf{L}.\mathbf{B};\ {}^s\mathbf{x}_0 = \mathbf{b} - \mathbf{L}.\mathbf{x}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{B}_s = \begin{pmatrix} -1\,1 & -1\,1 & 0\,0 & 0\,0 & 0\,0 \\ 0\,0 & -1\,1 & -1\,1 & 0\,0 & 0\,0 \\ 0\,0 & 0\,0 & -1\,1 & -1\,1 & 0\,0 \\ 0\,0 & 0\,0 & 0\,0 & -1\,1 & -1\,1 \\ -1\,1 & 0\,0 & 0\,0 & 0\,0 & -1\,1 \end{pmatrix}$$

$$\mathbf{F}_s = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,1\,0 \\ 1\,0\,0\,0\,0\,0\,0\,0\,1\,0 \end{pmatrix};\ \mathbf{G}_s^T = \begin{pmatrix} 0\,1\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix}$$

The structural matrices $\mathbf{F}_s$, $\mathbf{G}_s$ of the supervisor give us the structural interconnections between the robots and the devices. Using the supervisor synthesis the problem was easily resolved. The PN-based model of the solution - the cooperating agents - is given in Fig. 3.

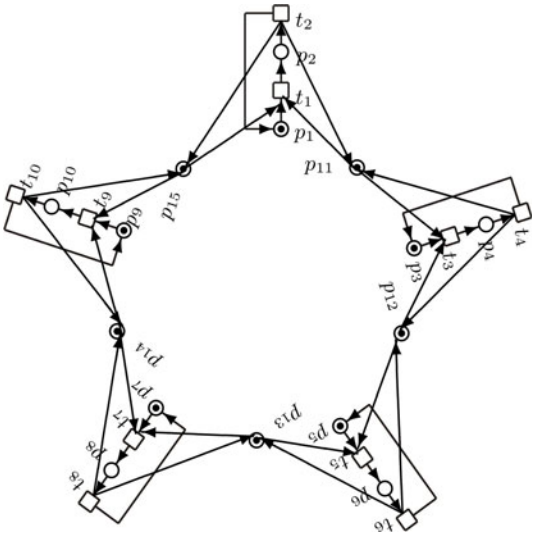


Fig. 3. The PN-based model of the five robots cooperating by means of the supervisor

3.3 Communication of Agent Models through the PN Transitions and Places

Finally, let us analyze the third (combined) form of the agent communication on the concrete example.

Example 3. Consider Example 1 introduced in the part 3.1. The cooperation through the transitions were illustrated there. To illustrate the combined cooperation (i.e. through both the transitions and the places) let us request from the structure presented in Example 1 satisfying the additional condition that only two of the three agents A_1, A_2, A_3 are allowed to want the communication simultaneously. This yields the additional condition

$$\sigma_{p3} + \sigma_{p6} + \sigma_{p9} \leq 2 \tag{4}$$

$$\mathbf{L} = \begin{pmatrix} 0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0 \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} 2 \end{pmatrix}$$

$$\mathbf{B}_s = -\mathbf{L}.\mathbf{B}; \; {}^s\mathbf{x}_0 = \mathbf{b} - \mathbf{L}.\mathbf{x}_0$$

$$\mathbf{B}_s = \begin{pmatrix} 0\,0\,-1\,1\,0\,0\,-1\,1\,0\,0\,-1\,1\,1\,1\,0\,2\,1\,1\,0\,2\,1\,1\,0\,2 \end{pmatrix}$$

$$\mathbf{F}_s = \begin{pmatrix} 0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{pmatrix}$$

$$\mathbf{G}_s^T = \begin{pmatrix} 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,2\,1\,1\,0\,2\,1\,1\,0\,2 \end{pmatrix}$$

For $\mathbf{x}_0 = (0\,0\,1\,0\,0\,1\,0\,1\,0\,0\,1\,0\,1\,0\,1)^T$ we have ${}^s\mathbf{x}_0 = 0$. The supervised system is displayed in Fig. 4. The system has 35 feasible states. Its reachability graph having 35 nodes expresses the mutual interconnections among the feasible states.

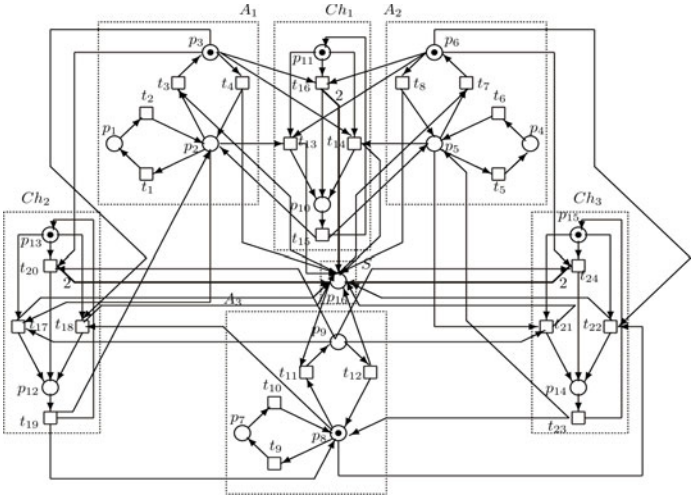


Fig. 4. The communication of three agents A_1, A_2, A_3 supervised by the supervisor S

The nodes of the graph are the vectors being the columns of the matrix

$$\mathbf{X}_{reach} = \begin{pmatrix}
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&1&1&1&0&0&1&1\\
0&1&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&1&1&0&0\\
1&0&1&1&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&1&0&0&1&1&1&1\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&0&0&1&1&0&0&0&0\\
1&1&0&1&0&1&0&1&0&1&1&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&1&0&1&0&1&0&1\\
1&1&1&0&1&0&0&1&1&0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&1&0&1&0&1&1\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&0&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&0&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&1&1&0&2&1&1&1&2&1&0&2&1&1&0&2&2&1&2&1&0&2&2&2&1&1&0&2&2&2&1&2&1&2&1&1
\end{pmatrix}$$

4 Conclusion

Three kind of agents communication were defined and presented. They yield the room for the agent cooperation and negotiation. The particular kinds of the communication - the communication through the PN transitions, through the PN places and combined cooperation through transitions and places - were illustrated by simple examples to point out their applicability. The individual autonomous agents were modelled by means of P/T PN as well as their communication. The instances of the applicability indicate soundness of the presented

approach. The dynamics of the cooperating agents were tested by means of PN-based methods - namely by finding the space of reachable states.

Acknowledgement. The work was partially supported by the Slovak Grant Agency for Science VEGA under grant # 2/0075/09 and by the Slovak Grant Agency APVV under grant APVV-0261-10. The author thanks the agencies for the support.

References

1. Čapkovič, F.: Modelling, Analysing and Control of Interactions among Agents in MAS. Computing and Informatics 26(5), 507–541 (2007)
2. Čapkovič, F.: Automatic Control Synthesis for Agents and their Cooperation in MAS. Computing and Informatics 29(7), 1045–1071 (2010)
3. Čapkovič, F.: Cooperation of Agents in Manufacturing Systems. In: Jedrzejowicz, P., Nguyen, N.T., Howlett, R.J., Lakhmi, C.J. (eds.) KES-AMSTA 2010. LNCS, vol. 6070, pp. 193–202. Springer, Heidelberg (2010)
4. Čapkovič, F.: DES Control Synthesis and Cooperation of Agents. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 596–607. Springer, Heidelberg (2009)
5. Čapkovič, F.: A Modular System Approach to DES Synthesis and Control. In: Proceedings of the 2009 IEEE Conference on Emerging Technologies & Factory Automation - ETFA 2009. CD ROM, 8 pages. IEEE Press, Piscataway (2009)
6. Čapkovič, F., Jotsov, V.: A System Approach to Agent Negotiation and Learning. In: Sgurev, V., Hadjiski, M., Kacprzyk, J. (eds.) Intelligent Systems: From Theory to Practice. Series: Studies in Computational Intelligence, vol. 299, pp. 133–160. Springer, Heidelberg (2010)
7. Dijkstra, E.W.: Hierarchical Ordering of Sequential Processes. Acta Informatica 1(2), 115–138 (1971)
8. Iordache, M.V., Antsaklis, P.J.: Supervision Based on Place Invariants: A Survey. Discrete Event Dynamic Systems 16(4), 451–492 (2006)
9. Iordache, M.V., Antsaklis, P.J.: Supervisory Control of Concurrent Systems: A Petri Net Structural Approach. Birkhauser, Boston (2006)
10. Iordache, M.V.: Methods for the supervisory control of concurrent systems based on Petri nets abstraction. Ph.D. dissertation, University of Notre Dame, USA (2003)
11. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77(4), 541–580 (1989)
12. Saint-Voirin, D., Lang, C., Guyennet, H., Zerhouni, N.: Scoop Methodology: Modeling, Simulation and Analysis for Cooperative Systems. Journal of Software 2(4), 32–42 (2007)
13. Saint-Voirin, D., Lang, C., Zerhouni, N.: Distributed Cooperation Modeling for Maintenance Using Petri Nets and Multi-Agents Systems. In: International Symposium on Computational Intelligence in Robotics and Automation - CIRA 2003, pp. 366–371. IEEE Computer Society Press, Piscataway (2003)

Modelling Relationship between Antecedent and Consequent in Modal Conditional Statements

Grzegorz Skorupa and Radosław Katarzyniak

Wroclaw University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{grzegorz.skorupa,radoslaw.katarzyniak}@pwr.wroc.pl

Abstract. We consider a cognitive agent that is able to utter natural language conditional statements with modal operators of possibility and belief. Our long term aim is to simulate humans' ability to choose a statement within an agent. We are developing formal criteria on when a conditional statement can be said based on agent's knowledge. In previous works we defined a formal model for choosing a proper conditional statement. Here we focus more on a relation between antecedent and consequent. The relation between two phrases that is required to utter a conditional statement. We generalise our previous model to make it more flexible according to requirements and constraints imposed on an agent.

Keywords: conditional statement, antecedent and consequent relation, natural language processing, modal operator, cognitive agent.

1 Introduction

A communication between a computer and a human suffers from lack of appropriate methods for understanding and properly choosing natural language statements. So far studies concentrated on logical properties of a statement, its truth evaluation and formal proof systems. Unfortunately statement being true is not equivalent to statement being suitable to situation. Let us assume some logical formula is true. Let us take a natural language statement equivalent of this formula. It doesn't have to be rational to use this natural language statement. Grice noticed this distinction. He proposed a formal term: *implicature* to define what is meant by a statement but not directly given by it. Ajdukiewicz [1] claims that a true conditional statement is not necessarily a well used statement. He concludes that proper usage of a statement relies on *language common*.

Our aim is to formally constraint usage of some of the most common natural language statements. We search for common sense criteria for deciding whether a statement can be properly used basing on agent's knowledge. We design a formal mathematical model and definitions for implementing these criteria within a cognitive agent [8].

Currently we focus on conditionals with modal operators of possibility and belief. We analyse their meaning and common usage patterns. Much of the work

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 120–129, 2011.

on simple statements, conjunctions and alternatives [9] and on conditionals in particular [12,13,14] has already been done.

Within this work we generalise a model developed in [13]. We formally model a relation between antecedent and consequent. The relation that allows humans to join two phrases within a conditional statement. Connection suggesting, that the antecedent has impact on the consequent.

Within section 2 we define types of considered conditional statements and how do we understand their proper usage. Section 3 describes how do we understand a relation between antecedent and consequent. Within section 4 we describe common sense constraints, that we later use to define a formal model for catching conditional relation (section 5). In section 6 we redefine formal criteria given in [13] using a definition of conditional relation.

2 Considered Statements and Their Usage

A conditional is a statement of the form: "If ψ, then ϕ". Where ψ is called antecedent and ϕ is consequent. Conditional statements can be classified according to many criteria [11]. The most common classification is into indicative and subjunctive conditionals. Indicative conditionals refer to situations that have a realistic chance of happening. Subjunctive conditionals describe purely hypothetical situations that can't happen or are very improbable to happen. For example [4] "If it rains the match will be cancelled" is a indicative conditional, while "If it were to rain, the match would be cancelled" is a subjunctive conditional. Within this work we focus only on indicative conditionals.

We consider modal conditionals what is understood here as a conditional statement where consequent can contain a modal operator of possibility or belief. The formal language has already been introduced in [13]. In table 1 we recall some of the most important definitions crucial in understanding rest of this work.

Table 1. Language semantics from [12]

Formula	Meaning
$q(o)$	o is q. (Object o has property q.)
$\neg q(o)$	o is not q. (Object o does not have property q.)
$\phi \rightarrow \psi$	If ϕ, then ψ.
$\phi \rightarrow Bel(\psi)$	If ϕ, then I believe, that ψ.
$\phi \rightarrow Pos(\psi)$	If ϕ, then it is possible, that ψ.

Variables ϕ and ψ are assumed to be simple sentences of the form $q(o)$ or $\neg q(o)$. The language is not extensible. We do not consider composite conditionals.

Interpretation of the language considers common understanding of statements and is not a formal interpretation of classical or modal logic. Conditional statements should be understood literally as conditional statements, not any kind of

formal implication. Modal operators *Pos* and *Bel* have intuitive meaning consistent with [7] and should not be misunderstood as a formal modal operators with possible worlds semantics from [10].

Our aim is to formally define criteria of properly using a modal conditional statement. Although the understanding of properly using a conditional statement and its logical truth are closely related they are not the same notions. Not every true statement can be properly used in any situation. This distinction has already been recognised and analysed [1,6].

Proper usage of a statement and how it is understood has been already discussed in [12,13]. We assume a statement is properly used when it fits speaker's knowledge well. When a listener is not mislead by a speaker and finally when the listener reads the meaning of a statement in a way that is meant by the speaker. Statements proper usage can be checked against various criteria: conversational context, aims of the listener and the speaker, cultural context and also speakers and listeners knowledge. Within our work we focus only on the speakers knowledge as we believe it is the most basic and first checked criterion. We wish to answer the question: When a speaker is allowed to utter a modal conditional statement considering his knowledge.

3 The 'Magic' Connection

The knowledge that antecedent holds (or doesn't hold) must have impact on the subjective chance of consequent holding. Strawson [15] claims that natural language conditional implies some kind of connection between antecedent and consequent. Ajdukiewicz [1] shifts the problem to language common. Grice calls it *conversational implicature*. It is difficult to deny that there must be some kind of relation that joins antecedent and consequent. Otherwise there is no point in stating an indicative conditional.

The major drawback of existing formal theories on implications and conditionals (see [2] for a broad review), is that they do not filter out statements referring to unrelated items, events, properties etc.. Existing theories focus on logical properties required for reasoning and decision making, not the proper usage of natural language.

All statements below are true as material implications of Boolean Logic, epistemic conditionals [5]. Some of these statements are true as strict conditionals proposed by Lewis C.L. (described later in [3]):

- If the moon is a piece of cheese then, I can jump 100 meters.
- If the moon is a piece of cheese then, I am a human.
- If $2 + 2 = 5$, then I will die on a day with an even date.
- If Roosevelt was a president, then birds can fly.

On the other hand no one would use these statements in normal circumstances. There simply is no connection between expressions embodied within them.

Humans are able to decide when there is a relation between two phrases that enables them to utter a conditional statement. Further in this paper we shall call this relation a *conditional relation*. Our aim is to design a formal model of *conditional relation* which can be embedded into a cognitive agent.

4 Common Sense Constraints

The first conclusion is that neither antecedent, nor consequent can be true (or false) according to the speaker. This is consistent with [4,15]. If the speaker knows any of the truth values there is no point in stating a conditional. We simply don't use indicative conditionals when facing facts we already know. This assumption seems to fix the problem with indicative conditionals. All exemplary statements from section 3 fail the assumption. On the other hand, when truth values are unknown to the speaker, material implication is usually true when indicative conditional is intuitively a properly used statement. This conclusion already appeared in [12].

Problem becomes more complex when we want to model the connection between antecedent and consequent in modal conditionals. How do we decide if phrases in modal conditional are related or not? Suppose someone says: "If he is a smoker, then it is possible, that he has lung cancer". It is obvious to us that smoking and having cancer are related. On the other hand we wouldn't normally say: "If the ball is red, then it is possible that he is old". We clearly see antecedent and consequent have no meaningful relation.

4.1 Change of Chance Rule

In order to formally define *conditional relation* we suggest an intuitive rule that allows humans to find connection between antecedent and consequent. We easily decide when antecedent influences the consequent. Intuitively our unconscious thought process can be written as follows:

1. Assume that antecedent is true and evaluate the subjective chance of consequent being true.
2. Now assume that antecedent is not true and evaluate the same chance.
3. If both chances are significantly different there is a relation between antecedent and consequent.

This remark leads to formulating a *change of chance rule*:

The subjective chance of consequent being true must be significantly different in a hypothetical situation where antecedent is true compared to the situation where antecedent is not true.

This rule suggests that we must consider two chances from the speakers point of view. Can it be strengthened? The answer is: usually yes but not always.

4.2 Rise of Chance Rule

When searching for a rule stronger than *change of chance rule* one can consider if that chance shouldn't be always greater when antecedent holds.

Suppose someone says: "If he doesn't smoke, then it is possible, that he has a lung cancer". Let us assume the statement is used without any conversational context. We claim that this is not a proper usage of this statement. The listener is suggested that smokers have smaller chance of having lung cancer. Humans know that smoking may cause cancer and unconsciously fix the meaning of the statement to fit this well known fact. The situation becomes more clear when we replace meaningful statements with symbols. Suppose someone says "If P, then it is possible, that Q". You, as a listener, infer that Q is much less or even impossible without P. The statement clearly suggests that P increases the chance of Q. This leads to a stricter rule, the *rise of chance rule*:

The subjective chance of consequent being true must be significantly greater in a hypothetical situation where antecedent is true compared to the situation where antecedent is not true.

It is a very useful rule, because it drops many unintuitive conditionals. Unfortunately, when considering conversational context, the rise of chance rule does not have to hold. Let us consider a short dialog between A and B:

- A: "Is it possible that a non-smoker has a lung cancer?"
- B: "If he doesn't smoke, then it is possible, that he has a lung cancer."

The answer seems correct within the dialog. Speaker A forces a conditional statement on speaker B. We shall call such situations *forced conditional situations*. Speaker B only checks, if there is a relation between antecedent and consequent but does not consider whether this chance rises or drops. This does not mean that B doesn't consider the change of chance at all. Let us consider following dialog:

- A: "Is it possible, that if the ball is yellow, then the apple is ripe?"
- B: "If the ball is yellow, then it is possible, that the apple is ripe."

The answer seems weird. There clearly is no relation between ball and apple. Agent B should not respond as in given dialog. He should rather say: "It is possible that the apple is ripe regardless of the ball's colour". One can see that speaker B still checks the relation between antecedent and consequent.

The *rise of chance rule* should be used whenever possible. It seems to work well in all situations except *forced conditional situations*. This rule limits the amount of conditional statements agent can use. On the other hand it also makes her more exact in all 'normal' situations. When to use a *rise of chance rule* and when to limit agent to *change of chance rule* is left here as an open problem.

5 Modelling the Relation between Antecedent and Consequent

Our aim is to design a formal model of conditional relation meeting common sense constraints described in section 4. Let us consider a conditional statements of the forms: $\phi \rightarrow \psi$, $\phi \rightarrow Pos(\psi)$ and $\phi \rightarrow Bel(\psi)$. It is assumed phrases ϕ and ψ describe a feature of some chosen object. The object can have or lack the feature, there are no intermediate states. There are four hypothetical cases, a speaker has to consider, in order to make a decision about uttering a conditional statement:

$\neg\phi \wedge \neg\psi$ - ϕ and ψ do not hold
$\neg\phi \wedge \psi$ - ϕ does not hold, and ψ holds
$\phi \wedge \neg\psi$ - ϕ holds, and ψ does not hold
$\phi \wedge \psi$ - both ϕ and ψ hold

According to our assumptions the speaker must be unaware which of the cases is taking place. We assume the agent is able to evaluate how probable each case is. She knows respective probabilities: $P(\neg\phi \wedge \neg\psi), P(\neg\phi \wedge \psi), P(\phi \wedge \neg\psi), P(\phi \wedge \psi)$. Events are disjoint and their probabilities sum to 1.

Case $\phi \wedge \psi$ supports the conditional and case $\phi \wedge \neg\psi$ neglects it. Remaining cases, where ϕ does not hold, don't have impact on the truth of material implication. On the other hand we cannot deny that they may interfere with conditional statement's usage. Cases $\neg\phi \wedge \psi$ and $\neg\phi \wedge \neg\psi$ can give us information about chance of ψ holding when ϕ does not hold. This information is crucial in determining whether there is a conditional relation between ϕ and ψ. The relation that is required in every conditional statement. When ψ behaves the same way regardless of ϕ, there is no conditional relation. This implies that the speaker must consider all the cases.

We claim that knowing respective probabilities is enough to determine whether there exists a conditional relation between ϕ and ψ. The remaining question is how to construct conditional relation based on these probabilities. Let:

$$p_T = P(\psi|\phi) = \frac{P(\phi \wedge \psi)}{P(\phi \wedge \psi) + P(\phi \wedge \neg\psi)} \tag{1}$$

$$p_F = P(\psi|\neg\phi) = \frac{P(\neg\phi \wedge \psi)}{P(\neg\phi \wedge \psi) + P(\neg\phi \wedge \neg\psi)} \tag{2}$$

be conditional probabilities of antecedent holding in case consequent holds and does not hold respectively. Probability of each of hypothetical cases is included within equations above.

These two conditional probabilities tell a lot about the behaviour of ψ depending on ϕ. If p_T and p_F are equal, ψ doesn't depend on ϕ. On the contrary, if there is a large difference between the values p_T and p_F, then ψ and ϕ are highly dependent. The second situation is a good candidate to utter a conditional statement.

Let $\underline{f} : [0,1] \rightarrow [0,1]$ and $\overline{f} : [0,1] \rightarrow [0,1]$ be functions describing lower boundary and upper boundary respectively. It is assumed functions meet following criteria:

1. $\underline{f}(x) \leq x$ and $\overline{f}(x) \geq x$ for all x
2. $\underline{f}, \overline{f}$ are monotonically increasing (non-decreasing)
3. $\underline{f}, \overline{f}$ are not constant functions of 0 and 1 respectively
4. $\underline{f}, \overline{f}$ are continuous

These functions shall be used to determine whether there is a conditional relation between ψ and ϕ.

Definition 1. *ψ and ϕ are conditionally related iff $p_T > \overline{f}(p_F)$ or $p_T < \underline{f}(p_F)$*

Function $\underline{f}$ determines if p_T probability is significantly greater than p_F. Similarly $\overline{f}$ determines if p_T probability is significantly smaller than p_F. Suppose $p_T > \overline{f}(p_F)$ is true, it implies that ψ is more probable in case of ϕ, than in case of not ϕ. Definition is compatible with *change of chance rule* but not with *rise of chance rule*.

Figure 1a represents two arbitrary chosen boundary functions. Horizontal axis represents a conditional probability p_F. Vertical axis represents conditional probability p_T. Any point on line $p_T = p_F$ represents a situation when ϕ does not depend on ψ. The gray area above upper boundary and below lower boundary represents situations where the conditional relation is met.

Conditional relation can be constrained if one does not require handling forced conditional statements described in section 4.2.

Definition 2. *ϕ and ψ are strictly conditionally related iff $p_T \geq \overline{f}(p_F)$*

Strict conditional relation requires that the probability of consequent is greater in case of antecedent. This is compatible with *rise of chance rule*.

5.1 Some Properties of Conditional Relation

Theorem 1. *If $P(\psi) = 1$, then conditional relation is not met.*

$\underline{P}(\psi) = 1$ implies that $p_F = p_T = 1$ which in turn is neither greater than $\overline{f}(1) = 1$, nor smaller than $\underline{f}(1) \leq 1$. This property means that the speaker can't know consequent holds regardless of antecedent.

Theorem 2. *If $P(\psi) = 0$, then conditional relation is not met.*

$P(\psi) = 0$ implies that $p_F = p_T = 0$ which in turn is neither smaller than $\underline{f}(0) = 0$, nor greater than $\overline{f}(0) \geq 0$. This property means that the speaker can't know consequent does not hold regardless of antecedent.

Theorem 3. *High probability of the consequent ψ does not guarantee that conditional relation is met.*

Example: $P(\neg\phi \wedge \neg\psi) = 0.05, P(\neg\phi \wedge \psi) = 0.45, P(\phi \wedge \neg\psi) = 0.05, P(\phi \wedge \psi) = 0.45$ implies that $P(\psi) = 0.9$ and results in $p_T = p_F$. Conditional relation fails for any upper and lower boundary functions.

Theorem 4. *Low probability of the consequent ψ does not guarantee that conditional relation is not met.*

Example: $P(\neg\phi \wedge \neg\psi) = 0.9, P(\neg\phi \wedge \psi) = 0, P(\phi \wedge \neg\psi) = 0, P(\phi \wedge \psi) = 0.1$ implies that $P(\psi) = 0.1$ and results in $p_T = 1$ and $p_F = 0$. Conditional relation succeeds for any lower and upper boundary functions.

Theorem 5. *If $P(\psi|\phi) = 1$, $P(\psi) < 1$ and $\forall x \in [0, 1) : \overline{f}(x) < 1$ the conditional relation is met.*

In such case $p_T = 1 > f(p_F)$ and $p_F < 1$. This means that conditional relation holds in almost all cases where material implication holds. The only cases that meet material implication and not the conditional relation are when $P(\psi)$ or $P(\phi)$ is known to be 0 or 1. These are exactly the cases we wanted to rule out.

Theorem 6. *When ψ and ϕ are independent ($P(\phi \wedge \psi) = P(\phi)\,P(\psi)$) conditional relation is never met.*

Such case always results in $p_T = p_F$. This means that the speaker must notice a dependance between antecedent and consequent.

Theorem 7. *If strict conditional relation holds, then $p_T > p_F$*

Can be proven directly from the fact that $\overline{f}(x) \geq x$. Property 7 means that antecedent truth must increase chance for consequent truth when considering strict conditional relation.

6 Defining Epistemic Relation Using Conditional Relation

We define epistemic relation to formalise when a conditional statement is properly used. Each time relation holds a statement can be properly used. When relation does not hold the statement can't be used. This relation has already been defined for conditional statements in [13] but in a less general manner. Here we redefine it to use a *conditional relation*.

Definition 3. *Epistemic relation $\models^E \phi \rightarrow \psi$ holds iff all following conditions are met:*

a. $\underline{\alpha} < P(\phi) < \overline{\alpha}$

b. $P(\psi|\phi) = 1$

c. $P(\psi|\phi) \geq \overline{f}(P(\psi|\neg\phi))$ *or* $P(\psi|\phi) \leq \underline{f}(P(\psi|\neg\phi))$ *(conditional relation)*

where $0 < \underline{\alpha} < \overline{\alpha} < 1$ *are fixed parameters.*

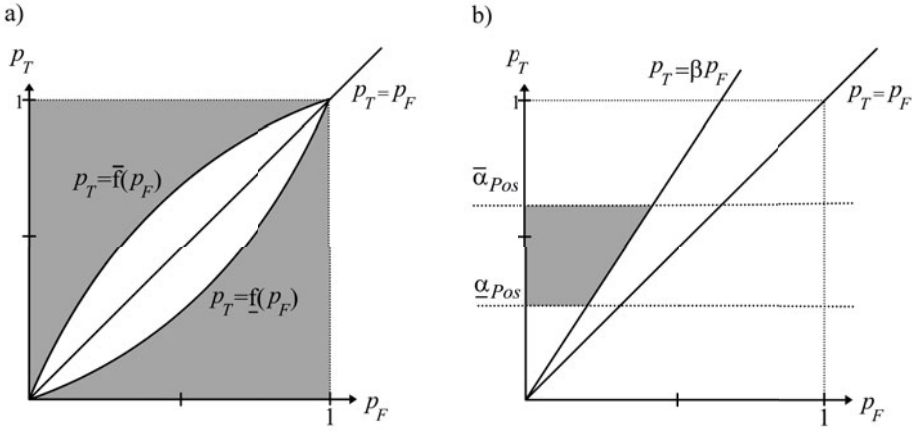


Fig. 1. a) Gray area represents situations where conditionals can be used according to conditional relation. b) Gray area represents situations where $\phi \rightarrow Pos(\psi)$ can be used according to epistemic relation definition for linear upper boundary function and strict conditional relation definition.

Definition 4. *Epistemic relation* $\models^E \phi \rightarrow Bel(\psi)$ *holds iff all following conditions are met:*

a. $\underline{\alpha} < P(\phi) < \overline{\alpha}$

b. $\underline{\alpha}_{Bel} \leq P(\psi|\phi) < \overline{\alpha}_{Bel}$

c. $P(\psi|\phi) \geq \overline{f}(P(\psi|\neg\phi))$ *or* $P(\psi|\phi) \leq \underline{f}(P(\psi|\neg\phi))$ *(conditional relation)*

where $0 < \underline{\alpha} < \overline{\alpha} < 1$, $0 < \underline{\alpha}_{Bel} < \overline{\alpha}_{Bel} < 1$ *are fixed parameters.*

Definition 5. *Epistemic relation* $\models^E \phi \rightarrow Pos(\psi)$ *holds iff all following conditions are met:*

a. $\underline{\alpha} < P(\phi) < \overline{\alpha}$

b. $\underline{\alpha}_{Pos} < P(\psi|\phi) < \overline{\alpha}_{Pos}$

c. $P(\psi|\phi) \geq \overline{f}(P(\psi|\neg\phi))$ *or* $P(\psi|\phi) \leq \underline{f}(P(\psi|\neg\phi))$ *(conditional relation)*

$0 < \underline{\alpha}_{Pos} < \overline{\alpha}_{Pos} < 1$ *are fixed parameters.*

Condition *a* in definitions 3-5 guarantees that the speaker does not know whether antecedent is true. There must be some realistic chance of it holding or not. Condition *b* defines how probable consequent is based on antecedent. Condition *c* is different than in [13]. It uses a conditional relation. If one wants to use *strict conditional relation* he can replace definition of *conditional relation* from condition *c* with definition of *strict conditional relation.*

7 Summary

We formally defined *conditional relation* and *strict conditional relation.* These relations can be used within a cognitive agent to filter out meaningless natural

language conditional statements. Defined relation can be implemented in an agent that uses probabilistic world model.

We generalised a model for *epistemic relation* presented in [13]. New model defines a relation between antecedent and consequent more precisely. On the other hand it requires the developer to choose appropriate lower and upper boundary functions.

Acknowledgements. This paper was partially supported by Grant no. N N519 407437 funded by Polish Ministry of Science and Higher Education (2009-2012).

References

1. Ajdukiewicz, K.: Conditional sentence and material implication. Studia Logica 4(1), 135–153 (1956)
2. Arlo-Costa, H.: The Logic of Conditionals. Entry for the Stanford Encyclopedia of Philosophy (2007)
3. Bennett, J.: A Philosophical Guide to Conditionals. Oxford University Press, Oxford (2003)
4. Clark, M.: Ifs and Hooks. Analysis 32(2), 33–39 (1971)
5. Gilles, A.S.: Epistemic Conditionals and Conditional Epistemics. Nous 38(4), 585–616 (2004)
6. Grice, H.P.: Meaning. Philosophical Review 66, 377–388 (1957)
7. Hintikka, J.: Knowledge and belief. An introduction to the logic of the two notions. Cornell University Press, Ithica (1962)
8. Huhns, N., Signh, M.: Cognitive Agents. IEEE Internet Computing 2(6), 87–89 (1998)
9. Katarzyniak, R.: Gruntowanie modalnego języka komunikacji w systemach agentowych. Exit, Warsaw (2007) (in Polish)
10. Kripke, S.: Semantical Analysis of Modal Logic. Zeitschrift für Mathematische Logic und Grundlagen der Mathematik 9, 67–96 (1963)
11. Pelc, J.: Jeżeli, to (If, then) Studia Semiotyczne, XIV-XV 271-286 (1986) (in Polish)
12. Skorupa, G., Katarzyniak, R.: Extending modal language of agents' communication with modal implications. Information Systems Architecture and Technology, 127–136 (2009)
13. Skorupa, G., Katarzyniak, R.: Applying Possibility and Belief Operators to Conditional Statements. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6276, pp. 271–280. Springer, Heidelberg (2010)
14. Skorupa, G., Katarzyniak, R.: Conditional Statements grounded in Past, Present and Future. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS (LNAI), vol. 6423, pp. 112–121. Springer, Heidelberg (2010)
15. Strawson, P.F.: Introduction to Logical Theory, pp. 82–90. Methuen & Co. Ltd., New York (1952)

Semantic Simulation Engine for Supervision of Mobile Robotic System

Janusz Bedkowski[1,2] and Andrzej Masłowski[1,2]

[1]Institute of Automation and Robotics, Warsaw, Poland
[2]Institute of Mathematical Machines, Warsaw, Poland

Abstract. In the paper semantic simulation engine for supervision of mobile robotic system is described. Semantic simulation engine provides tools to implement mobile robot simulation based on real data delivered by robot observations in INDOOR environment. The supervision of real objects such as robots is performed by association with its virtual representation in the simulation, therefore events such as object intersection, robot pitch roll are defined. Semantic simulation engine is composed of data registration modules, semantic entities identification modules and semantic simulation module. The data registration modules delivers 3D point clouds aligned with ICP (Iterative Closest Point) algorithm. Semantic entities identification modules provide implementation of methods for obtaining semantic entities from robot observations. Semantic simulation module executes rigid body simulation with predefined simulation events. The simulation can be integrated with real part of the system with an assumption of robust localization of real entities.

Keywords: semantic mapping, robotic system, supervision.

1 Introduction

In this paper new idea of semantic simulation engine is proposed. Semantic simulation engine combines semantic map with rigid body simulation to perform supervision of its entities such as robots moving in INDOOR environment composed by floor, ceiling, walls, door etc. Cognitive aspect is related to automatic detection of these entities based on robot's observations, the relations between them are defined by semantic net. The paper is organized as follows: section *Problem formulation* describes the problem taken in this paper, section *Multi robot system structure* describes the structure of robotic system, section *Data registration* is showing the new approach of parallel computing applied for data registration, Section *Semantic entities identification* shows the implementation of 3D data segmentation techniques to obtain semantic entities, section *Semantic simulation* shows the main concept of rigid body simulation. Section *Conclusion and future work* finalizes the paper.

2 Related Work

Proposed supervision of mobile robotic system is related to semantic mapping. Semantic information [1] extracted from 3D laser data [2] is recent research

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 130–139, 2011.

topic of modern mobile robotics. In [3] a semantic map for a mobile robot was described as a map that contains, in addition to spatial information about the environment, assignments of mapped features to entities of known classes. In [4] a model of an indoor scene is implemented as a semantic net. This approach is used in [5] where robot extracts semantic information from 3D models built from a laser scanner. In [6] the location of features is extracted by using a probabilistic technique (RANSAC) [7]. Also the region growing approach [8] extended from [9] by efficiently integrating k-nearest neighbor (KNN) search is able to process unorganized point clouds. The improvement of plane extraction from 3D Data by fusing laser data and vision is shown in [10]. The automatic model refinement of 3D scene is introduced in [11] where the idea of feature extraction (planes) is done also with RANSAC. The semantic map building is related to SLAM problem [12]. Most of recent SLAM techniques use camera [13], laser measurement system [14] or even registered 3D laser data [15]. Concerning the registration of 3D scans described in [16] [17] we can find several techniques solving this important issue. The authors of [18] briefly describe ICP algorithm. In [19] the comparison of ICP and NDT algorithm is shown. In [20] the mapping system that acquires 3D object models of man-made indoor environments such as kitchens is shown. The system segments and geometrically reconstructs cabinets with doors, tables, drawers, and shelves, objects that are important for robots retrieving and manipulating objects in these environments.

A detailed description of computer based simulators for unmanned vehicles is shown in [21]. Also in [22] the comparison of real-time physics simulation systems is given, where a qualitative evaluation of a number of free publicly available physics engines for simulation systems and game development is presented. Several frameworks are mentioned such as USARSim which is very popular in research society [23] [24], Stage, Gazebo [25], Webots [26], Matlab [27] and MRDS [28]. Some researchers found that there are many available simulators that offer attractive functionality, therefore they proposed a new simulator classification system specific to mobile robots and autonomous vehicles [29]. A classification system for robot simulators will allow researchers to identify existing simulators which may be useful in conducting a wide variety of robotics research from testing low level or autonomous control to human robot interaction. Another simulation engine - the Search and Rescue Game Environment (SARGE), which is a distributed multi-player robot operator training game, is described in [30]. To conclude this section it can be stated that there are many available hardware and software techniques to implement sophisticated mobile robotic system, but in the same time the ergonomic solutions for HMI (Human Robot Interface) are needed to perform complex tasks. Based on this observation cognitive supervision and control can improve operator skills by replacing them in critical situations to avoid system failure.

3 Problem Formulation

Supervision of mobile robotics system is needed to prevent crisis situation defined as event when hardware damage occur. Proposed semantic simulation engine is

composed of several modules for which separate problems can be formulated. The goal for data registration module is to provide methods that can be executed on-line. The goal for semantic entities identification is to provide robust on-line methods that deliver accurate 3D models based on robot observations. Semantic simulation has to perform realistic entities behavior.

4 Multi Robot System

The main object of research in this work is a web connected mobile robot system, for example, the inspection – intervention system consisting of a mobile base station, a group of autonomous robots and remote-controlled robot, equipped with a manipulator with n degrees of freedom. The main tasks of the system is an inspection and intervention of hazardous environment for human activities. Mobile robots are represented by virtual entities in semantic simulation, robust localization and tracking is needed to obtain accurate position of these entities (tracking and localization problem is not discussed in this paper).

5 Data Registration

Alignment and merging of two 3D scans, which are obtained from different sensor coordinates, with respect to a reference coordinate system is called 3D registration [31] [32] [33]. Range images are defined as a model set M and data set D, where N_m and N_d denotes the number of the elements in the respective set. The alignment of these two data sets is solved by minimization of the following cost function:

$$E = (\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{ij} \left\| \mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t}) \right\|^2 \tag{1}$$

w_{ij} is assigned 1 if the i-th point of M correspond to the j-th point in D as in the same bucket (or neighbor bucket). Otherwise w_{ij}=0. $\mathbf{R}$ is a rotation matrix, $\mathbf{t}$ is a translation matrix. $\mathbf{m}_i$ and $\mathbf{d}_i$ corresponds to the i-th point from model set M and D respectively.

Solving equation 1 is related to Nearest Neighborhood search. The distance between two points in Euclidean distance metric for point $p_1 = \{x_1, y_1, z_1\}$ and $p_2 = \{x_2, y_2, z_2\}$ is defined as:

$$distance(p_1, p_2) = \left[(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \right]^{\frac{1}{2}} \tag{2}$$

To find pairs of closest points between model set M and data set D the decomposition of XYZ space, where x,y,z $\in < -1, 1 >$, into 2^8x2^8x2^8 buckets is proposed. It should be noticed that in case of 2^8x2^8x2^8 buckets cubic subspace length,width,height = $2/2^8$,$2/2^8$,$2/2^8$. Each bucket that does not belong to border has 26 neighbors. The 27 neighboring cubic subspaces are shown on figure 1

where also the way of indexing in CUDA GPU is given. The approach is new idea that differs from [34] [35] (improved ICP procedure fast searching algorithms such as the k-d tree) by no need of building complex data structure, threrefore computation time is decreased.

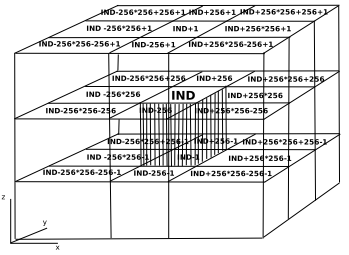


Fig. 1. Cubic subspaces - neighboring buckets, the way of indexing

5.1 Parallel Point to Point Registration

The ICP point to point algorithm using CUDA parallel programming is shown in algorithm 1. It should be noted that all computation is implemented in CUDA architecture, therefore the time needed to transfer data between host and device is decreased.

Algorithm 1. ICP - point to point parallel computing approach

allocate the memory
copy data from the host(M_{host}, D_{host}) to the device(M_{device}, D_{device})
for $iter = 0$ to $max\_iterations$ **do**
  select closest points between M_{device} and D_{device}
  calculate (R, t) that minimizes equation 1
  transform D_{device} by (R, t) and put the results into $D_{deviceRt}$
  copy $D_{deviceRt}$ to D_{device}
  if D_{device} is aligned to M_{device} **then**
    break
  end if
end for
copy D_{device} to D_{host}
free memory

5.2 Parallel Point to Plane Registration

Algorithm 2 - point to plane registration differs from algorithm 1 - point to point registration by nearest neighbor search. In this procedure nearest neighbor is computed as projected point onto plane. For each bucket plane is computed in initial step of the algorithm. It should be emphasized that the best plane is chosen that satisfies minimum of sum distances between points and chosen

plane in the range of 27 neighboring buckets, therefore local approximation is performed. It is very important to realize that planes approximating cloud of points from neighborhood improve noisy data matching, what is significant from real applications point of view. The weak point of proposed point to plane algo-

Algorithm 2. ICP - point to plane parallel computing approach

```
allocate the memory
copy data from the host(M_host, D_host) to the device(M_device, D_device)
in parallel compute M_planes for M_device points
choose best M_planeofbucket for each bucket
for iter = 0 to max_iterations do
   in parallel find M_planeofbucket for each data D_device
   in parallel compute distances between data D_device and its projection onto
   M_planeofbucket
   calculate (R, t) that minimizes equation 1
   transform D_device by (R, t) and put the results into D_deviceRt
   copy D_deviceRt to D_device
   if D_device is aligned to M_device then
      break
   end if
end for
copy D_device to D_host
free memory
```

rithm is the procedure of choose best $M_{planeofbucket}$ for each bucket that is not parallelized. At this stage of proposed implementation the computation is done in CPU, therefore the time computation is longer because of memory copying between host and device.

6 Semantic Entities Identification

For semantic entities identification semantic net shown on figure 2 is proposed. The proposed approach is dedicated to structured INDOOR environment where floor is flat and walls are straight and orthogonal to floor and ceiling, stairs are parallel and ergonomic. It is obvious that if we try to use this approach in unstructured environment, algorithm will generate numerous not-labeled objects. To find semantic objects such as Wall, Ceiling, Floor, Doors (with joint), Stairs we are using a new idea that is based on prerequisites generation from projected single 3D scan (onto OXY plane for Wall, Door, Stairs and onto OXZ for Ceiling and Floor). Prerequisites are generated using image processing techniques such as Hough transform for line extraction. The prerequisites are checked in next step, if the constraint is satisfied, the semantic object is assigned. For instance if we assume 3D scan projected onto OXY plane, a single line is related to a wall prerequisite, 3 parallel lines are prerequisite of stairs, 2 connected lines are prerequisite of opened doors and the connection can be a joint. Single long line in 3D scan projected onto OXZ plane is a prerequisite of Ceiling and Floor.

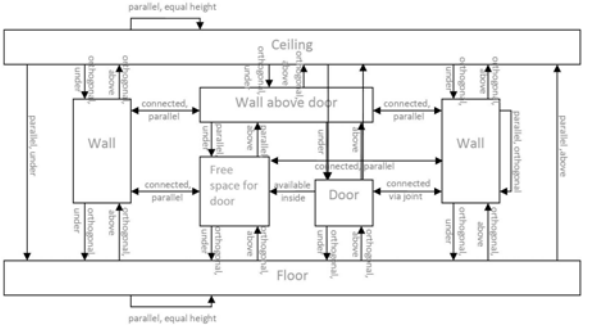


Fig. 2. Semantic net defined for semantic entities identification

The interpretation of the scene comprises generic architectural knowledge like in [4], [11] and [6]. Nodes of a semantic net represent entities of the world, the relationships between them are defined. Possible labels of the nodes are L = {Wall, Wall above door, Floor, Ceiling, Door, Free space for door}. The relationships between the entities are R = {parallel, orthogonal, above, under, equal height, available inside, connected via joint}. The semantic net can easily be extended to more entities and relationships which determine a more sophisticated feature detection algorithms.

6.1 Walls and Doors Detection

The procedure of prerequisites generation using image processing methods is used. The set of lines is used to obtain segmentation of $3D$ cloud of points, where different walls will have different labels. For each line segment the orthogonal $plane_{orth}$ to $plane_{OXY}$ is computed. It should be noted that the intersection between this two planes is the same line segment. All $3D$ points which satisfy the condition of distance to $plane_{orth}$ have the same label. In the first step all prerequisites of walls were checked separately - it is data segmentation. To perform the scene interpretation semantic net is used (figure 2). The feature detection algorithm is composed by the method of cubes generation (see figure 3 right), where each cube should contain measured 3D point after segmentation (see figure 3). In the second step of the algorithm wall candidates are chosen. From this set of candidates, based on relationships between them, proper labels are assigned and output model is generated (see figure 4 left).

6.2 Stairs Detection

The image processing methods are used for stairs prerequisites generation. It is important to emphasize that the set of parallel lines (obtained by projected single 3D scan onto OXY plane) in the same short distance between each other is prerequisite of stairs. Possible labels of the nodes are L = {stair}. The relationships between the entities are R = {parallel, above, under}. Figure 4 right shows

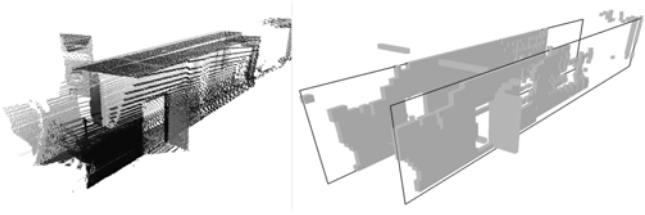


Fig. 3. Left - segmentation of 3D cloud of points, right - boxes that contain measured points

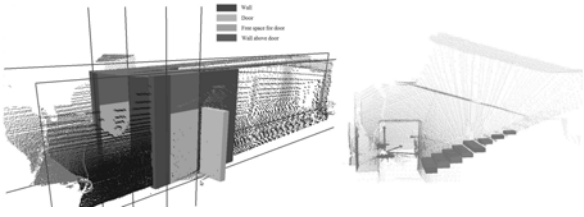


Fig. 4. Scene interpretation left - door, walls, right - stairs

resulting model of stairs generated from 3D cloud of points. In this spatial model each stair (except first and last one obviously) is in relation r=above&parallel with the previous one and in relation r=under&parallel with next one.

7 Semantic Simulation

The concept of semantic simulation is a new idea, and its strength lies on the semantic map integration with mobile robot simulator. The engine basic elements for INDOOR environment are: semantic map nodes(entities) L_{sm}={Wall, Wall above door, Floor, Ceiling, Door, Free space for door, Stairs...}, it is important to noticed that the L_{sm} set can be extended by another objects, what is dependent on robust and accurate 3D scene analysis, robot simulator nodes(entities) L_{rs}={robot, rigid body object, soft body object...}, semantic map relationships between the entities R_{sm} = {parallel, orthogonal, above, under, equal height, available inside, connected via joint...}, robot simulator relationships between the entities R_{rs} = {connected via joint, position...}, semantic map events E_{sm} = robot simulator events E_{rs} = {movement, collision between two entities started, collision between two entities stopped, collision between two entities continued, broken joint...}. Our robot simulator is implemented in NVIDIA PhysX. The entities from semantic map correspond to actors in PhysX. L_{sm} is transformed into L_{rs} based on spatial model generated based on registered 3D scans i.e. walls, doors and stairs correspond to actors with BOX shapes. R_{sm} are transformed into R_{rs} with remark that doors are connected to walls via revolute joints. All entities/relations R_{sm} has the same initial location in R_{rs}, obviously the location of each actor/entity may change during simulation. The transformation

from E_{sm} to E_{rs} effects that events related to entities from semantic map correspond to the events related to actors representing proper entities. It is important to emphasize that following events can be noticed during simulation: robot can touch each entity, open/close the door, climb the stairs, enter empty space of the door, damage itself (broken joint between actors in robot arm), brake joint that connects door to the wall. It is noteworthy to mention that all robot simulator semantic events are useful for operator supervision, where computer has to monitor simulation events judge them and report for the control purpose.

8 Conclusion and Future Work

In the paper new concept of semantic simulation engine composed of data registration modules, semantic entities identification modules and semantic simulation module is proposed. Compared to the State of The Art new approaches of parallel computing applied for data registration is proposed. Achieved performance has never been published before. Semantic simulation engine provides tools to implement mobile robot simulation based on real data delivered by robot and processed on-line using parallel computation. On-line means that computation is performed during robot motion (for example data registration of 262144 points with ICP point to point algorithm takes 300ms with GF 470 GTX). Two approaches of parallel ICP algorithm is proposed, point to point and point to plane. In comparison point to plane algorithm is more accurate but in the same time slower (1000ms for 262144 points) than point to point algorithm. Semantic entities identification modules can classify door, walls, floor, ceiling, stairs in indoor environment. Data can be delivered by robot observation based on modern sensors such as laser measurement system 3D and RGB-D cameras. Semantic simulation uses NVIDIA PhysX for rigid body simulation. By the association between real objects and simulation entities it is possible to supervise them by prediction of possible collisions and dangerous motion (pitch, roll). Future work will be related to AI techniques applied for semantic entities identification (furnitures, victims, cars, etc...), localization and tracking methods.

References

1. Asada, M., Shirai, Y.: Building a world model for a mobile robot using dynamic semantic constraints. In: Proc. 11 th International Joint Conference on Artificial Intelligence, pp. 1629–1634 (1989)
2. Nüchter, A., Wulf, O., Lingemann, K., Hertzberg, J., Wagner, B., Surmann, H.: 3d mapping with semantic knowledge. In: Robocup International Symposium, pp. 335–346 (2005)
3. Nüchter, A., Hertzberg, J.: Towards semantic maps for mobile robots. Robot. Auton. Syst. 56(11), 915–926 (2008)
4. Grau, O.: A scene analysis system for the generation of 3-d models. In: NRC 1997: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, p. 221 (1997)

5. Nüchter, A., Surmann, H., Lingemann, K., Hertzberg, J.: Semantic scene analysis of scanned 3d indoor environments. In: Proceedings of the Eighth International Fall Workshop on Vision, Modeling, and Visualization, VMV 2003 (2003)
6. Cantzler, H., Fisher, R.B., Devy, M.: Quality enhancement of reconstructed 3d models using coplanarity and constraints. In: Proceedings of the 24th DAGM Symposium on Pattern Recognition, pp. 34–41 (2002)
7. Fischler, M.A., Bolles, R.: Random sample consensus. a paradigm for model fitting with apphcahons to image analysm and automated cartography. In: Baurnann, L.S. (ed.) Proc. 1980 Image Understandtng Workshop (College Park, Md., Apr i980), Scmnce Apphcatlons, McLean, Va, pp. 71–88 (1980)
8. Eich, M., Dabrowska, M., Kirchner, F.: Semantic labeling: Classification of 3d entities based on spatial feature descriptors. In: IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska (May 3, 2010)
9. Vaskevicius, N., Birk, A., Pathak, K., Poppinga, J.: Fast detection of polygons in 3d point clouds from noise-prone range sensors. In: IEEE International Workshop on Safety, Security and Rescue Robotics, SSRR, pp. 1–6 (2007)
10. Andreasson, H., Triebel, R., Burgard, W.: Improving plane extraction from 3d data by fusing laser data and vision. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2656–2661 (2005)
11. Nüchter, A., Surmann, H., Hertzberg, J.: Automatic model refinement for 3D reconstruction with mobile robots, In: Fourth International Conference on 3-D Digital Imaging and Modeling 3DIM 2003, p. 394 (2003)
12. Davison, A., Cid, Y.G., Kita, N.: Real-time 3D SLAM with wide-angle vision. In: Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon (2004)
13. Castle, R.O., Klein, G., Murray, D.W.: Combining monoslam with object recognition for scene augmentation using a wearable camera, vol. 28(11), pp. 1548–1556 (2010)
14. Thrun, S., Burgard, W., Fo, D.: A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In: ICRA, pp. 321–328 (2000)
15. Magnusson, M., Andreasson, H., Nüchter, A., Lilienthal, A.J.: Automatic appearance-based loop detection from 3D laser data using the normal distributions transform. Journal of Field Robotics 26(11-12), 892–914 (2009)
16. Magnusson, M., Duckett, T., Lilienthal, A.J.: 3d scan registration for autonomous mining vehicles. Journal of Field Robotics 24(10), 803–827 (2007)
17. Andreasson, H., Lilienthal, A.J.: Vision aided 3d laser based registration. In: Proceedings of the European Conference on Mobile Robots (ECMR), pp. 192–197 (2007)
18. Besl, P.J., Mckay, H.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
19. Magnusson, M., Nüchter, A., Lörken, C., Lilienthal, A.J., Hertzberg, J.: Evaluation of 3d registration reliability and speed - a comparison of icp and ndt. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 3907–3912 (2009)
20. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. Robot. Auton. Syst. 56(11), 927–941 (2008)
21. Craighead, J., Murphy, R., Burke, J., Goldiez, B.: A survey of commercial and open source unmanned vehicle simulators. In: Proceedings of ICRA (2007)
22. Boeing, A., Bräunl, T.: Evaluation of real-time physics simulation systems. In: GRAPHITE 2007: Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia, pp. 281–288 (2007)

23. Wang, J., Lewis, M., Gennari, J.: Usar: A game-based simulation for teleoperation. In: Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, Denver, CO (October 13-17, 2003)
24. Greggio, N., Silvestri, G., Menegatti, E., Pagello, E.: A realistic simulation of a humanoid robot in usarsim. In: Proceeding of the 4th International Symposium on Mechatronics and its Applications (ISMA 2007), Sharjah, U.A.E (2007)
25. Rusu, R.B., Maldonado, A., Beetz, M., Systems, I.A., München, T.U.: Extending player/stage/gazebo towards cognitive robots acting in ubiquitous sensor-equipped environments. In: IEEE International Conference on Robotics and Automation (ICRA) Workshop for Network Robot System (April 14, 2007)
26. Hohl, L., Tellez, R., Michel, O., Ijspeert, A.J.: Aibo and Webots: Simulation, Wireless Remote Control and Controller Transfer. Robotics and Autonomous Systems 54(6), 472–485 (2006)
27. Petrinić, T., Ivanjko, E., Petrović, I.: Amorsim – a mobile robot simulator for matlab. In: Proceedings of 15th International Workshop on Robotics in Alpe-Adria-Danube Region, Balatonfüred, Hungary (June 15-17, 2006)
28. Buckhaults, C.: Increasing computer science participation in the first robotics competition with robot simulation. In: ACM-SE 47: Proceedings of the 47th Annual Southeast Regional Conference, pp. 1–4 (2009)
29. Craighead, J., Murphy, R., Burke, J., Goldiez, B.: A robot simulator classification system for hri. In: Proceedings of the 2007 International Symposium on Collaborative Technologies and Systems, pp. 93–98 (2007)
30. Craighead, J.: Distributed, game-based, intelligent tutoring systems – the next step in computer based training? In: Proceedings of the International Symposium on Collaborative Technologies and Systems (2008)
31. Huber, D., Hebert, M.: Fully automatic registration of multiple 3d data sets. Image and Vision Computing 21(1), 637–650 (2003)
32. Fitzgibbon, A.W.: Robust registration of 2d and 3d point sets. In: British Machine Vision Conference, pp. 411–420 (2001)
33. Magnusson, M., Duckett, T.: A comparison of 3d registration algorithms for autonomous underground mining vehicles. In: Proc. ECMR, pp. 86–91 (2005)
34. Nuchter, A., Lingemann, K., Hertzberg, J.: Cached k-d tree search for icp algorithms. In: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling, pp. 419–426 (2007)
35. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Third International Conference on 3D Digital Imaging and Modeling (3DIM) (2001)

Cognitive Supervision and Control of Robotic Inspection-Intervention System

Janusz Bedkowski[1,2] and Andrzej Masłowski[1,2]

[1]Industrial Research Institute for Automation and Measurements, Warsaw, Poland
[2]Institute of Automation and Robotics, Warsaw, Poland

Abstract. The main problem discussed in the following paper is control and supervision of web connected mobile robots. Taking up this subject is justified by the need of developing new methods for control, supervision and integration of existing modules (inspection robots, autonomous robots, mobile base station). The research problem comprises web connected mobile robots system development structure with exemplification based on inspection-intervention system. In the paper the application of semantic simulation engine is proposed. Semantic simulation engine combines semantic map (computer representation of cognitive map) automatically built based on robot's observations with rigid body simulation to perform its entities supervision. The results of experiments performed in real, virtual and hybrid environments are discussed. The methods are verified by exemplification based on a system composed of autonomous mobile robot and remotely controlled inspection robot equipped with robotic arm.

Keywords: supervision, control, robotic system, semantic mapping.

1 Introduction and Related Work

The main problem undertaken in the paper is control and supervision of web connected mobile robots, for example, inspection intervention robot system. The concept of cognitive model of human supervisor was described in [1]. Study of AI techniques applied for automated reasoning was introduced in [2]. Cognitive supervision is related to human machine interface described in [3], where problems with operating real mobile system is demonstrated. In this paper an improvement of cognitive supervision based on semantic simulation engine implementation is shown. The main applications of multi-robot inspection intervention system are actions in a disaster area, covering all the consequences of fires, chemical hazards, and the effects of a terrorist attack [4]. The environment of the system forces short time of inspection, and determines basic goals for the system. This provides clearly defined working conditions, the criteria for checking control and supervision algorithms. Many studies have shown extensive technical development in the area of mobile robotics. There have been many solutions [5] for technical issues related to unique mechatronics designs of high mobility mobile robots, dedicated for difficult terrain. In addition, number of robots equipped

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 140–149, 2011.

with sophisticated sensors increases, which enhances the effectiveness of search and detection of victims [6].

Cognition is related to semantic mapping. Semantic information [7] extracted from 3D laser data [8] is recent research topic of modern mobile robotics [9] [10] [11] [12]. In this paper new idea of semantic simulation engine used for robotic system supervision is proposed. Semantic simulation engine combines semantic map with rigid body simulation to perform supervision of its entities such as robots moving in INDOOR environment composed by floor, ceiling, walls, door etc. The approach is related to Augmented Reality techniques proposed in [13], where real and virtual-simulated part of mobile robotic system was integrated. Cognitive aspect is related to automatic detection of these entities based on robot's observations, the relations between them are defined by semantic net. To conclude this section it can be stated that many hardware and software techniques to implement sophisticated mobile robotic system are available, but at the same time the ergonomic solutions for HMI (Human Robot Interface) are needed to perform complex tasks. Based on this observation cognitive supervision and control can improve operator skills by replacing them in critical situations to avoid system failure.

The main contribution compared to previous work [1] [14] is developed 6D SLAM based on improved ICP (Iterative Closest Point) algorithm and loop closing using semantic images as a component of proposed in this paper semantic simulation engine. A GPU-based implementation of the standard 3D ICP algorithm is aligning two sets of points. The proposed solution is efficient since it performs nearest neighbor search using a bucket data structure (sorted using CUDA primitive) and computes the correlation matrix using parallel CUDA all-prefix-sum instruction.

The paper is organized as follows: Section *Problem formulation* describes the problem of cognitive supervision and control, Section *Multi robot system structure* describes the structure of robotic inspection intervention system and its components, Section *Cognitive model of human supervisor* shows the concept of cognitive model that supervises the system and controls when crisis occur, the cognitive map and its computer representation semantic simulation engine is described, also cognitive aspects are briefly discussed. Section *Experiments* shows main experiments performed in real and simulated environment, Section *Conclusion and future work* gives final discussion and shows areas of future work.

2 Problem Formulation

Robotic inspection intervention system is composed of mobile base station, autonomous mobile robot and a remote controlled inspection robot. System can be extended by another autonomous mobile robots to improve its functionality. The crisis in the system is defined as an event that can damage its components. Human operator is not able to fully supervise the system because of limited perception and focusing on task execution, therefore cognitive model of human supervisor is proposed. Cognitive model can intervene during crisis and take

control of the system to prevent hardware damage. Cognitive aspects are related to cognitive map (computer representation - semantic map) building. Based on robot's observations and semantic net new environment model is automatically generated and used for task execution supervision.

3 Multi Robot System Structure

The main object of research in this work is a web connected mobile robot system, for example, the inspection – intervention system consisting of a mobile base station, a group of autonomous robots and remote-controlled robot, equipped with a manipulator with n degrees of freedom. Figure 1 shows structure of such a system. The main tasks of the system is an inspection and intervention of

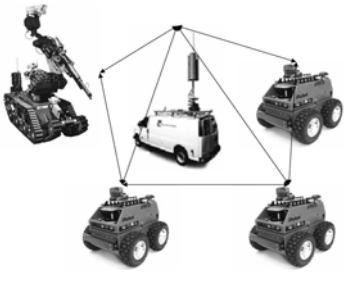


Fig. 1. Inspection-intervention system

hazardous environment for human activities. The considered system consists of three components: mobile base station, remotely controlled robot, autonomous mobile robot.

3.1 Mobile Base Station

Base station provides information concerning the robot's environment to the operator. It is equipped with HMI software that uses advanced graphical techniques for operator interaction with robots. An important problem for the operator is large quantity of information provided by the robot sensors, which can result in a problem with making quick and accurate decisions. During the routine work the system should assist the human operator, which will ensure proper operation despite operator's errors. For this purpose, a cognitive model of human supervisor is proposed, which solves some of the above-mentioned problems.

3.2 Remotely Controlled Robot

Main task of the robot is inspection and intervention in the disaster area. The robot is equipped with a video cameras, a manipulator with n degrees of freedom and communication system. The robot is able to partially replace the human in the environment that affects the health or even threatens human life.

3.3 Autonomous Mobile Robot

The robot's task is the inspection of the area of the disaster. For this purpose it is equipped with a laser measurement systems that provide 3D data of the environment, video camera, ultrasonic sensors for obstacles avoidance, local position system determining robot position. Robot can be, depending on the application, additionally equipped with chemical sensors or thermal camera.

4 Cognitive Model of Human Supervisor

From the standpoint of the system of web connected mobile robots there may occur a crisis situation that may lead to mission failure and even robots damage. For the purpose of this work crisis is an exceptional situation, which is interpreted by supervisor as a state of danger, potentially threatening the performance or safety of the system. Developed cognitive model of human supervisor is characterized by supervising the robot system showing similar reaction as human, in the event of an emergency. Therefore as it has the ability to recognize emergencies and the ability to generalize during making decisions, it is able to control the robots in a way similar to human. Developed cognitive model of the human supervisor is working with a distributed control system of mobile robots and has developed perception, which forms the basis of knowledge. Figure 2 shows the model diagram. Due to the potential loss of communication between the main inspection – intervention system components developed and implemented distributed cognitive model of the human supervisor of the robotic system is combining the elements of a centralized system and multi agent system. Figure 3 illustrates the idea behind the model. Multi agent cognitive architecture of the developed cognitive supervisor consists of three layers. The first layer is reserved for the most important in the hierarchy of agents - the cognitive supervisor of the robotic system. In the absence of communication problems the cognitive model works in a centralized system scheme, where agents from the lower layers, are fully subordinated to the execution of his orders. From the perspective of

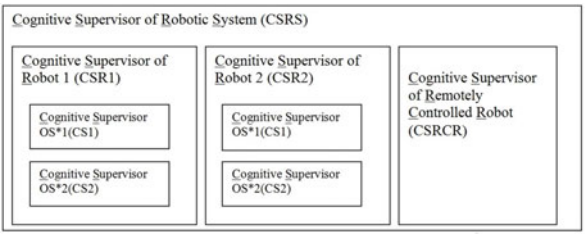


Fig. 2. Scheme of the distributed cognitive supervisor of the robotic system

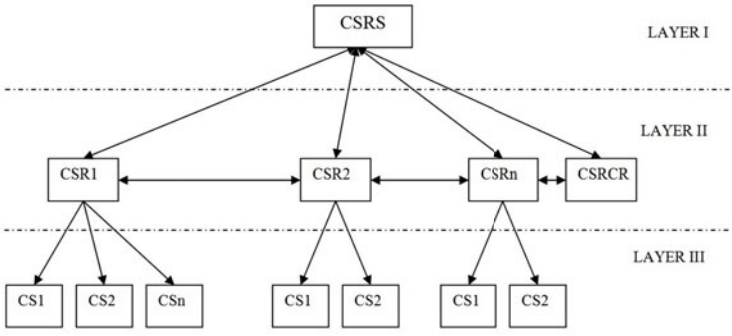


Fig. 3. Scheme of the distributed multi agent cognitive supervisor of the robotic system

software engineering in this case we are dealing with a distributed implementation of a single cognitive supervisor. Otherwise, if there are communication problems between layer I and layer II or within layer II, the agents from layer II are fully autonomous and operate as multi agent system. It should be noted that fault-free communication between the agents of layer II and layer III is assumed, as a result of using wired Ethernet communications. Cognitive architecture is strongly conditioned by supervising multi agent system where CSRS is installed at the base station, CSR1, CSR2, CSRn are installed on a mobile robot on-board computers, and coordinate the work of sub-CS1, CS2, CSn, which in turn, are installed on computational units of the robot, overseeing the work of operating systems. Additionally, in the architecture in Figure 3 CSRCR is an agent supervising the remote-controlled robot, assuming that the robot is equipped with a suitable on-board computer.

4.1 Cognitive Map

Cognitive map is a result of the sum of observations made with the use of robot perception. The cognitive map is a source of information for the cognitive model, on the state of the system, including robots, the environment and mission. On this basis the cognitive model of human supervisor is able to detect danger and possibly intervene. The result of perception, including receiving information and processing, and then thinking in terms of transformation of this information, is the cognitive map(figure 4). The cognitive map can be assumed as some form of description of the world. On the basis of sufficient information about the cognitive model of the world may decide to effect behavioral or cognitive. Behavioral effect is directly associated with making a particular action (following path movement of the robot), while the cognitive effect creates new structures, which in terms of machine learning can mean the extension of the decision tree, or creating a new rule-making, and for the monitoring of the emergence of the mission new graph of the mission.

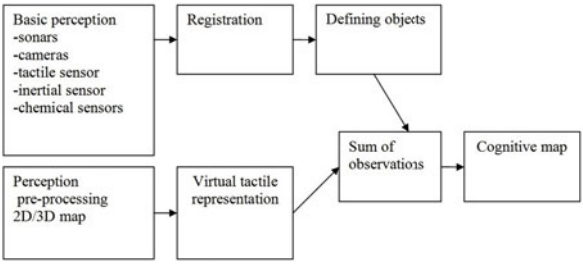


Fig. 4. Scheme of cognitive map building

4.2 Semantic Simulation Engine

The concept of semantic simulation engine is a new idea, and its strength lies in the semantic map integration with mobile robot simulator. Semantic simulation engine is composed of data registration modules, semantic entities identification modules and semantic simulation module. It provides tools to implement mobile robot simulation based on real data delivered by robot and processed on-line using parallel computation. Semantic entities identification modules can classify door, walls, floor, ceiling, stairs in indoor environment. Data can be delivered by robot observation based on modern sensors such as laser measurement system 3D and RGB-D cameras. It can be stated that semantic simulation engine is related to computer representation of cognitive map, where real objects are associated with virtual entities of simulated environment. Assuming that we have accurate positioning of mobile robots, accurate encoders of inspection robot arm and satisfying tracking system we can update virtual entities position during real robots task execution. The semantic simulation engine from semantic map of INDOOR environment point of view is composed of: semantic map nodes(entities) L_{sm}={Wall, Wall above door, Floor, Ceiling, Door, Free space for door, Stairs...}, it is important to noticed that the L_{sm} set can be extended by another objects, what is dependent on robust and accurate 3D scene analysis, semantic map relationships between the entities R_{sm} = {parallel, orthogonal, above, under, equal height, available inside, connected via joint...}, semantic map events E_{sm} = {movement, collision between two entities started, collision between two entities stopped, collision between two entities continued, broken joint...}.

From robot simulation in NVIDIA PhysX stand point the entities from semantic map are related to actors in PhysX. Spatial models generated based on registered 3D scans i.e. walls, doors and stairs are transformed into actors with BOX shapes. It is important to emphasize that following events can be noticed during simulation: robot can touch each entity, open/close the door, climb the stairs, enter empty space of the door, damage itself (broken joint between actors in robot arm), brake joint that connects door to the wall. It is noteworthy to mention that all robot simulator semantic events are useful for operator

supervision, where computer/cognitive model has to monitor simulation events judge them and report for the control purpose.

4.3 Cognitive Aspects of Supervision and Control

Computer representation of cognitive map - semantic map is composed of entities and relations between them. Semantic simulation engine executes supervision of entities by notification events such as intersection, robot pitch/roll etc. The cognitive aspects of supervision and control are related to automatic semantic map generation based on robot's observations and predefined semantic net. Figure 5 is showing initial stage of robotic system working in unknown environment. Cognitive model supervises all hardware components and mission planning (shown as goals-flags). Figure 5 right is showing path without collision because semantic map is not yet built. Cognitive aspect will be semantic map used by semantic simulation engine to prevent entities intersection and robot pitch/roll. It is important to emphasize that semantic simulation engine is not limited to intersection computation. It can also supervise which entity is inside another and how long (see figure 9 where robot is in the empty space of door). Semantic simulation engine provides necessary information about robot behavior and potential risks that can determine crisis.

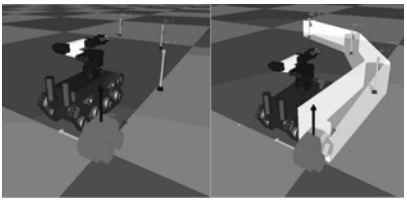


Fig. 5. Cognitive supervision in 3D environment. Initial stage of inspection intervention system in unknown environment. On the right path without collision is shown.

5 Experiments

Cognitive model of human supervisor was tested in several crisis situations. Figure 6 shows the result of robot arm supervision where intersection of arm elements is prevented and marked with red color. Concerning autonomous mobile robot supervision of robot pitch/roll and collision with obstacle is shown on figure 7. Path planning in unknown environment represented by semantic map built based on robot observations is shown in figure 8. Cognitive model will not allow defining path that lead to collision with obstacle, it is shown as red rectangular prism in right image of figure 8. The semantic map built based on autonomous robot observations (3D laser measurement system) is shown in

figure 9. It should be emphasized that semantic simulation engine was used to simulate robot climbing stairs with pitch and roll supervision and robot entering door. The main contribution compared to previous work published in [1] and [14] is developed 6D SLAM based on improved ICP (Iterative Closest Point) algorithm and loop closing using semantic images. The result is shown on figure 10. A GPU-based implementation of the standard 3D ICP algorithm is aligning two sets of points iteratively. The proposed solution is efficient since it performs nearest neighbor search using a regular bucket data structure (sorted using CUDA primitive) and computes the correlation matrix using parallel CUDA all-prefix-sum instruction. The accuracy and convergence of positioning and map building algorithm is very promising and obtained on-line capability will improve presented cognitive supervision.

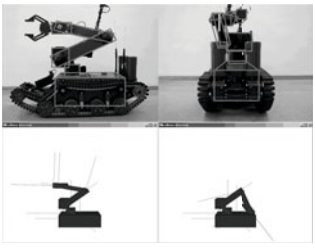


Fig. 6. Supervision of robot arm configuration

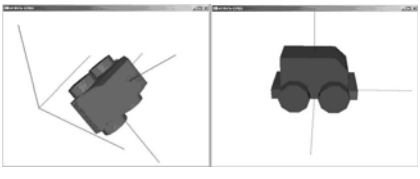


Fig. 7. Supervision of robot pitch/roll and collision with obstacle

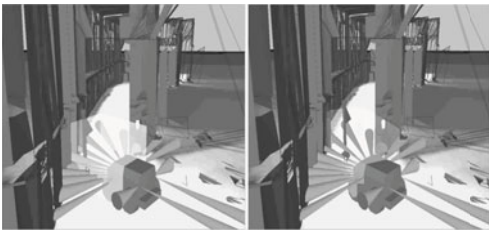


Fig. 8. Supervision of path planning. Cognitive model will not allow defining path that effects collision with obstacle (red rectangular prism on right image).

Fig. 9. Left - Climbing the stairs, pitch and roll supervision. Right - Entering opened door.

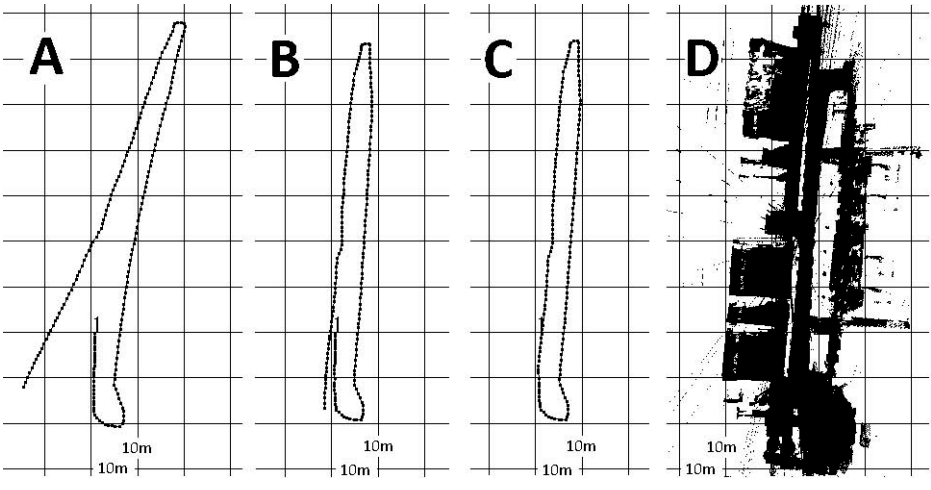


Fig. 10. Result of 6D-SLAM. A-trajectory with odometry, B-trajectory with GPGPU-based ICP, C-trajectory after loop closing, D-final 3D map.

6 Conclusion and Future Work

In the paper an improvement of cognitive supervision and control of robotic inspection intervention system is shown. The new idea is a semantic simulation engine that uses semantic map to prevent crisis in the system. Semantic map is constructed in unknown INDOOR environment based on robot observations. It is used for entities supervision (intersection, pitch, roll, etc.). An extension of intersection technique is used for path planning supervision, where cognitive model will not allow defining goal that can effect with a crash with an obstacle. It is important to realize that the complexity of the problem determines difficulty of supervision in dynamic environment. The techniques of dynamic objects-entities tracking are needed to update its 6D position in semantic simulation engine.

References

1. Bedkowski, J.: Methodology of control and supervision of web connected mobile robots with cuda technology application, Ph.D. thesis, Warsaw University of Technology (2010)
2. Bedkowski, J., Maslowski, A.: The hierarchical knowledge representation for automated reasoning. In: 15th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, August 23-26. CD-ROM (2010)
3. Bedkowski, J., Kowalski, P., Piszczek, J.: Hmi for multi robotic inspection system for risky intervention and environmental surveillance. In: Mobile Robotics: Solutions and Challenges, Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Istanbul, Turkey (September 9-11, 2009)
4. Bedkowski, J., Kowalski, P., Piszczek, J., Maslowski, A.: ATRVJr – mobile robot for fire fighter services. Techniki Komputerowe (1), 25–33 (2009)
5. Casper, J., Murphy, R.R.: Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. IEEE Transactions on Systems, Man and Cybernetics. Part B 33, 367–385 (2003)
6. Cubber, G.D., Marton, G.: Human victim detection. In: International Workshop on Robotics for risky interventions and Environmental Surveillance, RISE 2009, Brussels, January 12-14. CD-ROM (2009)
7. Asada, M., Shirai, Y.: Building a world model for a mobile robot using dynamic semantic constraints. In: Proc. 11 th International Joint Conference on Artificial Intelligence, pp. 1629–1634 (1989)
8. Nüchter, A., Wulf, O., Lingemann, K., Hertzberg, J., Wagner, B., Surmann, H.: 3d mapping with semantic knowledge. In: Robocup International Symposium, pp. 335–346 (2005)
9. Nüchter, A., Hertzberg, J.: Towards semantic maps for mobile robots. Robot. Auton. Syst. 56(11), 915–926 (2008)
10. Grau, O.: A scene analysis system for the generation of 3-d models. In: NRC 1997: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, p. 221 (1997)
11. Cantzler, H., Fisher, R.B., Devy, M.: Quality enhancement of reconstructed 3d models using coplanarity and constraints. In: Proceedings of the 24th DAGM Symposium on Pattern Recognition, pp. 34–41 (2002)
12. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. Robot. Auton. Syst. 56(11), 927–941 (2008)
13. Bedkowski, J., Maslowski, A.: Augmented reality approach for mobile multi-robotic system development and integration. Techniki Komputerowe (1), 15–24 (2009)
14. Bedkowski, J., Maslowski, A.: Methodology of control and supervision of web connected mobile robots with cuda technology application. Journal of Automation, Mobile Robotics and Intelligent Systems 5(2) (2011)

Declarative Design of Control Logic for Mindstorms NXT with XTT2 Method

Grzegorz J. Nalepa and Błażej Biesiada

AGH University of Science and Technology,
Al. A. Mickiewicza 30, 30-059 Krakow, Poland
gjn@agh.edu.pl

Abstract. The principal idea advocated in this paper is to design the control logic for the mobile robot using the XTT2 representation, and the execute it with a dedicated inference engine. XTT2 is a declarative rule-based representation, that has a strict logical description. It also provides a high level visual representation combining decision tables into an inference network. In the paper a dedicated API for Mindstorms NXT is also presented. It is combined with HeaRT rule engine running the XTT2 model to control a mobile robot.

Keywords: intelligent control, mobile robots, Mindstorms NXT, XTT2.

1 Introduction

Building intelligent robots has always been an important area of both pursuit and research in Artificial Intelligence [6], and applied engineering. Creating such robots requires skills from different domains, including deep knowledge of materials and mechanics, as well as control theory, artificial intelligence, computer science, and even psychology and linguistics, when we take human-machine communication into account. However, these days the field became much more accessible to non-experts, thanks to number of ready robotics solutions. These include some ready robots, that can be bought and trained, e.g. SONY Aibo, or iRobot Roomba. Recently, a new release from the LEGO company improved this situation even further.

LEGO Mindstorms NXT is a universal robotics platform, that offers advanced robot construction possibilities, as well as sophisticated programming solutions. The new version of Mindstorms is becoming a standard robotics platform for both teaching and rapid prototyping of robots. Numerous programming solutions for NXT exist, including the LEGO environment, LeJOS, Bricx/NQC and others. However, they fail to provide a clean high-level declarative logic programming solution for NXT. Programming robots, especially mobile ones, is a complex task, involving some typical AI problems, such as knowledge representation and processing, planning, etc. These areas are much more accessible with the use of a logic programming solutions, compared to classic, low-level imperative languages.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 150–159, 2011.

The main objective of the paper is to present an application of the HeKatE approach [5] to controlling mobile robots. Thanks to its openess, LEGO Mindstorms NXT has been chosen as a prototyping hardware platform. A new Prolog-based API for controlling Mindstorms NXT is introduced in Section 3. The API uses a multilayer architecture, composed of a behavioral, sensomotoric, and connection layer. This platform can be used as a generic solution for programming the NXT in Prolog [3]. It also serves as a foundation for a higher-level visual rule-based programming with XTT2 method. The principal idea includes visual design of the control algorithms for mobile robots with XTT2. Then, the algorithms are translated to HMR code which is executed by HeaRT engine using a set of custom callbacks to call PLNXT. The complete control architecture is presented in Section 4. Then, in Section 5 practical examples of control algorithms are described. Comparing to the available environments, the PLNXT + XTT2 platform provides a clean and transparent programming solution.

2 Mindstorms NXT

LEGO Mindstorms NXT is the second generation of programmable robotics kit released by LEGO, it is a successor to LEGO Robotics Invention System (also known as LEGO RCX). Since the initial release the product received a lot of attention from an academic world. This is of no surprise, since LEGO actively supports efforts of employing its products in a multitude of high schools and universities across the globe. For an example the NXT kit was created by a partnership with MIT Media Laboratory. Most importantly, the platform proved to be a perfect solution for easy and rapid prototyping of both hardware and software robotic designs.

Over the years LEGO has released several variations of the Mindstorms NXT kit, with an NXT 2.0 being the most recent one (released in 2009). Having said that, there are several core elements of the set that can be almost always found in the box: Brick, Servo motors, Sensors, Cables, LEGO Bricks, and a Test pad. The *Intelligent Brick* is an integrated embedded computer platform [1].

The Brick comes with a preinstalled LEGO firmware, which is capable of executing multithreaded applications. Also, LEGO SDK provides a very convenient integrated development environment called NXT-G for Graphical based on a LabVIEW platform[1]. Being easy to use and intuitive it is a good starting point, but it also imposes a few limitations, which led to a creation of many alternatives. From the runtime point of view, these solutions can be categorized into solutions that: communicate with the Brick using the LEGO protocol, provide a higher level language that compiles to Brick bytecode, replace the Brick firmware with a custom one.

The first approach is a simple, clean and straightforward one. The examples of the first group include LeJOS iCommand `http://lejos.sourceforge.net`, or NXT++ `http://nxtpp.sourceforge.net/`. The second approach requires a dedicated complier, which makes it more complicated. In the second group there

[1] See `http://www.ni.com/labview`

exists number of solutions including NXC (http://bricxcc.sourceforge.net/nxc), or RoboLab (http://www.ceeo.tufts.edu/robolabatceeo/). The third solution is the most complicated one, since it requires developing a dedicated embedded operating system. This type of solution is provided by the Java-based LeJOS (http://lejos.sourceforge.net).

Another flexible approach to robot programming is to use a high-level declarative language such as Prolog instead of low-level C-like, or Java-based programming. Some early attempts are within the LegoLog Project [2] (http://www.cs.toronto.edu/cogrobo/Legolog). Unfortunately the project did not offer a general API, and supported only the older Mindstorms RCX version. Besides basic programming languages, NXT robot programming can be supported on a higher logical level, offering a visual logic representation. The prime example is the default LEGO environment. In these cases the control logic is represented with use of flowcharts representing the control algorithm. However, this is mainly a procedural representation, not a declarative one.

3 PLNXT Library

Using knowledge representation methods from the classic AI, such as the decision rules, rules, and tables could improve NXT programming options. XTT2 offers a generic rule-based visual programming solution, combining the power of decision tables and decision trees. XTT2 is implemented with the use of a Prolog-based inference engine. Providing a Prolog-based API for Mindstorms NXT allows to develop control logic for NXT robots with the use of the XTT2 method.

Basing on the review of existing solutions presented above, requirements of a new Prolog API for NXT has been formulated [3]. The main requirements are:

- support for all functions of the standard NXT components, that is sensors and motors,
- crossplatform solution, for both Windows and GNU/Linux environments,
- integration with the visual rule-based logic design with XTT2.

The complete solution is ultimately composed of:

- PLNXT, a middleware executed on a PC, controlling an NXT-based robot, the control is performed with the use of the Bluetooth or USB cable connection,
- a set of custom callbacks for HEART bridging it with PLNXT,
- HEART running a control logic in HMR for a mobile robot.

A more detailed design of the PLNXT API is presented next.

Considering the requirements the following API architecture has been designed. It is composed of three main layers as observed in the Figure 1:

behavioral layer providing a higher-level functions, e.g. drive.

sensomotoric layer allowing the exchange information with sensors and motors,
communication layer providing the low-level communication with the robot,

The behavioral layer (*nxt_movement*) exposes to the programmer some-high level functions and services. It provides abstract robot control functions, such as *go*, or *turn*. Ultimately a full navigation support for different robot types can be provided. However, different robot configurations require different control logic (e.g. to move the robot).

The sensomotoric (*nxt_sensomoto*) layer controls the components of the Mindstorms NXT set motors, all the sensors, as well as Brick functions. This layer can be used to directly read the sensors, as well as program the motors. It is can be used by a programmer to enhance high-level behavioral functions.

The goal of the communication layer is to execute the actions of the sensomotoric layer and communicate with the NXT Brick. Currently in this layer several modules are present, providing different means of communication:

- a pure Prolog module, using a serial port communication, and the NXT protocol commands,
- a hybrid solution based on the Java-based iCommand library,
- a hybrid socket-based solution, using the NXT++ library, that communicates with the robot.

All of these actually wrap the *Mindstorms NXT Communication Protocol*. The first solution is the most straight forward one, with standard ISO Prolog stream predicates used to control the serial port. In the second case the Prolog communication module is integrated with iCommand with the use of the SWI Java to Prolog interface called JPL. In the third case, a simple server written in C++ exposes NXT communication with a TCP socket. The Prolog communication module connects to the server and controls the robot through a TCP connection. This opens up a possibility of a *remote* control, where the controlling logic is run on another machine, or even machines.

Besides some basic send/receive functions the library has to provide certain services. These are event and time-based callbacks implemented. So the library has to provide *timers* that trigger some callbacks, as well as *event-driven* callbacks. This requires *parallel* execution of certain threads.

Currently a prototype SWI-Prolog implementation of the API is available online[2]. Movement functions are implemented in the highest layer. The mid layer provides full control over robot's sensors and motors and exposes timer and event services. In the low-level communication layer, the iCommand, DirectSerial, and NXT++ communication modules are implemented.

Below, a simple example of a console communication with a robot is provided:

```
1 % Establishing connection.
2 ?- nxt_open.
3 % Moving forward at a speed of 300 degrees/second.
```

[2] See http://ai.ia.agh.edu.pl/wiki/plnxt

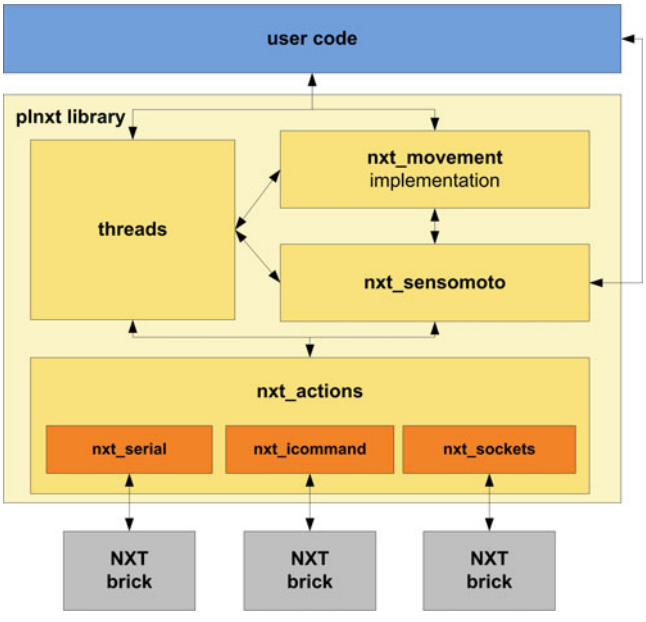


Fig. 1. PlNXT layered architecture

```
?- nxt_go(300).
% Stopping robot.
?- nxt_stop.
% Moving 80 cm forward at a speed of 400 degrees/second.
?- nxt_go_cm(400,80).
% Reading touch sensor.
?- nxt_touch(Value).
% Reading light sensor.
?- nxt_light(Value).
% Turning the light sensor diode on.
?- nxt_light_LED(activate).
% Reading ultrasonic sensor.
?- nxt_ultrasonic(Value).
% Rotating 360 degrees to the right at a speed of 350
    degrees/second.
?- nxt_rotate(350,360).
% Playing tone at frequency 500 Hz for 2000 ms.
?- nxt_play_tone(500,2000).
?- nxt_close.
```

The Prolog API has been successfully tested on number of simple control algorithms. The complete solution combibing PLNXT with HEART is discussed in the following section.

4 Rule-Based Control with XTT2 and HeaRT

The principal idea advocated in this paper is to design a control logic for a mobile robot using the XTT2 representation, and execute it with the dedicated inference engine, HeaRT. XTT2 is a declarative rule-based representation, that has a strict logical description. It also provides a high level visual representation combining decision tables into an inference network. The prototype of the network can be built using the ARD method.

First of all, a conceptual ARD+ diagram is designed in order to define robot's attributes. Next a rule-based XTT2 model, based on the conceptual project, is created with the HQEd visual editor [5]. Then, it can be exported to an executable HMR code. Finally, an HMR file can be executed with the HeaRT interpreter [4], using one of the available inference modes (algorithms presented in next section utilize the Token-Driven Inference). The interpreter will continue to execute the model in a loop as long as it is traversable.

Communication with an external environment, like reading sensors states or setting motors speeds, is performed with a set of predefined synchronous callbacks and actions. This technique enables HeaRT to talk to PlNXT library, which connects with NXT bricks via USB and/or Bluetooth (more than a one interface can be used if there is more than a one brick to work with). Also, there is no hindrance to make HeaRT utilize multiple completely different libraries simultaneously, which makes this mechanism a truly powerful tool.

Most of PlNXT predicates that are suitable for usage in XTT2 models have been directly (in terms of names) translated into HeaRT callbacks and actions. As a consequence, anybody who is already familiar with PlNXT will be very comfortable with them. All of the linked predicates come from *Movement* and *Sensomoto* layers of the library. Below are few samples of the implemented callbacks (`xcall`) and actions (`xactn`).

```
xcall plnxt_motor :[Motor,Speed]
      >>> (alsv_values_get(current, Motor, MotorVal),
           nxt_motor(MotorVal,SpeedVal,force),
           alsv_new_val(Speed, SpeedVal)).
xactn plnxt_motor :[Motor,Speed]
      >>> (alsv_values_get(current, Motor, MotorVal),
           alsv_values_get(current, Speed, SpeedVal),
           nxt_motor(MotorVal,SpeedVal,force)).
xcall plnxt_light :[Brightness]
      >>> (nxt_light(Val, force),
           alsv_new_val(Brightness, Val)).
xactn plnxt_light_LED :[Setting]
      >>> (alsv_values_get(current, Setting, Val),
           nxt_light_LED(Val, force)).
```

The next section contains an example of PlNXT + XTT2 robotic control system in order to demonstrate an ease of use and vast capabilities of this platform.

5 Examples of Control Algorithms

This section describes an *evader robot* which should implement three kinds of behaviors. First of all, it is supposed to patrol its environment with random trajectories. The environment is an area limited by walls and diversified with various obstacles, all of which may damage the robot in case of an impact, therefore they have to be avoided. Lastly, when the robot hears a loud sound it should run away, which simply means moving straight with an increased speed in any random direction. Those three behaviors are of different significance, as a consequence the *Patrol* behavior can be interrupted by either *Run Away* or *Avoid Wall* actions (behaviors). The Run Away behavior can be cancelled only by Avoid Wall action. And, as might be expected, the Avoid Wall behavior can not be cancelled by any other action.

Robot's construction is fairly simple, it utilizes Lego's Tribot design, two independent motors controlling movement speed and direction (each connected to one wheel). The evader robot uses two different sensors to interact with its environment, both of which are encapsulated within `Sensors` attribute in the conceptual model. Since behaviors required from this robot are quite complex there is also an additional conceptual attribute called `Behaviors`, representing the logic responsible for evaluating an appropriate action based on current sensors readings. Physical attributes of the robot are:

- `distance` (to an obstacle or a wall)
- `loudness` (an overall volume level in robot's environment)
- `avoid_wall` (behavior of avoiding walls and obstacles)
- `run_away` (running away behavior)
- `patrol` (patrolling behavior)
- `turn` (how fast the robot is turning, if at all)
- `speed` (robot's speed)

Figure 3 illustrates the last level of the conceptual design, how `loudness` and `distance` influence all the behavior attributes and how `turn` and `speed` are controled. History of transformations that were performed to evaluate the conceptual model all the way down to a physical level are shown in Figure 2.

Robot's logic model (Figure 4) can be divided into two key parts. The first one reads data from sensors (tables `Loudness Decision` and `Distance Decision`) and then decides what action to take next (table `Behavior`). The second part is responsible for executing an action chosen by the predecessor:

- **Avoid Wall** This behavior can be considered as the most self-preservational, the main goal here is to avoid hitting the walls and the obstacles (all of which have to be detectable by the ultrasonic sensor). This is achieved by either slowing down if an obstacle is still far from the robot or moving backwards if it is very close to the robot (table `Adjust Speed`). In both cases turn is randomly adjusted (table `Wall Avoidance`), there is 50% chances that the robot will start to turn right, and similarly 50% for turning left.

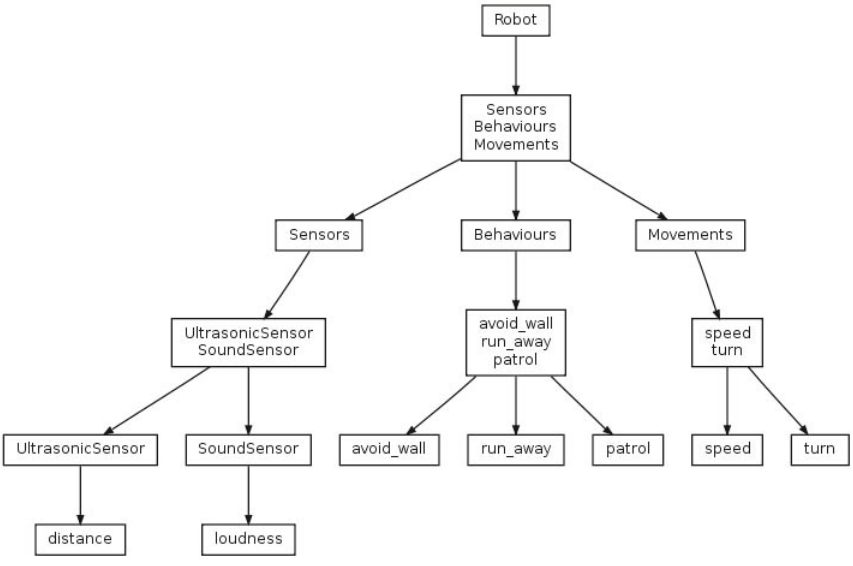


Fig. 2. A history of the conceptual design transformations of the evader robot

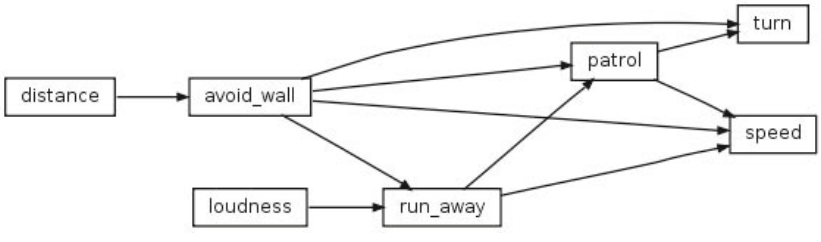


Fig. 3. The last conceptual ARD level of the evader robot

- **Run Away** The second most self-preservational behavior is `run_away`, triggered by sensing a loud sound. Since Lego NXT's sound sensor is omnidirectional and a source of a loud sound can not be easily established, the implementation of this behavior is highly simplified. It comes down to one table (`Run Away Speed`) with only one rule for setting robot's speed. The `run_away` behavior can can interrupt an execution of the `patrol` action, but on the other hand its execution can be cancelled by the `avoid_wall` action.
- **Patrol** When robot's sensors do not percept anything unusual `patrol` action will be executed. It consists of two different tables, one for modifying speed (table `Patrol Speed Modification`) and the other one for modifying

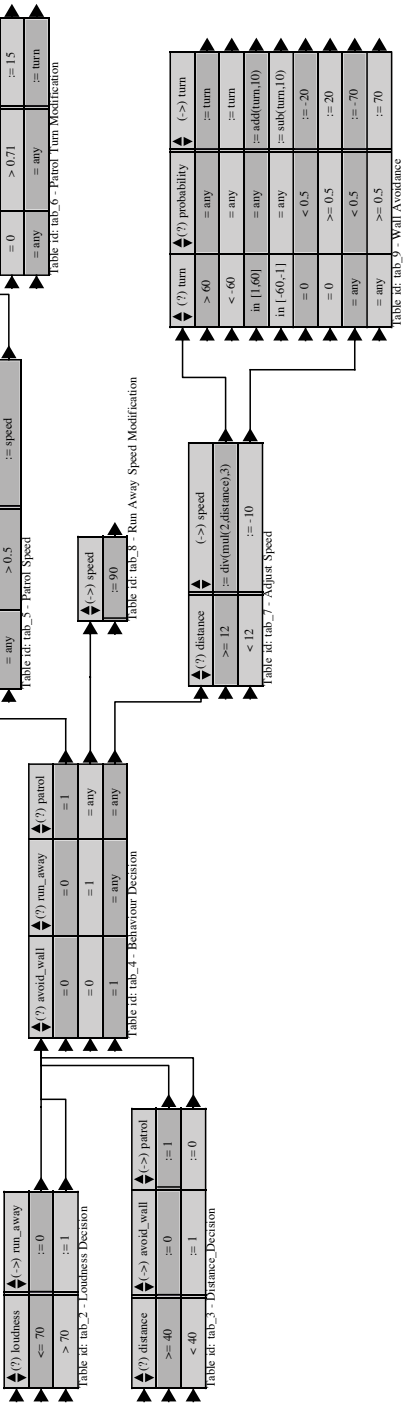


Fig. 4. The XTT2 logic model of the evader robot

direction (table `Patrol Turn Modification`). With each control cycle both parameters are randomly changed and there is also a chance that they will untouched.

In this example randomization was introduced into the XTT2 model in order to add human-like behaviors. This task was rather effortless thanks to the HeaRT's callbacks/actions mechanism, just one Prolog predicate was sufficient to do the job. Similarly many other Prolog standard predicates can be very easily adopted with this architecture.

6 Concluding Remarks

In the paper a design framework for mobile robots on the Lego Mindstorms NXT platform is presented. It uses a high-level rule-based method called XTT2 to represent control logic in a visual way. The logic can then be run by a dedicated inference engine called HeaRT that controls the robot. This approach is superior to the default design environment delivered with the LEGO platform. To control algorithm is represented on a logical level.

Future works includes more testing and performance improvements of the platform. Multirobot communication support is also considered. Finally, an integration with an cognitive architecture for a high-level control is also planned.

Acknowledgments. The paper is supported by the AGH UST Grant 11.11.120.859.

References

1. The LEGO Group: LEGO MINDSTORMS NXT Communication Protocol, 1.00 edn (2006)
2. Levesque, H.J., Pagnucco, M.: Legolog: Inexpensive experiments in cognitive robotics. In: Proceedings of the Second International Cognitive Robotics Workshop, Berlin, Germany (2000)
3. Nalepa, G.J.: Prototype Prolog API for Mindstorms NXT. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 393–394. Springer, Heidelberg (2008)
4. Nalepa, G.J.: Architecture of the HeaRT hybrid rule engine. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6114, pp. 598–605. Springer, Heidelberg (2010)
5. Nalepa, G.J., Ligęza, A.: HeKatE methodology, hybrid engineering of intelligent systems. International Journal of Applied Mathematics and Computer Science 20(1), 35–53 (2010)
6. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)

Planning in Collaborative Stigmergic Workspaces

Constantin-Bala Zamfirescu[1] and Ciprian Candea[2]

[1] "Lucian Blaga" University of Sibiu, Faculty of Engineering, Department of Computer Science and Automatic Control, Emil Cioran 4,
69121 Sibiu, Romania
zbc@acm.org
[2] Ropardo SRL, "Research and Development" Department, Reconstructiei 2A
550129, Sibiu, Romania
ciprian.candea@ropardo.ro

Abstract. The paper investigates how the engineered capabilities of structuring the knowledge encoded in collaborative workspaces affect the collective intelligence of its users. The investigation is made for the particular case of collaborative planning and is grounded on the theoretical framework of stigmergic systems. The knowledge structure encoded in collaborative workspaces in the form of a conceptual hierarchical task network is analysed by building a multi-agent simulation to evaluate the performance of different planning strategies. The results show that different representational complexities of collaborative planning knowledge have a great impact over the collective intelligence when the users are interacting directly or indirectly.

Keywords: collaborative planning, stigmergic systems, agent-based simulation.

1 Introduction

In the last decade the swarming models of computation become a popular engineering approach in designing decentralized systems that run in open and dynamic environments. Inspired primary from the behaviour of social insects, the concept covers a large set of indirect coordination mechanisms that generate an intelligent collective behaviour. One of the most cited examples is how the ants are building through the pheromone trails put in the surrounding environment their collective mental maps that guide the ants to find, without any central coordination, the shortest path from the nest to the food source. The coordination among the ants is realized by using simple *stigmergic coordination mechanisms*, such as aggregation of preferences, positive and negative feedback [1].

The stigmergic coordination mechanisms are not restricted solely to the use in engineering simple (reactive) agents, but may be recognized in several examples of social activities [2, 3]. Stigmergy is the coordination mechanism employed in many collaborative working environments such as wiki, Google's PageRank system, eBay's online auctioning, Amazon's recommender systems, Wikipedia.org, and Second Life [4]. In these applications, humans are exploiting the advantages of the stigmergic

P. Jędrzejowicz et al.(Eds.): ICCCI 2011, Part II, LNCS 6923 , pp. 160–169, 2011.

coordination mechanisms to prevent their cognitive effort to be exposed to the complexity of an open and dynamic environment [5].

All these advantages make the stigmergic models of computation a fast-spreading approach to engineer collaborative working environments that support complex problem solving. Given that the essential capability of any stigmergic system is to transfer the cognitive complexity from the humans to the environment [6], the problem-solving capabilities of the users decisively depend on how the problem is represented in the digital environment. A standard representation of the problem in the environment is realized as a composition of cognitive artifacts linked in a weighted graph [3]. Basically, this graph signifies a navigation map that supports the cognitive planning effort to reach any artifact from the current place. The artifacts commonly stand for the possible actions to solve the problem, while the links are the set of possible selections that guide the planning process (the conceptual navigation) from one state to another. Like the intensity of pheromone trails in the case of real ants, these actions are weighted in order to discriminate the most effective ones.

The paper analyses the collective intelligence in respect to the ability to structure the problem's representation in the digital environment. The digital environment is seen as a stigmergic environment that mediates the interactions among participants in collaborative planning from two different perspectives: diachronic – when the interactions are indirectly influenced by the users' past actions over the problem's representation in the digital environment; and synchronic – when the direct interactions among participants are driven by the problem's representation in the digital environment. The remaining part of this paper is organized as it follows. The next section presents the basic concepts of collaborative planning when the users are exploiting the digital environment to represent the problem knowledge. Section 3 describes the main components of the agent-based simulation (the computational model) used to investigate the impact of structuring the knowledge encoded in collaborative workspaces over the users' collective intelligence. The simulation results are depicted in Section 4. The paper concludes with some remarks regarding the engineering issues to support the collective intelligence of peoples who are using a collaborative working environment to represent and solve a planning problem.

2 A Stigmergic Framework for Collaborative Planning

A common way to support the cognitive process of constructing a solution for a problem is to externalize it in the digital workspace into a semantically-rich structure or a collective mental map [7]. Thus, the common workspace is made of cognitive artifacts which work as mediators for structuring the collective solution of a problem [8]. Structurally, the *topology* of the cognitive artifacts may be viewed as a fully connected weighted graph that codifies the domain knowledge (*the problem space*). This knowledge are dynamic in their nature and are created and refined in time by all the users who are indirectly interacting through the working space. For many reasons, when solving a problem it is completely impractical (due to many constraints such as time, expertise, etc.) for the entire community of users to be directly involved in all the types of problem-solving activities or for a group of users to exhaustively explore the problem space. Consequently, when one user is trying to construct a solution he/she will explore only a small part of the problem space (*solution space*).

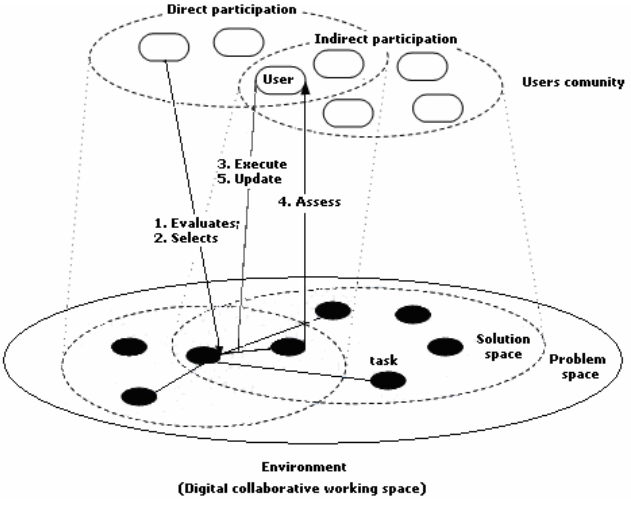


Fig. 1. The collaborative workspace as a stigmergic environment for collaborative planning

In this context, the construction of a solution consists in applying a heuristic to select the proper sequence of *tasks* or actions from the problem space that lead as efficiently as possible to the desired state. The solution quality depends naturally on the heuristic used to select the most promising tasks that lead to a goal. Thus, when the users are collaboratively constructing a solution for a problem there are two possible ways of interactions: *direct* (when the participants are confronting their (partial) solutions against some performance criteria and, consequently, there is a direct influence of a user over the collective solution of the group engaged in the problem solving) and *indirect* (when the users are constructing their solutions on the basis of the past assessments of the task's utilities for the current problem). Consequently, at a certain point in time during the solution design, a user is conceptually "located" in a node (a state after the execution of a task) of the problem space, performing one of the following basic actions (Fig. 1): 1) *evaluates the preferences* for the next possible tasks that may be executed from the current state; 2) *selects* the next best task for further completing the solution model; 3) *executes* the task from the model; 5) *assess the performance* for the executed task, and finally 6) *update the preference* recorded on the link between the previous executed two tasks. Note that these actions are not necessarily restricted to a single task and may be extended to any sequence of tasks.

3 The Computational Agent-Based Model

Envisioning the collaborative working space populated with cognitive artefacts as a stigmergic environment to support the solving of a planning problem, we developed an agent-based computational model that mimics the users' conceptual 'navigation' over the semantic structure of the knowledge encoded in the problem space [9]. As for any stigmergic system this model entails the description of agents' behaviour and the

structure of the shared environment [3]. For the collaborative planning process, the agents are the users responsible to design, execute, and evaluate a plan (i.e., to find a path through the conceptual space of the available tasks represented in the problem space), while the environment is the collaborative working environment that encode all the tasks discovered and documented by the users' community. Note that the primitive decomposition unit for a plan may be any conceptual artifact (a graphical representation of a task) that is used to design the structure/composition of a plan.

3.1 The Conceptual Structure of the Environment for Collaborative Planning

According to Parunak [3], a stigmergic environment presumes the definition of three main components: 1) topology, 2) states, and 3) processes. Structurally, the *topology* may be viewed as a fully connected weighted graph that codifies the knowledge of planning in a specific domain. Practically this structure reassembles a hierarchical task network [10], a framework that provides a natural way to formalize the knowledge used in many real-life applications [11]. Hierarchical task networks are used to encode large planning problems where the abstract tasks can be recursively decomposed into less abstract tasks, with executable tasks at the lowest level. Thus, the structure of a hierarchical task network allows a great flexibility to encode the interdependencies among tasks and alternatives of task decompositions. While in literature there are described various centralized hierarchical task planners, such as SHOP, SIPE and O-Plan [10], there is only one notable experiment that relates to our general framework of stigmergic collaborative planning which try to solve the planning problem by employing the swarming models of computation to an external representation of the task network. Anyway, for the purpose of this paper the hierarchical task network theory is used only as a source of inspiration to model and encode the users' interaction with the procedural knowledge and not to solve the hierarchical task network problem. On this basis, we made some assumptions to simplify the implementation of the agent-based simulation. Firstly, a task represents correlated information among an action and the resource used to perform that action. Consequently, there are multiple different tasks for a single action depending on the required resources (for example to rich one place from another you may use different transportation means such as bus, car, etc.). This is encoded in a particular variable of each task through a variable specifying the task-type. Secondly, in the graph there are represented only primitive tasks (executable), their structuration in more abstract task being considered to be part of user's intervention during the planning process.

In addition to the above clarifications regarding the nodes encoded in the problem space, a stigmergic environment presumes the definition of correlated information among the users and the tasks, reflecting the users' evaluation of the performance for a task (a node in the graph) relative to a problem type. The performance is stored for each problem type in a variable associated with each edge of the graph. The problem type is simply codified through a unique *ID* to distinguish among different performances when they are read (during the design phase of a plan) or modified (after the plan has been executed and evaluated). Evaluation of a plan entails (after its execution) a subjective assessment of the plan against some performance criteria.

The performance from all the graph's edges describes the state of the environment over time. Usually, the environment executes a set of processes on the variables (as aggregation end evaporation in the case of ants). For this case, we apply a simple weighted additive rule to simulate the aggregation of performances:

$$P_{jk}(task_k,t) = P_{jk}(task_k,t\text{-}1) + UP_{jk}(task_k, t)/w \quad (1)$$

where t represents the temporal component of the model which is incremented at each successive use; k is the task's identification index from the set of tasks used to construct the plan; $UP_{jk}(task_k, t)$ – is the user's performance of the k-th task evaluated from the side of task j at moment t; $P_{jk}(task_k, t)$ and $P_{jk}(task_k,t\text{-}1)$ are the new and previous values of the (collective) performance stored on the edge between the tasks j and k; and w is a tuning parameter to weight the impact of the last evaluation.

3.2 The User's Behaviour

The *agents* are the users who interact with the collaborative workspace when they are trying to find a plan for a certain problem. Conceptually, in any point in time an agent is "located" in a node (task) of the problem space, performing one of the following basic actions [9]: 1) evaluates the preference for the next possible task (or tasks) to be executed next, giving the current execution context; 2) selects the next best task (or a group of tasks) for further completing the structure of the plan; 3) executes the task (or the group of tasks) from the model, and finally; 4) evaluates the performance for the executed tasks. The evaluation activity is simulated in our agent-based simulation using the formula (1), while the first three actions with Luce's selection axiom [12]:

$$p_{jk} = e^{P_{jk}(task_k)/T} \Big/ \sum_{i=1}^{m} e^{P_{ji}(task_i)/T}, \quad (2)$$

where p_{jk} represents the preference for an alternative task, i.e. the selection probability of the task k from the task j; i is the task's index connected from the task j (in our model all the m tasks available in the problem space); $P_{jk}(task_k)$ is the (collective) performance stored on the edge between the tasks j and k (see formula 1); and T is a parameter used to define the deviation from a pure rational behaviour.

The above formula is the most common model of stochastic decisions due to its correlation with the psycho-social observations of human behavior in several domains [12]. As a result of normalization, the preferences for the unexploited tasks are diminishing after each performance update. This mechanism replicates the pheromone evaporation process of the real ants (e.g., even if a task has been positively evaluated after a plan execution, the associated preference will decrease once a better alternative is discovered and more frequently used). The uncertainty associated with the construction of preferences is modelled in formula (2) with the parameter T that range between 0 (when selection is deterministic as is the ideal case of a perfectly informed decision) and 1 (when the selection is completely random as in the case of a completely irrational decision). Note that Luce's selection axiom does not specify the reasons of uncertainty for modelling a plan; it may cover any aspect of complexity, unfeasibility, or users' refusal to evaluate the performance of a task.

3.3 Structuring the Knowledge from the Problem Space

The *agents,* reflects the users in the planning process and, consequently engage in means-ends reasoning activities to achieve the goal. During the execution of a plan they must adapt that plan to the changes and uncertainties associated with both: 1) the results of executing a task, and 2) the stability of the goal itself. Moreover, when the problem is not completely clear in advance or too complex to be fully tackled, the users are defining intermediate sub-goals that are going to be achieved in subsequent order. As a result, the design of the plan's structure in terms of composite tasks is done incrementally and interleaved with its execution.

Consequently, there are defined three basic **planning strategies** (PS), with direct implications on the way a plan is decomposed on different levels of abstractions [9]:

- *PS1* - when the user has a complete vision over the execution plan and therefore is able to provide a complete structure of the plan. It includes all the necessary actions together with their precedence constraints. This design strategy relates to the hypothetical conditions when the execution context remains stable in time as regards the tasks execution's outcome and the decision's objectives. Thus, each problem type is identically codified in all edges that connect the tasks used in structuring the plan and relates to a low level of semantically structured environment. In the context of collaborative planning, a group of users will evaluate in this case the complete solutions conceptualized individually by each member of the group to select, at the end, the most promising one.
- *PS2* - when the user is redefining the plan, after each execution of a composite task due to the uncertainty associated with its execution outcome. In this case the problem is considered to have stable objectives but uncertain outcome as regards the tasks execution. This planning strategy is codified in the conceptual graph of the problem space with different *IDs* for each sub-problem that corresponds to the variance from the current state of execution to the desired one in terms of task-types. In the context of collaborative planning, a group of users will evaluate, after each execution of a task, the solutions identified by each member of the group to select the most promising task to execute next.
- *PS3* - when the user is redefining the plan, after each execution of a composite task due to the uncertainty associated with its execution, in relation to the sub-goals in which the problem may be decomposed. In this case, when the problem is not entirely clear or too complex for defining an entire plan, the users are identifying intermediate sub-goals that are subsequently dealt with. This planning strategy is codified in the conceptual graph for the GDP modelling with different *IDs* for each sub-problem that corresponds to the variance from the current state of execution to any future possible one. In the context of collaborative planning, a group of users will evaluate (after each execution of a task) multiple solutions identified by each member of the group for every sub-goal in which the problem has been decomposed in order to select the most promising task to execute next.

4 The Simulation of Collaborative Planning

To evaluate the planning strategies for a problem we conducted a multi-agent simulation implemented in Netlogo [13] and following the classical methodology of virtual experimentation [14]. In the simulation the users ("turtles") are trying to identify the best plan for a problem by moving in the conceptual graph of tasks (the nodes and edges which are implemented as "turtles" as well). The number of tasks that compose the problem space is arbitrary chosen, while their utilities for that problem is assigned in the initialization phase with a random values between 0 and 1.

In the next sections are presented the normalized plans' performances for 100 successive explorations (iterations) of the planning strategies defined in the previous section (*PS1*, *PS2* and *PS3*). An exploration stands for a complete execution cycle of a plan. It includes three consecutive phases: 1) finding a suitable plan through the successive selection of the composite tasks for a given problem (using the formula 2); 2) executing the identified plan and assessing its performance by reading and averaging the predefined utility values of all the tasks that compose the plan; 3) evaluating the model by updating the performance value (using the formula 1). The statistics are aggregated from 30 experiments for a relatively simple problem type of 5 successive tasks. The parameter T from formula 2 is set to 0.7 to favour a faster convergence rate in finding a suitable solution in the problem space composed of 70 tasks from a predefined set of 5 task-types.

4.1 Indirect Collaborative Planning

From the engineering viewpoint, the planning strategies have direct implications on the way in which the collaborative workspace supports the plan structuration. As described before, the structure of the environment is simply reflected in the possibility to semantically decompose the problem in sub-problems on different levels of abstractions. This design issue implies to record with specific *IDs* the performance for each sub-problem type that results from the decomposition process.

In Fig. 2 are shown the aggregate performance (a relative number between 0 and 1 as resulted from averaging the predefined utility values of the tasks that compose the plan) from 30 experiments of a planning process in respect to the defined strategies *PS1*, *PS2* and *PS3*. As may be expected, the performance fits an exponential function, a typical behaviour for a stigmergic system [15, 16]. Like any heuristic model, the stigmergic coordination mechanisms do not guarantee finding an optimal solution, but a near-optimal or acceptable one. As a consequence, from one experiment to another there are some variations in performance for the convergence values.

The three strategies show different performance and convergences to an optimal solution. Contrasting with *PS1*, *PS2* takes benefit from the prior experiences not only in relation with the entire problem but also from the intermediate plans to solve the sub-problems in which the initial one has been decomposed. In addition, *PS3* increases the granularity by adding the opportunity to decompose the problem's goal.

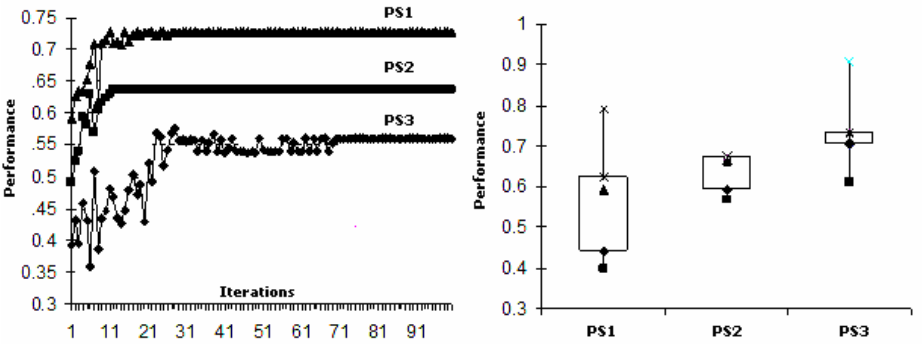


Fig. 2. The distribution of plan's performance for the three strategies [9]

Consequently, the figures show the influence of problem decomposition on the planning performance when the indirect communication between users is exhaustively used. As may be expected, *PS3* illustrate the best performance results and a low inconsistency among the identified solution. But, *PS2* converge faster to an optimal solution and a lower inconsistency among the feasible solutions. This can be explained by the additional constraints of having stable objectives.

4.2 Direct Collaborative Planning

In the previous section we investigated the impact of different strategies that requires different capabilities to structure the knowledge encoded in collaborative workspaces for the collective planning. As mention before we considered that the users are collaborating asynchronously through the stigmergic environment which support the most appropriate identification of tasks composition for a plan. In this section we introduce the possibility to synchronously collaborate inside the group of participants on different levels of abstraction regarding the plan structure as described in section 3.3. This experiment is based on the following premises: 1) all the participants that compose the group are actively engaged in the planning process; 2) there is no coordination mechanism regarding the intentions to find the composite tasks for a plan; 3) all the participants follow the same strategy to design a plan (i.e. *PS1*, *PS2*, *PS3*). Thus, *PS1* presumes the individual modeling of a complete plan after which is chosen the one with maximal performance. *PS2* is an intermediate case, when a group of users will evaluate, after each execution of a task, the solutions identified by each member of the group to select the most promising task to execute next. *PS3* extends *PS2* with the possibility to evaluate multiple solutions identified by each member of the group for the sub-goals in which the problem has been decomposed.

The **group size** (GS) is generally established by the nature of the problem and should have a minimal size to collaborate properly; if the group size is too low the group will be under an excessive pressure, if there is too large there is a waste of cognitive resources [17]. In Fig. 3 is represented the influence of GS over the planning performances. For the *PS1* the results show an optimal GS of approximately 7 participants, this number being reveled in various studies of group behaviors [17].

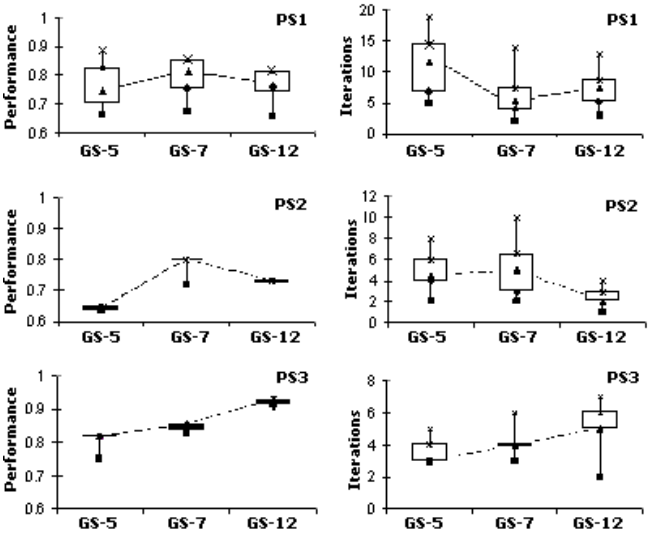


Fig. 3. The influence of GS over the plan's performances in respect to different planning strategies (PS1,PS2,PS3)

For the presented model, the diminishing performances for a GS over 7 participants is due to either a premature consolidation of the collective preferences (which drive the planning solution into local minima without the possibility to explore the problem space) or the lack of coordination opportunities during the planning process. The first case may be lessening by using high value for the *T* parameter (in formula 2), whilst the second by increasing the opportunities of coordination among participants. This aspect may be observed for the *PS2* which is more efficient for larger groups due to an increase of the exploration power in the problem space through the analysis of the intermediate results. For a GS of 7 participants *PS2* gives less variability among the solutions in respect to *PS1*, but despite its fast convergence to an optimal solution it is improper for small groups. Consequently, for collective planning, *PS3* provides the best performances despite the GS. Nevertheless for large groups, and in cases where a greater stability among solutions is needed, *PS2* is a better alternative, while *PS1* seems to be appropriate for small groups if the participants are actively involving in the planning process. In conclusion, as the degree of abstractization is higher, the collaboration opportunities become higher and the potential to directly support the collective intelligence may be amplified.

5 Conclusions

In this paper we investigated the impact of the engineered capabilities of structuring the knowledge encoded in collaborative workspaces over the collective performance in planning processes. The investigation is grounded on the theoretical framework of stigmergic systems and realized by building a multi-agent simulation to evaluate the performance of different planning strategies. The results show that different representational complexities of collaborative planning knowledge have a great

impact over the collective intelligence when the users are either interacting in a direct or indirect way through collaborative workspaces. They were applied in designing a collaborative workspace for planning the group decision process [18].

References

1. Grassé, P.P.: La theorie de la stigmergie: essai d'interpretation du comportement des termites constructeurs. J. Insectes Sociaux 6, 41–81 (1953)
2. Susi, T., Ziemke, T.: Social Cognition, Artefacts, and Stigmergy: A Comparative Analysis of Theoretical Frameworks for the Understanding of Artefact-mediated Collaborative Activity. J. Cognitive Systems Research 2, 273–290 (2001)
3. Parunak, V.D.: A Survey of Environments and Mechanisms for Human-Human Stigmergy. In: Weyns, D., Michel, F., Parunak, V.D. (eds.) E4MAS 2005. LNCS (LNAI), vol. 3830, pp. 163–186. Springer, Heidelberg (2006)
4. Elliott, M.A.: Stigmergic Collaboration: A Theoretical Framework for Mass Collaboration. PhD Thesis, Victorian College of the Arts, The University of Melbourne (2007)
5. Parunak, V.D., Brueckner, S.A.: The Cognitive Aptitude of Swarming Agents (2009), `https://activewiki.net/download/attachments/6258699/CASA.pdf`
6. Parunak, V.D., Belding, T.C., Bisson, R., Brueckner, S.A., Downs, E., Hilscher, R.: Stigmergic reasoning over hierarchical task networks. In: 8th International Conference on Autonomous Agents and Multi-Agent Systems, pp. 1195–1196. ACM Press, New York (2009)
7. Heylighen, F.: Collective Intelligence and its Implementation on the Web: algorithms to develop a collective mental map. J. Comput. and Math. Org. Theory 5, 253–280 (1999)
8. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive Stigmergy: A Framework Based on Agents and Artifacts. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2006. LNCS (LNAI), vol. 4389, pp. 124–140. Springer, Heidelberg (2007)
9. Zamfirescu, C.B., Filip, F.G.: Swarming models for facilitating collaborative decisions. J. of Computers, Comm. and Control V, 125–137 (2010)
10. Ghallab, M., Nau, D., Traverso, P.: Hierarchical Task Network Planning. Automated Planning: Theory and Practice. Morgan Kaufmann, San Francisco (2004)
11. Russell, S.J., Norvig, P.: AI: A Modern Approach, 2nd edn. Prentice-Hall, N. Jersey (2003)
12. Luce, D.: Individual Choice Behaviour. Wesley (1959)
13. Wilensky, U.: NetLogo (1999), `http://ccl.northwestern.edu/netlogo/`
14. Carley, K.M.: On Generating Hypotheses Using Computer Simulations. J. Systems Engineering 2(2), 69–77 (1999)
15. Guerin, S., Kunkle, D.: Emergence of constraint in self-organizing systems. J. Nonlinear Dynamics, Psychology, and Life Sciences 8(2), 131–146 (2004)
16. Parunak, V.D., Brueckner, S.A., Sauter, J.A., Matthews, R.: Global convergence of local agent Behaviors. In: 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 305–312. ACM Press, New York (2005)
17. Hwang, H., Guynes, J.: The effect of group size on group performance in computer supported decision making: J. Information and Management 26, 189–198 (1994)
18. Zamfirescu, C.B., Duta, L., Candea, C.: Implementing the "Design for Emergence" Principle in GDSS. In: Respício, A., et al. (eds.) Bridging the Socio-technical Gap in Decision Support Systems - Challenges for the Next Decade, pp. 61–72. IOS Press, Amsterdam (2010)

Signature Verification Based on a Global Classifier That Uses Universal Forgery Features

Joanna Putz-Leszczynska[1] and Andrzej Pacut[2]

[1] Research and Academic Computer Network NASK,
Wawozowa 18, 02796 Warsaw, Poland
[2] Warsaw University of Technology,
Nowowiejska 15/19, 00663 Warsaw, Poland
asia.putzleszczynska@gmail.com

Abstract. Handwritten signature verification algorithms are designed to distinguish between genuine signatures and forgeries. One of the central issues with such algorithms is the unavailability of skilled forgeries during the template creation. As a solution, we propose the idea of *universal forgery features*, where a global classifier is used to classify a signature as a genuine one or, as a forgery, without the actual knowledge of the signature template and its owner. This classifier is trained once, during the system tuning on a group of historical data. A global classifier trained on a set of training signatures is not be additionally trained after implementation; in other words, additional users enrollments have no effect on the global classifier parameters. This idea effectively solves the issue of the lack of skilled forgeries during template creation.

Keywords: signature verification, global classification.

1 Introduction

Handwritten signature has been used to confirm the authenticity of actions and documents since the written language become available. The general acceptance of a signature as person's identification mark makes the signature a viable option to be considered as a biometric trait. Despite an abundance of computers systems, handwritten signature is still widely used in everyday life. It is still essential for most of the financial transactions, notarial deeds, etc. Forging someone's signature is regarded as a serious crime. Forensic handwriting experts, whose discipline is forensic document examination, can identify a forged signature with almost 100% accuracy [6], but at the same time they identify almost 7% of genuine signatures as forgeries. Moreover, they need much more than a few seconds to perform their procedures. Additionally, the people who verify the signatures for most of the financial transactions are not specialists. Such a person might have attended a special training, but it is usually not enough to reliably recognise a genuine signature from a forgery. That is why there is a great demand for *automatic signature verification systems*.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 170–179, 2011.

2 Idea

Signature verification methods are designed to discriminate between genuine signatures and forgeries.

A signature is a multidimensional time series in (finite) time T. The time series dimension is determined by the number of signature characteristics which the device is able to capture, e.g. position coordinates $x \in X, y \in Y$, pen pressure $p \in P$ and angles: azimuth $z \in Z$ and altitude $l \in L$ (Fig. 1). An on-line signature, denoted by g, can thus be represented as a parametric curve in $X \times Y \times P \times Z \times L$, i.e.:

$$g : T \rightarrow X \times Y \times P \times Z \times L \tag{1}$$

where $g = [g^{\mathrm{x}}\, g^{\mathrm{y}}\, g^{\mathrm{p}}\, g^{\mathrm{z}}\, g^{\mathrm{l}}]^{\mathrm{T}}$

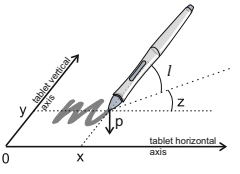


Fig. 1. On–line signature acquisition using Wacom Intuos

If we use only a sub–vector of functions, with coordinates in a list *list*, we denote it by g^{list}. For example, if $list = \mathrm{x,y}$, we have $g^{\mathrm{x,y}} = [g^{\mathrm{x}}\, g^{\mathrm{y}}]^{\mathrm{T}}$.

A projection of this curve onto the X, Y plane is simply an image of the signature "on paper", called the ***off–line signature***:

$$|g_{\mathrm{off}}| = \left\{(x, y) \in X \times Y : \exists_{t \in T}\ [x, y]^{\mathrm{T}} = g^{\mathrm{x,y}}(t)\right\} \tag{2}$$

In contrast to other biometric characteristics, signature verification has to deal not only with signatures belonging to other people but also with *skilled forgeries*. The latter are obviously not available during template creation. This fact limits the use of non–linear global classifiers like neural networks or support vector machines; to effectively train such a classifier, one has to provide not only the genuine signatures, but also skilled forgeries. Typically, randomly selected signatures of other people (*random forgeries*) are used instead, which results in a sub–optimal classifier [5].

To effectively solve the issue of the lack of skilled forgeries during template creation we propose to use historical data, which has been collected in the past and is available at the time of system creation. We use data that contains not only genuine signatures, but also the skilled forgeries for a group of users. Thus an important question arises, namely if this historical data can be exploited in the verification process. A positive answer would suggest that there exist some user independent features — the *forgery features* -that differentiate between the genuine signatures and the forgeries, independently from user. The universal forgery feature idea can thus be formulated as:

There exist features, independent from the person signing, that distinguish between genuine signatures and skilled forgeries.

If such features exist, they can be used in the verification process. One then would be able to construct a global classifier, trained on historical data including skilled forgeries.

Such an approach splits the verification process into two distinct stages (two-stage classifier). In the first stage, a number of *individual component classifiers* would provide the *individual component classification scores*. In the second stage, the individual classification scores are fed to the *global classifier*, which provides the verification result, classifying a signature as a forgery or not a forgery. Each new user would be first required to provide a set of enrollment signatures, which would be used for estimating his own template for individual component classifiers. It means that only the first phase would be carried out for a new subject. No skilled forgeries would be needed in this phase, because no additional training would be conducted for a global classifier in the second stage.

2.1 Individual Component Classifiers

Here, we will consider signatures for either on–line or off–line cases. The signature instance g is thus either the data sequences in the on–line case or the images in the off–line case. For each user, we define K individual component classifiers. The k-th classifier, $k = 1, .., K$, differs between the users u by the *individual component template* $\psi_{k,u}$. Here the "individual" refers to individual estimation of the template for each user, and "component" refers to the fact that each of these component classifiers is a classifier by itself, and can be used as an independent classifier for signature verification.

Let us denote by $s_k(g; \psi_{k,u})$ the individual component classification score for input signature g for k–th component classifier and u-th user. We propose two biometric estimation procedures, which differ by the signatures used for template estimation:

- **Procedure 1.** The individual component classifier for the u–th user uses both the enrollment signatures $\mathbf{O}_{E,u}$ and the random forgeries $\mathbf{O}^{\mathbf{c}}_{E,u}$.

$$\{\mathbf{O}_{E,u}, \mathbf{O}^{\mathbf{c}}_{E,u}\} \rightarrow \psi_{k,u} \tag{3}$$

  Here, the most common solutions are based on neural networks or support vector machines.
- **Procedure 2.** The individual component classifier for the u–th user uses only the enrollment signatures $\mathbf{O}_{E,u}$.

$$\mathbf{O}_{E,u} \rightarrow \psi_{k,u} \tag{4}$$

  For this procedure, common solutions are based on misalignment measurement e.g. DTW or Mahalanobis distance.

When the individual component template $\psi_{k,u}$ is known, the individual component classifier can calculate the individual component classification score for each new verified signature. The individual component classification score is stripped of any specific qualities of the person creating the template.

In Sections 3.1 and 3.2, a number of examples of various individual component classifiers are presented for both on–line and off–line verification cases.

Global Classifier. The *global classifier* idea works on the assumption that there exist forgery features that differentiate between the genuine signatures and the forgeries, independently from the user. This means that we create a single global classifier that should work correctly for all users. The global classifier is trained with the use of historical data belonging to different users. This data contains both the genuine signatures $\mathbf{O}_E$ and skilled forgeries $\mathbf{O}_F$.

The global classifier takes as its inputs the individual component classification scores $\boldsymbol{s} = [s_1(g;\psi_{1,u}), s_2(g;\psi_{2,u}), ..., s_K(g;\psi_{K,u})]$. Its output, namely, the global classification score $\mathcal{S}(\boldsymbol{s};\phi)$ g is calculated as a function of the inputs and a certain global parameter vector ϕ.

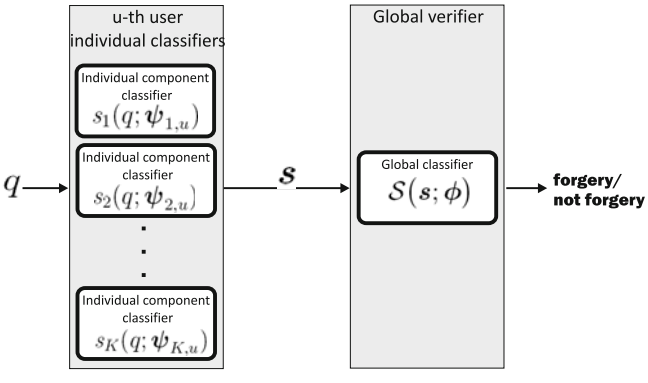


Fig. 2. Computation of the global classification score for the verified signature q using a two stage classifier

The data set used for the estimation of the global classifier parameter vector is different from the one used to estimate the parameter vectors of the individual component classifier. During the training, the optimal parameter vector ϕ of the global classifier is estimated using a supervised iterative training, with historical data as the reference outputs. The optimal global parameter vector ϕ is estimated independently from the individual component templates. When the optimal global parameter vector ϕ is known, the global classifier can calculate the global classification score for each new verified signature q (Fig.2). The global classifier can be constructed as a neural network, a support vector machine (SVM) or other non–linear classifier.

3 Testing the Idea with the Use of Correlation Coefficients

Our aim was to check the relations between the individual component classification scores, computed for genuine signatures and skilled forgeries. Tests were conducted on the MCYT on-line and off-line databases:

- **MCYT-Signature-100** (100 users: 25 genuine signatures, 25 skilled forgeries for each) [7]
- **MCYT-SignatureOff-75** (75 users: 15 genuine signatures, 15 skilled forgeries for each) [7]

Hence, we investigate a relationship between the individual component classification scores for these two groups. The simplest approach is to use Pearson's correlation coefficients. We expect that the statistical properties of these correlation coefficients are different for these two groups. In particular, we show that these correlation coefficients are statistically significantly different. Specifically, the correlation coefficients in the genuine group may be equal to zero. To compare between the coefficients computed for genuine signatures and skilled forgeries, we use two statistical approaches based on testing:

a) The hypothesis of no correlation $H_0 : \rho = 0$
b) The equality of correlation coefficients $H_0 : \rho_1 = \rho_2$

To perform the aforementioned testing, and thus to check the viability of the universal forgery features, we carried out two sets of experiments. The first one was conducted with the use of independent individual component classifiers created for off–line signature verification (Sec. 3.1).

In the second set of experiments, the global system employs individual component classifiers for on–line verification. These classifiers use the DTW(Dynamic Time Warping) to calculate the individual component classification scores for different non-parametric features, however they use the same warping path in this calculation, which effects in a dependency of the component classifiers. The main goal of the tests was to check how this dependency of component classifiers affects the correlation coefficients (Sec. 3.2).

3.1 Off-Line Verification System

There exist many off–line signature verification methods, each based on different signature features. We selected seven methods to be individual component classifiers, generating forgery features.

Individual Component Classifiers. As the individual component classifiers we selected the ones based on global image analysis with Mahalanobis distance [4], neural networks based on three types of feature global, grid and texture [1]. Two next component classifiers use Dynamic Time Warping, applied to a time series derived from the signature image. In the approach of [3], a two–dimensional signature pattern is converted into a one–dimensional waveform by taking its vertical projection calculated by counting the pixels in the image columns. In the DTW of [2], the image is binarized and dilated with a morphological mask, and

closed contours are traced. The spatial coordinates, together with the tangential angle are recorded for every point of each contour. The selected approaches differ in the way of signature transformation and feature choice. We selected them due to their relatively good performance during verification test on skilled forgeries. All of them give one classification score.

Comparison of Individual Component Classifiers. We use the classification scores produced by the 6 individual component classifiers as the inputs to a neural network. To investigate the relations between the components, sample correlation coefficients were calculated between the classification scores obtained for each component classifier, separately for genuine signatures g and for skilled forgeries. This gave us $(6 \cdot 5)/2 = 15$ correlation coefficients for each case. The 95% significance limits were calculated for these coefficients, and the correlation coefficients were tested *separately* against the hypotheses of null correlation. The results are marked symbolically in Fig. 3, where black squares denote the significant correlation cases. For genuine signatures, only (27%) individual component classification scores are significantly correlated. For skilled forgeries, almost all 87% of the individual component classification scores are significantly correlated.

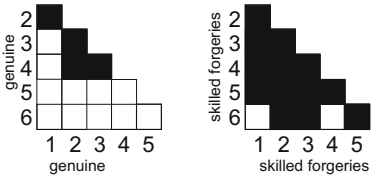


Fig. 3. Significance of results correlation; black squares: significant correlation, white squares: insignificant correlation

The used individual component classifiers used in this experiment were created independently. Since almost all individual component scores are uncorrelated for genuine signatures, each of them carries different ŞnoveltyŤ, hence merging the methods may give some additional gain. The situation looks opposite for skilled forgeries where almost all individual component scores are correlated. This fact suggests that it is exactly the forged data that causes the classification scores to be correlated. Thus, if all classifiers are treated as feature extractors, even linear combination should work in the global classifier, but nonlinear combination should improve the overall result even more.

3.2 On-line Verification System

Individual Component Classifiers. We selected DTW based system modules of *our basic DTW* verification system described in [8], to be the individual component classifiers. The first three components sc_x, sc_y, sc_p use DTW to compute misalignment between the template signature and the verified signature - each for a different non-parametric feature: Δx–coordinates, Δy–coordinates

and the pressure p. The warping path is obtained with the spatial dimensions and which is also used for computing all three misalignments. Hence, these three individual component classification scores are dependent. The fourth classifier sc_γ calculates the relative length difference γ between the verified signature and the template signature.

Comparison of Individual Component Classifiers. First, sample correlation coefficients were calculated between the four individual component classification scores obtained for each individual component classifier, separately for genuine signatures and for skilled forgeries. They were tested against the zero hypothesis of no correlation. This gave the matrix of $(4 \cdot 3)/2 = 6$ correlation coefficients for each case. The significant and– non–significant results are marked symbolically in Fig. 4, where black squares denote the significant correlation cases. Here, unlike for independent classifiers, almost all individual component classification scores are correlated – both for genuine signatures, and for skilled forgeries, which is the result of the individual component classifiers dependency.

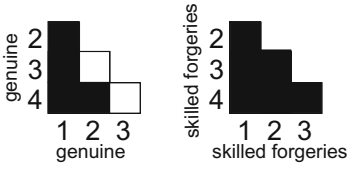


Fig. 4. Significance of results correlation; black squares: significant correlation, white squares: insignificant correlation

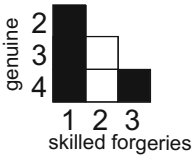


Fig. 5. Equality of correlation coefficients; black squares: coefficients are not equal, white squares: coefficients are equal

We still can find out whether there is a difference between the sample correlations calculated for genuine signatures and skilled forgeries. In this order, a second test was performed regarding the equality of sample correlation coefficients obtained for the same two component classifiers, one for genuine and the second for forgeries. The equality and non–equality results are marked symbolically in Fig. 5, where black squares denote the unequal sample correlation cases at 95% significance level. The null hypothesis regarding the equality of correlation coefficients was rejected for 66% correlation coefficients comparisons. This suggests that the forgery features may exist also in the case of dependent individual component classifiers. Consequently, a classifier may distinguish a new genuine signature from a forgery when trained on a different data not related to this signature.

4 Testing the Two–Stage Classifier

The goal of this testing was to compare the verification errors obtained individually for all individual component classifiers with the ones obtained for the two–stage global classifier, both for off-line and on-line classifiers.

We individually divided each of the signature databases into two parts. The Estimation Data used for the training/estimation phase included data for 40% of persons, and the remaining data for 60% of persons were used as the Testing Data. Each classifier were trained using the estimation data, and the threshold was set to the value that equalizes the False Acceptance Rate (FAR) and False Rejection Rate (FRR) namely for (Equal Error Rate). To check the variability and the generalization ability of the classifiers, the same threshold was used for the Testing Data. This threshold certainly did not produce the equal error rate, but for each classifier the resulting FAR and FRR were close to the EER obtained for the Estimation Data (Tab. 1 & 2). Additionally, 1000 different divisions of the database into the estimation and testing sets was used. These 1000 randomly chosen divisions were the same with every experiment.

4.1 Two Stage Classifier for Off–Line Verification

All the experiments were conducted according to the testing procedure using Estimation and Testing data. For each user u, 10 enrollment signatures $\mathbf{O}_{E,u}$ were used to compute the individual component classifier parameters; for neural networks, 10 random forgeries $\mathbf{O}^{\mathbf{c}}_{E,u}$ were additionally used.

We implemented all the individual component classifiers presented in Section 3.1. Three neural network sub–methods were treated as independent classifiers (Baltzakis, Papamarkos [1](1)–(3)).

The EER evaluated individually for the individual component classifiers varies here from 24% to 34%. In the testing phase, we obtained FAR and FRR valued of 20.9% up to 31.7% after testing on a group of skilled forgeries and genuine signatures belonging to different sets of people, respectively.

The component classifiers form a basis for the two–stage global approach capitalising on the forgery features. Each individual component classifier is created separately and then, its parameters are fixed. The classification scores are used as the inputs for a nonlinear (neural network) global classifier. This classifier is trained with the classification scores obtained for genuine signatures and for skilled forgeries – the set used for the EER estimation. This global classifier is trained once, during the estimation phase. We must stress that during the testing phase there was no additional training of the global classifier. Only the component classifiers parameters were calculated for new users, with the use of their genuine signatures and random forgeries.

The experiments results are presented in Table 1. The errors evaluated for the two stage classifier are significantly better than the ones obtained individually for the individual component classifiers. The global approach actually halved these error rates; the EER for the global classifier was 10.42%, and during the testing phase the FAR= 15.67% and FRR=17.98%. Thus, the two-stage classifier with the global classifier in the second stage gives much better results than the individual component classifiers working individually.

Table 1. Performance of component classifiers and two-stage global classifiers

Verification algorithm - our implementation	Estimation	Testing	
	EER [%]	FRR [%]	FAR [%]
(1) Fierrez-Aguilar et al. [4]	24.40 ± 1.92	24.66 ± 4.95	24.92 ± 4.49
(2) Baltzakis, Papamarkos [1](1)	33.11 ± 2.21	33.74 ± 6.09	33.38 ± 6.73
(3) Baltzakis, Papamarkos [1](2)	30.46 ± 2.31	30.60 ± 5.16	30.63 ± 4.16
(4) Baltzakis, Papamarkos [1](3)	34.28 ± 2.17	34.54 ± 4.53	34.23 ± 4.10
(5) Fang, Leung et al. [3]	23.40 ± 2.00	23.65 ± 5.51	23.60 ± 4.54
(6) Deng et al. [2]	29.85 ± 2.50	29.84 ± 4.52	30.33 ± 5.47
Two stage classifier for off–line	**10.42 ± 1.01**	**17.98 ± 4.2**	**15.67 ± 3.58**

4.2 Two Stage Classifier for On–line Verification

Now we compare the verification errors obtained for on–line individual component classifiers sc_x, sc_y, sc_p, sc_γ with the ones obtained for the two stage global classifier (Tab. 2). All the experiments were conducted according to the testing procedure using Estimation and Testing data. For each user u, 10 enrollment signatures $\mathbf{O}_{E,u}$ were used to compute the individual component classifier parameters.

The errors, evaluated individually for the individual component classifiers, vary from 4% to 16%.

The individual component classifiers (sc_x, sc_y, sc_p, sc_γ) form a basis for the two–stage global approach capitalizing on the forgery features. Each of the component classifiers is first trained separately. Then, the classification scores are used as the inputs of a nonlinear (neural network) global classifier. This classifier is trained during the estimation phase with the classification scores obtained for genuine signatures and for skilled forgeries. Again (like in the off–line case) the global classifier was not re-trained during testing. Only the component classifiers parameters were computed for new users with the use of genuine signatures.

The experiments show that it is possible to construct a classifier that tells a genuine signature from a forgery without the actual knowledge of the template the signature belongs to. Again, it confirms that it is possible to obtain a classifier that is able to distinguish between genuine signature and forgery, by training it only with a given set of signature samples even when the component classifiers

Table 2. Performance of component classifiers and two-stage global classifier

Verification algorithm	Estimation	Testing	
	EER[%]	FRR[%]	FAR[%]
Component classifier for feature x	4.16 ± 0.66	4.21 ± 1.22	4.52 ± 1.47
Component classifier for feature y	13.41 ± 0.91	13.50 ± 1.33	13.93 ± 1.58
Component classifier for feature p	15.79 ± 1.4	15.78 ± 2.83	16.15 ± 2.66
Component classifier for feature γ	16.05 ± 1.4	16.17 ± 2.59	16.05 ± 1.73
Two stage classifier for on–line	1.99 ± 0.39	2.71 ± 0.84	2.94 ± 1.33

are dependent. The results obtained for global training are significantly better than the ones calculated individually for the individual component classifiers.

5 Conclusions

The proposed two–stage signature verification procedure with global training in the second stage shows that various verification algorithms can be joined together by a global classifier that classifies into genuine–false classes to gain a synergic effect. The algorithm was used for new (testing) data without further learning or other modifications. To work properly, the algorithm must thus have acquired some knowledge of a common property of Ťbeing falsifiedŤ, independent from any particular signature. Both for off-line and on-line verification, our approach shows the importance of a global training using skilled forgeries, and paves a way for further improvement of error rates. The experiments show that it is possible to construct a classifier that tells a genuine signature from a forgery without the actual knowledge of the template the signature belongs to. It is important to underline that even when the individual component classifiers are dependent, the universal forgery idea works, which suggests that the classifiers do not have to be selected in a proper way. This shows the need of training the classifier with skilled forgeries. Without this possibility, the classifier trained with random forgeries is not able to successfully detect skilled forgeries.

References

1. Baltzakis, H., Papamarkos, N.: A new signature verifcation technique based on a two-stage neural network classifier. Engineering Applications of Artifcial Intelligence 14, 95–103 (2001)
2. Deng, P.S., Liao, H.-Y., Ho, C.W., Tyan, H.-R.: Wavelet-based off-line signature verification. Computer Vision and Image Understanding 76(3), 173–190 (1999)
3. Fang, B., Leung, C.H., Tang, Y.Y., Tseb, K.W., Kwokd, P.C.K., Wonge, Y.K.: Off-line signature verifcation by the tracking of feature and stroke positions. Pattern Recognition 36, 91–101 (2003)
4. Fierrez-Aguilar, J., Alonso-Hermira, N., Moreno-Marquez, G., Ortega-Garcia, J.: An off-line signature verification system based on fusion of local and global information. In: Maltoni, D., Jain, A.K. (eds.) BioAW 2004. LNCS, vol. 3087, pp. 295–306. Springer, Heidelberg (2004)
5. Impedovo, D., Pirlo, G.: Automatic signature verification: The state of the art. IEEE Transactions On Systems, Man, And Cybernetics. Part C, Applications And Reviews 38(5), 609–635 (2008)
6. Kam, M., Gummadidala, K., Fielding, G., Conn, R.: Signature authentication by forensic document examiners. Journal of Forensic Sciences 46(4), 884–888 (2001)
7. Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Hernaez, I., Satue, A., Igarza, J., Vivaracho, C., Escudero, D., Moro, Q.: MCYT baseline corpus: a bimodal biometric database. IEE Proc.-Vis. Image Signal Process 150(6), 395–401 (2003)
8. Putz-Leszczynska, J.: On-line signature verification using dynamic time warping with positional coordinates. In: Proceedings of SPIE, the International Society for Optical Engineering, vol. 6347 (2006)

Functional and Dependability Approach to Transport Services Using Modelling Language

Katarzyna Michalska and Jacek Mazurkiewicz

Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology,
Janiszewskiego 11-17, 50-372 Wroclaw, Poland
{katarzyna.michalska,jacek.mazurkiewicz}@pwr.wroc.pl
http://www.pwr.wroc.pl

Abstract. The paper describes an approach to functional dependability modelling of discrete transport systems. The proposed method is based on specified description languages that can be seen as a bridge between system description (as a mathematical or expert specification) and an analysis tools. In this paper system analysis is based on computer simulation using Monte Carlo simulation. System is analysed from the functional point of view (focusing on business service realized by a system) with support of specified dependability measures. The paper presents some exemplar system modelling based on a case study. Proposed approach can be used as a tool helpful to administrate the system.

Keywords: functional dependability, modelling language, discrete transport systems.

1 Introduction

The transportation systems are characterized by a very complex structure. This complexity affects a need to use mechanisms and means helpful to administrate the system (i.e. make a decisions related to the performance). The building or the optimization of the transportation systems can be expensive and problematic. The necessary analysis mechanisms should be created not only for the money saving, by also as a tool for the future administration of the system and decision support (based on some specified metrics). The main problem is to realise multicriteria optimization for transport system management. The solution ought to combine the sets of reliability, functional and economic parameters. The mentioned data are modelled by distributions - so it makes the optimization problem more sophisticated. The classic solution way is based on human-expert experience to play the dispatcher or the manager role. But nowadays the elements of the transportation systems are characterized by more and more screwed-up parameters and the historical experience is not enough to create real and actual solution for transportation systems driving. This is the reason why we propose the computational collective intelligence to create the device to support human's decisions. Among many advantages, the analysis device has some disadvantages.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 180–190, 2011.

One of them is that every simulator (used as an analysis tool) has its own input format - very specific and sometimes sophisticated. Moreover it requires description as close to reality as possible. The required generality of the input model can be reached by using proper modelling languages that allow us to combine the mathematical description of the system and usage of it as an analysis tool. However it allows us also to be unchained to a specified tool, since the conversion between languages can be realized easily. This way it provides universality of the language and guarantees the independence from software solutions. Unfortunately, for transportation systems that are considered in the paper, the description languages are not so common (as for the computer network systems). In section 4 we propose a language that can be used for describing Discrete Transport System in case of further analysis. The transportation system is analysed from the functional point of view, focusing on business service realized by a system [11]. It allows to calculate different system measures which could be a base for decisions related to administration of the transportation systems. The metric are calculated using Monte Carlo techniques [3]. No restriction on the system structure and on a kind of distribution is the main advantage of the method. Such approach allows to forget about the classical reliability analysis based on Markov or Semi-Markov processes [13] - idealised and hard for reconciliation with practice. The approach is based on the transportation system of the Polish Post regional centre of mail distribution. The origin discrete transport system model is presented in section 2. The main service given by the post system is the delivery of mails. From the client point of view the quality of the system could be measured by the time of transporting the mail from the source to the destination. We propose the formal model of the system (sections: 3.1, 3.2, 3.3, 3.4) and the heuristic approach to management process (section 3.5). The quality of the analyzed system is measured by the availability defined as an ability to realize the transportation task at a required time (described in section 6). Next (section 7), we give an example of using presented model and proposed simulation methodology (described in section 5) for the analysis of the Dolny Slask Polish Post regional transportation system.

2 Discrete Transport System Based on Polish Post

The analysed transportation system is a simplified case of the Polish Post. The business service provided by the Polish Post is the delivery of mails. The system consists of a set of nodes placed in different geographical locations. Two kinds of nodes could be distinguished: central nodes *(CN)* and ordinary nodes *(ON)*. There are bidirectional routes between them. Mails are distributed among ordinary nodes by trucks, whereas between central nodes by trucks, by trains or by planes. The mail distribution could be understood by tracing the delivery of some mail from point A to point B. At first the mail is transported to the nearest ordinary node. Different mails are collected in ordinary nodes, packed in larger units called containers and then transported by vehicles scheduled according to a time-table to the nearest central node. In central node containers are repacked

and delivered to appropriate (according to delivery address of each mail) central node. In the Polish Post there are 14 central nodes and more than 300 ordinary nodes. There are more than one million mails going through one central node within 24 hours. It gives a very large system to be modelled and simulated. Therefore, we have decided to model only a part of the Polish Post discrete transport system - one central node with a set of ordinary nodes. The income of mail containers to the system is modelled by a stochastic process. Each container has a source and destination address. The central node is the destination address for all containers generated in the ordinary nodes. In case of central node, there are separate random processes for each ordinary node. Each vehicle has a given capacity - maximum number of containers it can haul. Central node is a base place for all vehicles. The vehicle management system takes into account a set of routes (sequence of nodes starting and ending in the central node, time-moments of leaving each node in the route and the recommended size of a vehicle). The process of vehicle operation could be stopped at any moment due to a failure (random process). After the failure, the vehicle waits for a maintenance crew (if there are no available), is being repaired (random time) and after it continues its journey. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity.

3 Formal Model of Discrete Transport System

Depending on the level of detail in modelling the granularity of traffic flow, traffic models are broadly divided into two categories: macroscopic and microscopic models. According to Gartner et al. [4], a macroscopic model describes the traffic flow as a fluid process with aggregate variables, such as flow and density. The state of the system is then simulated using analytical relationships between average variables such as traffic density, traffic volume, and average speed. On the other hand, a microscopic model reproduces interaction of punctual elements (vehicles, road segments, intersections, etc) in the traffic network. Each vehicle in the system is emulated according to its individual characteristics (length, speed, acceleration, etc.). Traffic is then simulated, using processing logic and models describing vehicle driving behaviour, such as car-following and lane-changing models. Those models reproduce driver-driver and driver-road interactions. Despite its great accuracy level, for many years this highly detailed modelling was considered a computationally intensive approach. Since the last twenty years, with the improvements in processing speed, this microscopic approach becomes more attractive.

3.1 Model Overview

We can model discrete transport system as a quadruple [13]:

$$DTS = \langle Client, BS, TI, MS \rangle , \tag{1}$$

where: *Client* - client model, *BS* - business service, a finite set of service components, *TI* - technical infrastructure, *MS* - management system.

3.2 Technical Infrastructure

The technical infrastructure of *DTS* could be described by three elements [15]:

$$TI = \langle No, V, MM \rangle , \tag{2}$$

where: *No* - set of nodes, *V* - set of vehicles, *MM* - maintenance model. Set of nodes *(No)* consists of single central node *(CN)* and a given number of ordinary nodes (ON_i). The distance between each two nodes is defined by the function:

$$distance : No \times No \rightarrow R_+. \tag{3}$$

Each node has one functional parameter the mean (modelled by normal distribution) time of loading a vehicle:

$$loading : No \rightarrow R_+. \tag{4}$$

Moreover, the central node *(CN)* has additional functional parameter: number of service points (in each ordinary node there is only one service point):

$$servicepoints : CN \rightarrow N_+. \tag{5}$$

Each vehicle is described by following functional and reliability parameters [10]: mean speed of a journey - $meanspeed : V \rightarrow R_+$, capacity - $capacity : V \rightarrow N_+$, mean time to failure - $MTTF : V \rightarrow R_+$, mean repair time - $MRT : V \rightarrow R_+$. The traffic is modelled by a random value of vehicle speed and therefore the time of vehicle *(v)* going from one node (n_1) to the other (n_2) is given by a formula:

$$time(v, n_1, n_2) = \frac{distance(n_1, n_2)}{Normal(meanspeed(v), 0.1 \cdot meanspeed(v))} \tag{6}$$

where: *Normal* denotes a random value with the Gaussian distribution. Maintenance model *(MM)* consists of a set of maintenance crews which are identical and unrecognized. The crews are not combined to any node, are not combined to any route, they operate in the whole system and are described only by the number of them. The time when a vehicle is repaired is equal to the time of waiting for a free maintains crew (if all crews involved into maintenance procedures) and the time of a vehicle repair which is a random value with the Gaussian distribution: $Normal(MRT(v), 0.1 \cdot MRT(v))$.

3.3 Business Service

The business service can be seen as a set of service components and tasks that are used to provide service in accordance with business logic for this process. Therefore, *(BS)* is modelled a set of business service components *(sc)* [12]:

$$BS = \{sc_1, ..., sc_n\} , \ n = length(BS) > 0, \tag{7}$$

the function *length(X)* denotes the size of any set or any sequence *X*.
Each service component in DTS consists of a task of delivering a container from a source node to the destination one.

3.4 Client Model

The client model consists of a set of clients *(C)*. Each client is allocated in some node of the transport system:

$$allocation : C \rightarrow No. \tag{8}$$

A client allocated in an ordinary node generates containers (since, we have decided to monitor containers not separate mails during simulation) according to the Poisson process with destination address set to ordinary nodes. In the central node, there is a set of clients, one for each ordinary node. Each client generates containers by a separate Poisson process and is described by intensity of container generation [11]:

$$intensity : C \rightarrow R_{+}. \tag{9}$$

3.5 Heuristic Approach to Management System

The main goal of management model is to control the movement of trucks. We proposed a promising and effective heuristic management approach [10] which allows to react for the critical situations which can occur during the normal system work [15]. The decisions (send a truck to a given destination node) are taken in moments when a container arrives to the central node. The truck is send to a trip if: the number of containers waiting in for delivery in the central node of the same destination address as that just arrived is larger than a given number, there is at least one available vehicle, the simulated time is between 6 am and 22 pm minus the average time of going to and returning from the destination node. The truck is send to a node defined by destination address of just arrived container. If there is more than one vehicle available in the central node, the vehicle with size that a fits the best to the number of available containers is selected, i.e. the largest vehicle that could be fully loaded. If there are several trucks with the same capacity available the selection is done randomly. On the other hand we observe in the same way the vehicles available in the ordinary nodes. The only difference is the greater level of threshold to initialize the vehicle journey.

4 Modelling Languages

Since, the purpose of the work is to analyse transportation system based on specified mathematical model, there is a need to transfer the data into a format that would be useful in an analysis tool. It requires specify data format that can be easily shared between various tools or even several of transport architectures (independent form complexity). Several data sharing and exchange standards have been developed in the Intelligent Transport Systems [8]. They define a standard data format for the sharing and exchange of the transportation data mostly based on UML (Unified Modelling Language) diagrams. Other solutions, i.e. Japanese standard called UTMS (Universal Traffic Management

Systems) focuses rather on the road traffic system. Still none of them is coherent with solutions proposed in this paper, since they describe different types of transport system. Moreover they are based on UML diagrams, which are the graphical representation of a model, but not the one, that can be simply used as an input format for any available analysis tool (computer simulator). Additionally description language for this system should be as close as real, not only to a mathematical description of the system, but to real system behaviour and its parameters. For this reasons, we propose an authors solution of a description language for a proposed model, called DTSML (Discrete Transport System Modelling Language). As a format of the proposed language XML (Extensible Markup Language) was chosen. Main reason is a simple (easy to learn) and readable structure, that can be easily convert to text or other format. Moreover, XML is supported not only with various tools (providing validation possibilities) but is also supported by many programming languages and framework in case of quicker and more efficient implementation. Fig. 1 shows a part of the XML Schema for Discrete Transport System Modelling Language. As can be seen, each element of the system described in section 3 is modelled as a complex element with appropriate sub-elements and attributes. The proposed language assures aggregation of dependability and functionality aspects of the examined systems. One language describes whole system and provides a universal solution for various techniques of computer analysis as an effective and suitable input for those tools. Expect easiness of potential softcomputing analysis, promising scalability and portability (between analysis tools) can be named as the main advantage of the language usage. Moreover the language is easy to read and to process using popular and open-source tools; however the metadata used in this format are still a significant problem in case of file processing (large size of the file).

5 Simulation Methodology

A simulator, performing a Monte-Carlo simulation [3], is composed of four basic elements: input data file, system description language analyzer, event-driven simulator, output data file. The system description language analyzer creates, based on data taken from the input file objects which represent system in the memory. The event-driven simulator repeats N-times the following loop [14]:

- initial state of a DTS - event initial state, set $t = 0$ - repeat until $t < T$:
- take first event from event list - set time equals time of event - realize the event.

The event is a basis for a simulation process. It is described by the following data: time of event occurring, type of event - vehicle failure for example, part of the *DTS* where event has its influence. The events are placed in the ordered list. Time of event occurring is the key for the list order. We have following events in the *DTS*: vehicle reached the node, vehicle is failing, vehicle is repaired, task is starting, end of simulation [14]. For the purpose of simulating *DTS* we have used Parallel Real-time Immersive Modelling Environment *(PRIME)* [7]

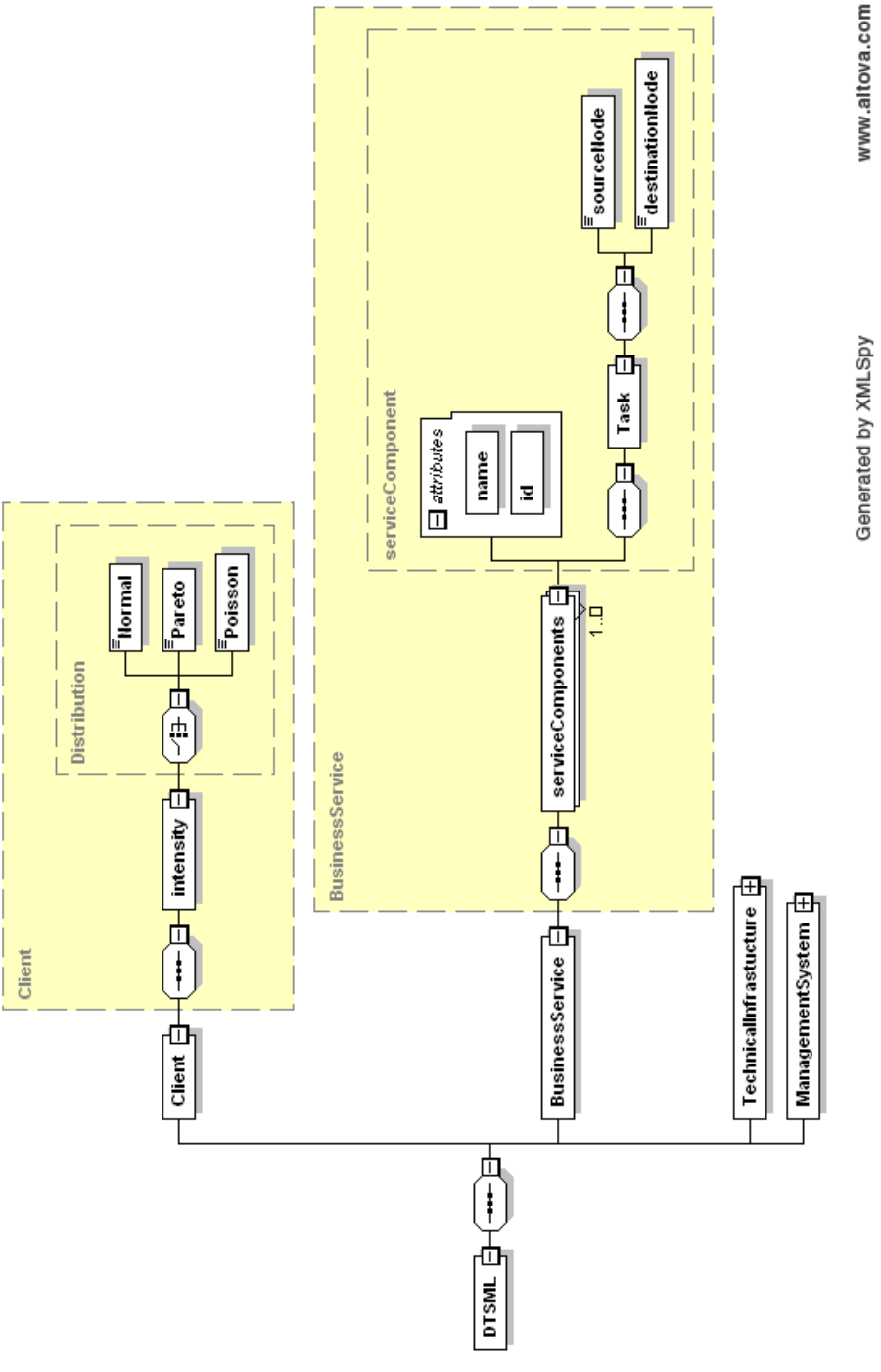


Fig. 1. XML Schema for Discrete Transport System Modelling Language

implementation of *SSF* due to much better documentation then that available for original *SSF* [2]. We have developed a generic class (named DTSObject) derived from *SSF* Entity [6] which is a base of classes modelling *DTS* objects like: scheduler, node, truck and crew which model the behaviour of presented in section 2 discrete transport system. Due to a presence of randomness in the *DTS* model the analysis of it has to be done based on Monte-Carlo approach. It requires a large number of repeated simulations. The *SSF* is not a Monte-Carlo framework but by simple re-execution of the same code (of course we have to start from different values of random number seed) the statistical analysis of system behaviour could be realized [10]. Data stored in output file can be used for different measures calculations.

6 Functional Availability of DTS

The analysis of a given system requires a metric. We propose to use the availability of the system. We define it as an ability of realising the transportation task in required time. The availability is a probability measure. Introducing the following notation [12]: T - time measured from the moment when the container was introduced to the system to the moment when the container was transferred to the destination (random value), T_g - guaranteed time of delivery, if exceeded the container is delayed, *N(t)* - stochastic process describing the number of delayed containers at time t, k - the level of acceptable delay. we can define the functional availability $A_k(t)$ as a probability that the number of delayed containers at time t does not exceed k, i.e.:

$$A_k(t) = Pr\{N(t) \leq k\}. \tag{10}$$

The calculation of stochastic process *N(t)* is based on analysing a state of each not yet delivered container. As illustrated in Fig. 2. we can observe two possible situations: (a) - delivery was realised before guaranteed time T_g - there is no delay, (b) - delivery was delayed - time of delay: T - T_g.

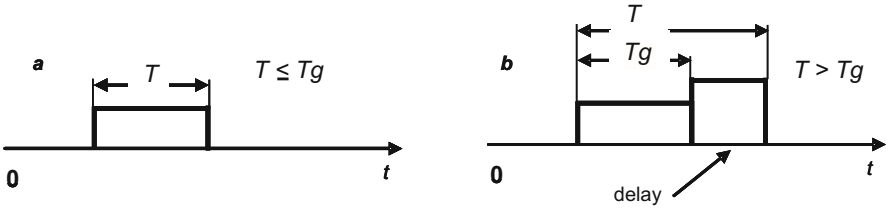


Fig. 2. The delivery in guaranteed time (a) and delayed delivery (b)

7 Case Study

For testing purposes of presented system (section 2) exemplar transport system was proposed. It consists of one central node (city Wroclaw, Poland) and three ordinary nodes (cities nearby Wroclaw: Rawicz, Trzebnica and Opole). The distances between nodes have been set approximating the real distances between used cities and they equal to: 85, 60 and 30 km. We assumed a usage of 5 trucks (two with capacity set to 10 and three with capacity 15) with mean speed 50km/h. The vehicles realized 19 trips a day: from central node to ordinary node and the return trip (i.e. Wroclaw-Opole). Failures of trucks were modelled by exponential distribution with mean time to failure equal to 1000h. The repair time was modelled by normal distribution with mean value equal to 2h and variance of 0.5h. The containers addressed to ordinary nodes were available in the central node at every 0.5, 0.4 and 0.3 of an hour respectively. Containers addressed to the central node were generated at every 0.6, 0.4, and 0.3 of hour in following ordinary nodes. There was a single maintenance crew. The availability of the system $A_k(t)$ was calculated with guaranteed time T_g=24h and parameter k=20. Based on 10 000 time simulations (in each 100 days) the availability of system was calculated. Results presented in Fig. 4. shows the periodic changes. The situation is an effect of used method of containers' generation. The containers are generated during all day (by Poisson process) but according to management system assumptions trucks do not operate in the night. The probability of delay increases at the night, but selected number of trucks (5) is satisfactory for given system. We have also analyzed a system with a reduced number of vehicles (with 4). The resulting value of the availability function is presented also in Fig. 4. It could be noticed that the availability of the system decreases due to lack of sufficient number of trucks. It should be noticed here that looking in the used management rules and not taking into consideration a randomness of the transport system (failures and traffic jams) only three vehicles should be enough to transport all the generated containers.

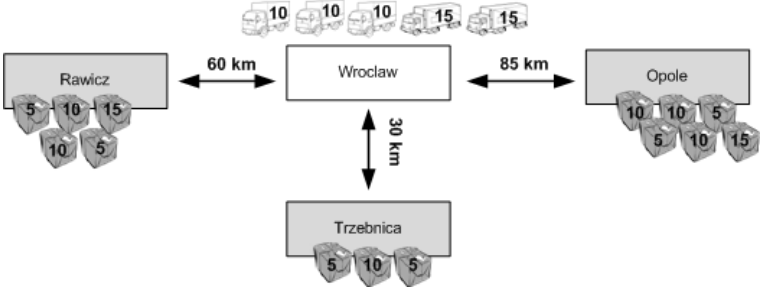


Fig. 3. Exemplary Discrete Transport System

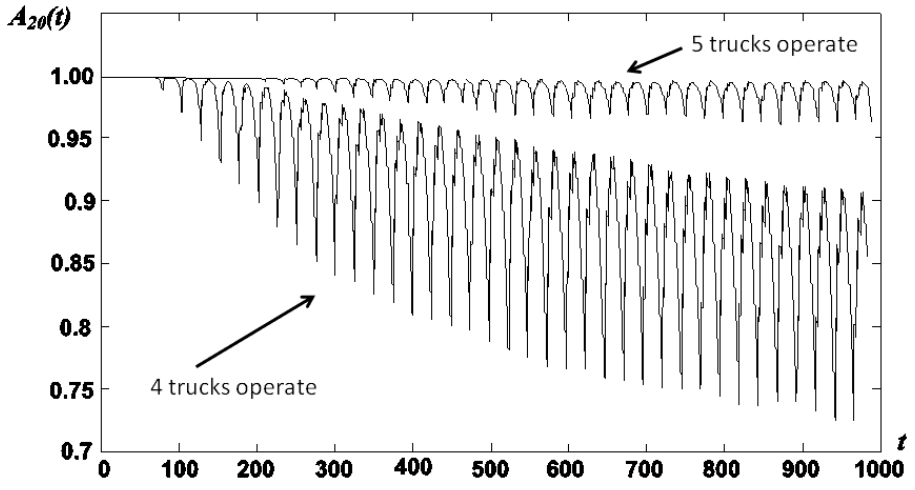


Fig. 4. Functional availability of *DTS*, 4 or 5 trucks operate

8 Conclusion

The paper presents an approach to discrete transport system modelling based on Polish Post delivering idea. The proposed description language is helpful for the creation of a computer-aided analysis tools processing according to *DTS* model. Main advantage is the language supports further usage of analysis tools. Moreover the format of the proposed language is fully expendable since its wide support of an XML tools. For this reason it can be seen as a base for further (even more precise) system description in case of both dependability and detailed description of the system. Proposed dependability metrics can be used as an example of attempt for dependability characteristics as it was shown on a case study. Further work is to extend proposed model (DTSML language) with more detailed description that allow even more detailed system specification as much as its accurate behaviour.

Acknowledgments. The presented work was funded by the Polish National Science Centre under contract no. 4759/B/TO2/2011/40.

References

1. Avižienis, A., Laprie, J., Randell, B.: Fundamental Concepts of Dependability. In: 3rd Information Survivability Workshop ISW 2000 (2000)
2. Cowie, J.H.: Scalable Simulation Framework API reference manual (1999), http://www.ssfnet.org/SSFdocs/ssfapiManual.pdf
3. Fishman, G.: Monte Carlo: Concepts, Algorithms, and Applications. Springer, Heidelberg (1996)

4. Gartner, N., Messer, C.J., Rathi, A.K.: Traffic Flow Theory and Characteristics. In: Board, T.R. (ed.), University of Texas at Austin, Texas (1998)
5. Gold, N., Knight, C., Mohan, A., Munro, M.: Understanding service-oriented software. IEEE Software 21, 71–77 (2004)
6. JooBeom, Y., Sohn, K., Hyunsoo, Y.: Dynamic Simulation on Network Security Simulator Using SSFNET. Convergence Information Technology, 1168–1172 (2007)
7. Liu, J.: Parallel Real-time Immersive Modeling Environment (PRIME), Scalable Simulation Framework (SSF), User's manual, Colorado School of Mines Dep. of Mathematical and Computer Sciences (2006), http://prime.mines.edu
8. Liu, K.S.: Studies on the Transfer Data Modeling in the Public Transport Information System. In: 7th International Conference on ITS, Telecommunications, ITST 2007, pp. 1–6 (2007)
9. Michalska, K., Walkowiak, T.: Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation. In: The Second International Conference on Emerging Security Information, Systems and Technologies SECURWARE 2008, pp. 356–361. IEEE Computer Society Press, Los Alamitos (2008)
10. Walkowiak, T., Mazurkiewicz, J.: Algorithmic Approach to Vehicle Dispatching in Discrete Transport Systems. In: Sugier, J., et al. (eds.) Technical approach to dependability, pp. 173–188. Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw (2010)
11. Walkowiak, T., Mazurkiewicz, J.: Analysis of Critical Situations in Discrete Transport Systems. In: Proceedings of International Conference on Dependability of Computer Systems, pp. 364–371. IEEE Computer Society Press, Los Alamitos (2009)
12. Walkowiak, T., Mazurkiewicz, J.: Functional Availability Analysis of Discrete Transport System Realized by SSF Simulator. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS (LNAI), vol. 5101, pp. 671–678. Springer, Heidelberg (2008)
13. Walkowiak, T., Mazurkiewicz, J.: Functional Availability Analysis of Discrete Transport System Simulated by SSF Tool. International Journal of Critical Computer-Based Systems 1(1-3), 255–266 (2010)
14. Walkowiak, T., Mazurkiewicz, J.: Reliability and Functional Analysis of Discrete Transport System with Dispatcher. In: Advances in Safety and Reliability - ESREL 2005, pp. 2017–2023. Taylor & Francis Group, Abington (2005)
15. Walkowiak, T., Mazurkiewicz, J.: Soft Computing Approach to Discrete Transport System Management. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS (LNAI), vol. 6114, pp. 675–682. Springer, Heidelberg (2010)

Swarm-Based Multi-agent Simulation: A Case Study of Urban Traffic Flow in the City of Wroclaw

Dariusz Król[1] and Maciej Mrożek[2]

[1] Institute of Informatics, Wrocław University of Technology
Wybrzeże S. Wyspiańskiego 27, 50-370 Wrocław, Poland
Dariusz.Krol@pwr.wroc.pl
[2] Faculty of Computer Science and Management, Wrocław University of Technology
Łukasiewicza 5, 50-371 Wrocław, Poland
Maciej.Mrozek@gmail.com

Abstract. Simulations using swarm-based multi-agent distributed architecture may provide appropriate tools to optimise traffic flow in today's traffic clogged cities. This paper describes an application of this approach in the city of Wroclaw (Poland) using ant algorithms and a multi-agent system. A model of the cities road network was generated which, combined with an analysis of traffic data, enabled traffic flow in the city to be simulated and then optimised. As a result, modifications were proposed to existing road layouts including delineation of a new road section. The results achieved clearly demonstrate the suitability of this approach for route optimisation.

Keywords: swarm intelligence, nature-inspired networking, multi-agent system, intelligent transportation system.

1 Introduction

This paper presents a simulation which can be used to model and optimise urban traffic flow and thus help solve one of the major problems affecting our cities. Currently there are several solutions proposed for this area using, for instance, meta-heuristics based on observation of natural occurrences like ant colony optimisation or particle swarm optimisation [1]. In particular, Alves et al. propose the Ant Dispersion Routing algorithm to determine recommended routes for every driver in the city [2]. Other solutions use a short-term traffic prediction methods using a multi-agent coordination system of road agents based on a pheromone communications model, which adaptively responds to increasing traffic congestion [3], [4]. Narzt et al. [5] have implemented a micro-simulation test bed, where vehicles drive along the shortest path considering different acceleration time, speed limits, traffic rules and safety distances. Scellato et al. [6] show how by controlling local knowledge, it is possible to minimize the overall congestion of the network and achieve results as good as those obtained with global traffic information. Most popular implementation platforms for these solutions use multi-agent environments [7], [8].

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 191–200, 2011.

The aim of this paper is to present how a model of a road traffic environment can be developed and then used to optimise transit traffic flow. Therefore, three elements of the model required particular attention: the national and regional highway networks passing through the city, movements of individual vehicles, traffic light timing and it's resultant impact on vehicle movement. The research is carried out using actual traffic data on selected roads within the city. It thus has the potential to provide an answer to the question whether the proposed algorithm could result in increased vehicle throughput and if so, what would be the extent of this improvement. This paper also proposes criteria which could be used to assess vehicle movement optimisation solutions. However, in spite of this additional benefit, the main purpose of this paper is to describe a tool to analyse existing traffic volumes and identify potential improvements in traffic transit time.

Although there have been many studies on urban traffic flow using the methods discussed above, there is no general approach to evaluating performance of swarm intelligent systems in terms of errors, or deviations from the ideal path.

This paper is organised as follows. Section 2 describes the model, it's environment and the simulation configuration. Section 3 presents simulation results using swarm agent based system. Finally, we conclude with a discussion of the results and their possible implications.

2 Traffic Flow Model

A traffic flow model was developed consisting of the following elements: roads, traffic lanes, road intersections, traffic lights and vehicle volumes. The model was developed in JAVA however, JADE was used to model crossroads as agents.

To ensure the model achieved it's expected results an ant algorithm was applied whereby vehicles moving on the roads in the model left pheromone depending on how long they would wait to cross a particular intersection. For congested roads with poorly coordinated traffic light sequences, an appropriate level of deposited pheromone would encourage traffic to be directed in an appropriate direction. This relationship between waiting time and pheromone concentration also worked as a signal initiating changes in traffic light sequences at the intersections so that vehicles could more rapidly exit from the congested area.

The proposed solution was developed on a JADE multi-agent framework. This gave the possibilities of dispersing the application across many computers plus making it robust in case of interchange calculation failure because, should one interchange fail, the calculations could easily be transferred to other computers in the network. Road intersections were to work as agents splitting vehicle movement into discontinuous sections where the agents themselves could decide on various signalling combinations or on the length of time taken for a particular traffic light cycle.

Route optimisation was only carried out on the network of national and regional roads going through the city ignoring roads which had a minor impact on the density of through traffic. The following roads/highways were included in the model: national highways DK5, DK8, DK94, regional highways DW320, DW336, DW337, DW327, DW322, DW342, DW347, DW362, DW356, DW349, DW395, DW452, DW453 and DW455 (see Fig. 1).

The most commonly encountered intersections in the city connect three to four roads and, whilst the model assumed that traffic moving in different directions would keep to different lanes, the model allowed traffic movement in each direction without restrictions. In this way intersections were allowed to operate as interoperating agents which themselves could take decisions to show a green light for the length of time be allowed to enter a particular road in a particular direction.

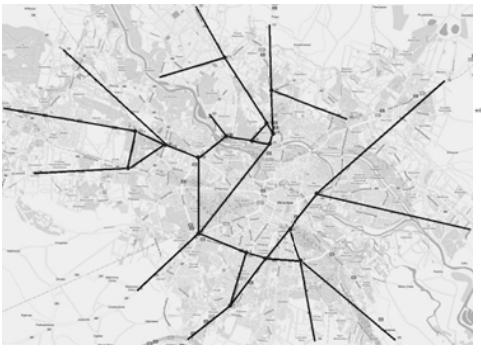


Fig. 1. Part of the Wroclaw street network used for the case study. Roads selected according to measured traffic flow are marked in black

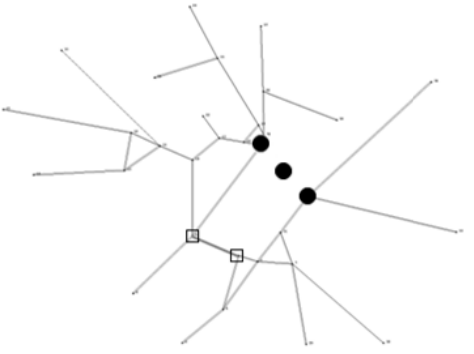


Fig. 2. Wroclaw transit routes. Proposed new route connection is marked by three black dots. Two squares marks the road where traffic volumes were measured to test the effectiveness of the new connection.

The model applied the following assumptions regarding vehicle traffic:

- Vehicles could not move faster then the speed limit in force on a particular section of road.
- The speed of vehicle movement was dependent on the nearest obstruction to be encountered by a particular vehicle (either a vehicle in front or an intersection).
- Depending on the distance between a moving vehicle and the obstruction, it was assumed that the moving vehicle takes appropriate actions to ensure a safe distance where this safe distance was defined as follows: $dist_{safe}[m] = \frac{v[\frac{km}{h}]}{10}$. Applying this formula a speed of 10 km/h resulted in a safe distance of 1 m. Obviously, in practice there could be a difference between the safe distance as calculated and actual driver behaviour. These approximations arose from actual observations and their main function was to model a real situation where vehicles start moving after a traffic light change or stop moving when encountering a red light. Vehicles which start to move after a traffic light change do not accelerate in a uniform manner thus each vehicle, depending on individual speed, has to ensure safe distance and, in accelerating, keep an appropriate distance from an obstruction.

Vehicles generated and added to the model had to move along defined roads and, after starting the algorithm, vehicle volumes were calculated in the following manner. As a first step, movements were generated for vehicles moving between city exit roads, i.e. those vehicles which were only passing through the city. This ensured that vehicles on

roads around the city centre going to destinations within the city boundaries were not disrupted by vehicles passing through the city simply because there was no other bypass route. In this way traffic density data for the roads around the city centre translated directly into the volume of vehicles which should be generated. Second, the combined traffic density on inner city roads was calculated and compared with data from the real traffic survey. If the traffic survey results for a particular route were higher then those generated by the algorithm this implied that a higher traffic volume should be generated by the algorithm i.e. that the algorithm should generate more vehicles along a particular route. If on the other hand, the traffic survey results were lower then the traffic volume along a particular route it implied that an excessive traffic volume was being generated and the additional vehicles generated should be evenly distributed across traffic volumes on other roads.

As mentioned earlier, a key element of applying an ant algorithm is the requirement to ensure accurate levels of pheromone are left by the moving ants. Therefore, in this particular application, the amount of pheromone left by a vehicle had to be inversely proportional to the time needed to drive over a specific intersection [9]. Thus, the higher the levels of pheromone left at an intersection, the shorter the time needed for vehicles to cross it.

Whilst running the algorithm the arithmetic mean of the pheromone levels left on the roads was measured. This mean was then used to define a percentage level of pheromone needed to "encourage" a vehicle to change it's route. Thus, when a vehicle encountered a pheromone level which deviated from the mean by a specified percentage value, it could change it's route on condition the new route also led to the same destination.

The model was based on a premise that each vehicle leaves a pheromone trace where the pheromone concentration is dependent on the length of time needed by that vehicle to drive over a specific intersection. Applying this premise, the cumulative pheromone deposit was expressed as follows :

$$\Delta\tau = \frac{1}{t_w} \cdot \rho_{ph},$$

where : $\Delta\tau$ - amount of pheromone increase due to one vehicle,
t_w - waiting time - time taken for a vehicle to cross a specific intersection,
ρ_{ph} - pheromone coefficient.

Pheromone evaporation took place when vehicle movements were recalculated. In this case the level of pheromone τ_1 in time interval t_1 was expressed as follows:

$$\tau_1 = \tau_0 - (t_1 - t_0) \cdot \rho_{ev},$$

where : τ_1 - level of pheromone at time interval t_1,
τ_0 - level of pheromone at time interval t_0,
$(t_1 - t_0)$ - time difference between successive updates of the pheromone level,
ρ_{ev} - evaporative rate.

The constant factors: pheromone coefficient and evaporative rate were set empirically for the following reasons. If the evaporative rate were to be set to a value which was too high (i.e. rapid pheromone evaporation) this would prevent a correct functioning of the algorithm because the deposited pheromone would disappear (evaporate) too quickly. On the other hand, too low a value would cause "interference" whereby the algorithm would not be able to adjust to such a high level of pheromone on the roads.

Likewise, the second factor - pheromone coefficient (level of pheromone left by one vehicle), also if incorrectly set, would have had a similar impact on the algorithm. It should however be noted that (logically) pheromone was much more frequently deposited on roads with high traffic density whereas in the case of roads with insignificant traffic the level of pheromone deposited by moving vehicles was close to zero. This was a factor of significant importance in the case of roads with significant traffic density because there had to be a pheromone trail both in the case of severely congested roads as also in the case of less frequently used roads.

3 Results of the Traffic Flow Simulation

Results are presented in two parts. The first part focuses on showing the general impact on traffic volume of a new transit route within the Wroclaw city area whereas the second part is concentrated mainly on route optimisation of the existing road network.

The evaluation data was from Wroclaw City Road and Communication Authorities. This was collected in a traffic noise survey carried out in 2004. This survey included data for all the national and regional roads within Wroclaw City boundaries. Using this data to generate the traffic flow model, the major criterion was assumed to be the number of vehicles at any one time actually present on the roads in the city. These simulations were carried out on a Centrino Duo 2.0 GHz, 1 GB RAM computer.

3.1 Delineation of a New Transit Route

Today, many larger cities have ring roads or bypasses enabling efficient distribution of traffic volumes. Quite often there are two rings, one functioning as an inner-city ring road and the second as a bypass at a significant distance from the city. Because the inner-city ring in Wroclaw is not yet completed, an analysis was made to simulate in what way an additional road connection could assist in lowering intra-city and transit traffic congestion.

It should be noted that it is a common opinion that the network of national and regional highways running through Wroclaw are underdeveloped. Moreover, analysing traffic data it soon becomes apparent there is a need for a road linking eastern parts of the city with highways going in a northerly direction. Delineating this additional link would complete the inner-city ring and would seem an essential step in a city where transit traffic is currently forced to go on city streets.

Bearing this in mind, the model was modified to include an extra section of transit road running from DK8/DW455 intersection to DK5/DW342 intersection. The proposed new road would therefore run along Sienkiewicza St, Drobnera St, Trzebnicka St and Zmigrodzka St. This new connection should significantly improve communication along transit routes running in a North-South and North-East direction. This proposed new connection is shown marked by black dots, see Fig. 2.

To analyse the effectiveness of the solution proposed above, traffic density was calculated for the roads and intersections where the impact of the change would be most significant. For this reason an existing road link was selected which had the

highest traffic density and which connected the South-East and North-West parts of the city (Haller Ave - marked by two squares on Fig. 2).

A traffic density analysis was carried out on both the existing road link and on the proposed new connection. A second analysis was carried out on the average waiting time for vehicles to traverse the intersection where the new connection was linked to traffic entering the city from the North-East. This second analysis was important because the existence of a new connection could actually extend the average vehicle waiting time at this intersection. Presented below is a comparison of traffic density as analysed on the major roads affected by the change.

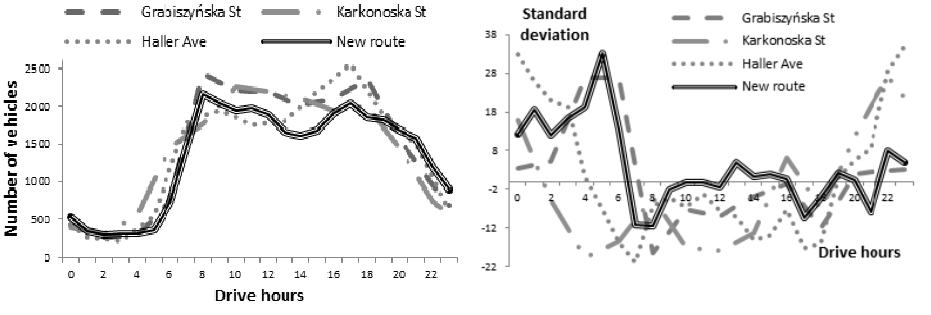


Fig. 3. Traffic density on the new connection compared with that on existing roads (left). Standard deviation of average traffic density per road (right).

Fig. 3 presents a percentage comparison of the variation of the standard deviation from average traffic density levels. Looking at the data, there is a clear spread between the affected roads which occurs between 19:00 and 07:00. However, as the "normal working day" starts, the standard deviation values come significantly closer together with around a 10% spread in the period from 08:00 to 18:00.

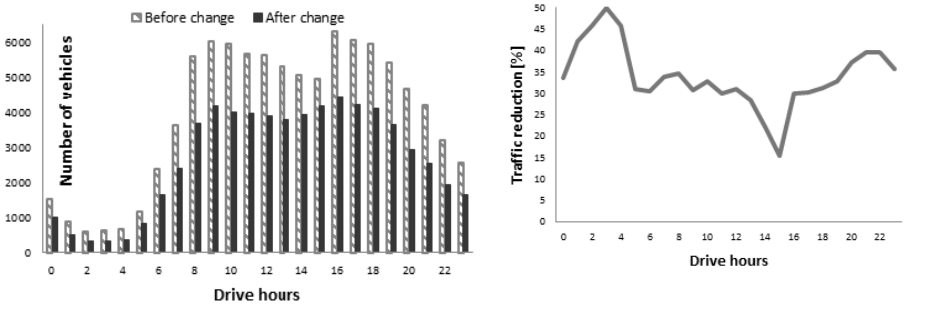


Fig. 4. Comparison of traffic volumes on Haller Ave before and after the proposed new road section is in use (left) percentage traffic volume reduction arising as a result of the proposed new road section (right)

To verify the impact of the proposed changes, a second analysis was performed on the average waiting time for vehicles to cross the intersection at Reagan roundabout (DK8/DW455 intersection) along with the decrease of traffic volume along Haller Ave. This analysis provided the results shown in Fig. 4.

As is evident from the above, there is a significant decrease in traffic volume. However, reviewing the results on a "per hour" basis, it became apparent that this reduction was not evenly distributed (Fig. 4). The largest reductions were during the late evening or at night when there is much lower traffic and thus much less vehicles for statistical analysis. However, in spite of this the results still showed a 33,89% reduction in road use when compared to the original model.

3.2 Application of Ant Algorithm for Traffic Flow Optimisation

A further analysis was carried out on the average length of time taken by a specific vehicle to cross the intersection at Reagan roundabout before and after implementation of the ant algorithm. Comparison of the results enabled an assessment of the effectiveness of the proposed solution. The results are presented below.

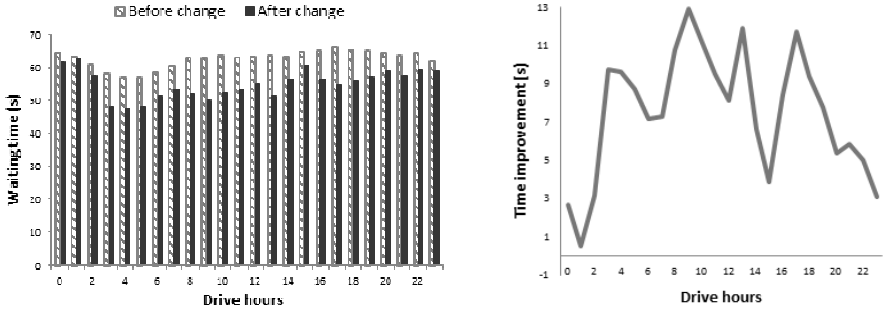


Fig. 5. Intersection waiting time before and after implementing an ant algorithm (left) Intersection waiting time improvement after implementation an ant algorithm (right)

From the graphs presented in Fig. 5 one can deduce that although the saving is not evenly distributed, the average time needed to cross a particular intersection was shortened by 7,5 sec. This lack of evenness in distribution is caused by two factors: first, the traffic density at particular times of the day (which obviously varies), second, the values taken for the constant factors evaporative rate and pheromone coefficient. The algorithm effectively handled variations in traffic volume however, analysis of the above graphs does indicate certain errors in the way it operated.

In the evening (i.e. from 20:00 onwards) the time needed to cross a particular intersection was reduced to almost zero (which was an obvious error). This was because the amount of pheromone left by vehicles traveling in peak traffic hours was still significant and had a knock-on “positive" effect on vehicles traveling when the roads were much clearer. This was especially noticeable on the most congested roads (e.g. Haller Ave) where this residual pheromone level caused the intersection agent to set traffic lights to green for one direction of movement because volumes in other

directions were not significant enough to deposit enough pheromone to oblige the agent to take a decision to reset the traffic lights. This situation was noticed on a number of intersections which in certain directions were significantly more difficult to cross in the evening hours. This situation would improve around 03:00 when the pheromone left on congested intersections had evaporated to the extent that the agents started to properly set traffic light timing. Unfortunately, in carrying out the tests, optimal parameters for evaporative rate and pheromone coefficient were not identified.

4 Discussion of Results

The model presented above proved very effective in simulating road traffic movement and the results of analysing and verifying the results indicate the application is suitable for use as a tool for traffic flow optimisation. Moreover, using the application, the environment was relatively easy to model and implement. The following paragraphs are devoted to a short discussion of the benefits and drawbacks.

The first benefit is the one stated above, the application provides a relatively simple way of modelling road networks. One can add new road connections or intersections to the model without greater difficulty. Traffic flow results or statistical data regarding average waiting time or the number of vehicles crossing a particular intersection can be easily simulated and analysed. Cumulation of this data was straightforward - an essential requirement to model traffic flow at peak periods. For this reason the model provides some simple to use tools for prediction of traffic flow, especially on new road sections.

A second very important factor is scalability. Implementing this application in the JADE environment enables distribution across multiple servers. As a result the application can be used to simulate traffic flow across very complex road networks with high volumes of traffic. This means that the possibility exists to successfully analyse traffic flow across a complete city network where the only barrier is the availability of distributed processing power to run the application. The only issue in dispersing the application across many calculation entities would be to carry out a fairly straightforward configuration of the main application container. Through this, future users of the application could without difficulty configure the environment and use it to carry out complex transportation analysis.

One of the drawbacks which came up when using the application was a problem caused by the incomplete modelling of the road network. The national and regional highways taken into consideration for the model did not provide a complete model of the road network in Wroclaw. Moreover, the roads included in the model often had other intersections with local roads and some of these intersections could have a significant impact on the way the system operated especially if there were traffic lights at this intersection. A second area not included was those intersections where no traffic lights exist. It would be useful to carry out research into how these intersections affect vehicle movement, average vehicle speed or traffic flow along a given road section.

Although, it turned out to be feasible to optimize average vehicle intersection transit times, the results turned out to be less then fully satisfactory because they

were lower then expectations. Optimisation resulted in a 7,5 second average saving which gave a 12% improvement compared to the non-optimised situation and further analysis revealed two interrelated items causing the relatively low optimisation results. The first item was the difficulty of identifying optimal parameters for the constant factors: evaporative rate and pheromone coefficient. In the context of the specific road network modelled for this application, using this specific network model it proved impossible to identify optimal parameters for these constant factors. The road network was the second item causing less then fully satisfactory optimisation results. As written earlier Haller Ave (together with it's connecting roads) is the only traffic artery connecting the South-East part of the city with the North-West. Therefore, "by definition" this road carries a large part of the volume going through the city. Applying the model, this led to significant pheromone deposits which resulted in such high levels of pheromone that traffic lights would continue working in an unoptimised manner long after the end of peak traffic periods. The best solution to this problem would have been to add all the city arteries to the model. However, whilst ensuring all traffic volumes were included, this would in turn have necessitated collecting additional traffic flow data for the additional roads added to the model.

Another aspect worth commenting is the difficulty of modelling vehicle behaviour when it is impeded by, for instance, a narrower road section, road-works or vehicle stoppage or breakdown. It proved extremely complex (bordering on the impossible) to model traffic flow in this situation. Additionally, the part of the application used for the simulation modelling could not take into consideration the manner in which vehicles cross a specific intersection. Examples of this are: vehicle access to direction lanes or the circumference of the route taken by vehicles to cross an intersection in a particular direction. One could envisage solving this particular problem using hybrid traffic flow models using graphs or cellular automation. The problem situations described earlier could also be modelled by applying some automation on the intersection entry/exit points along with a graph of the road network. Obviously, including complex road events would bring the benefits associated with much more accurate modelling of traffic flow. Moreover, using graphs would bring additional benefits connected with limiting the memory requirements for a large road network. However, up to the present, the authors have not come across any applications using this type of solution.

In the specific examples presented in this paper, once the model had been constructed, it was then used to study two specific areas. Firstly, how to "de-block" some of the most congested roads in the city by adding a communication link to the existing road network. The results showed that the proposed route could significantly reduce vehicle congestion in the most congested parts of the city. We have a plan to repeat our simulations on newer data and we predict the results, even better. The second area of study was to implement a system to analyse the optimisation potential of the existing road network. Although the efficiency of the system could be improved, the results of the analysis were fairly positive and could well indicate a simple jet potentially quite powerful approach to modelling city road environments. Analysing the results, deficiencies were identified which were due to the accuracy of the road network model and to the way in which the ant algorithm was applied. On the other hand, a significant advantage of this solution is the overall accuracy with which it modelled traffic flow. Thus it enabled a detailed analysis of the impact of

changes in the road network. A second advantage is the possibility of dispersing the application across many calculation units. This gives the possibility of modelling traffic flow in very complex road networks and for these reasons, the solution described in this paper may be very useful in assisting efficient traffic flow in our rapidly growing cities.

Acknowledgments. This paper was partially supported by Grant no. N N519 444939 funded by Polish Ministry of Science and Higher Education (2010-2013).

References

1. Teodorovic, D.: Swarm intelligence systems for transportation engineering: Principles and applications. Transportation Research Part C: Emerging Technologies 16(6), 651–667 (2008), doi:10.1016/j.trc.2008.03.002
2. Alves, D., van Ast, J., Cong, Z., De Schutter, B., Babuska, R.: Ant Colony Optimization for Traffic Dispersion Routing. In: 13th International IEEE Conference on Intelligent Transportation Systems, pp. 683–688 (2010), doi:10.1109/ITSC.2010.5625146
3. Kurihara, S., Tamaki, H., Numao, M., Yano, J., Kagawa, K., Morita, T.: Traffic Congestion Forecasting based on Pheromone Communication Model for Intelligent Transport Systems. In: IEEE Congress on Evolutionary Computation, pp. 2879–2884 (2009), doi:10.1109/CEC.2009.4983304
4. Ando, Y., Fukazawa, Y., Masutani, O., Iwasaki, H., Honiden, S.: Performance of pheromone model for predicting traffic congestion. In: 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 73–80 (2006), doi:10.1145/1160633.1160642
5. Narzt, W., Wilflingseder, U., Pomberger, G., Kolb, D., Hortner, H.: Self-organising congestion evasion strategies using ant-based pheromones. Intelligent Transport Systems 4(1), 93–102 (2010), doi:10.1049/iet-its.2009.0022
6. Scellato, S., Fortuna, L., Frasca, M., Gómez-Gardeñes, J., Latora, V.: Traffic optimization in transport networks based on local routing. The European Physical Journal B 73(2), 303–308 (2010) doi:10.1140/epjb/e2009-00438-2
7. Król, D., Drożdżowski, M.: Use of MaSE methodology for designing a swarm-based multi-agent system. J. of Intelligent & Fuzzy Systems 21(3), 221–231 (2010), doi:10.3233/IFS-2010-0453
8. Król, D., Popiela, Ł.: Modelling Shortest Path Search Techniques by Colonies of Cooperating Agents. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 665–675. Springer, Heidelberg (2009)
9. Renfrew, D., Yu, X.H.: Traffic Signal Control with Swarm Intelligence. In: 5th International Conference on Natural Computation, pp. 79–83 (2009), doi:10.1109/ICNC.2009.653

Evolving Equilibrium Policies for a Multiagent Reinforcement Learning Problem with State Attractors

Florin Leon

Department of Computer Science and Engineering, Technical University of Iaşi,
Bd. Mangeron 53A, 700050 Iaşi, Romania
fleon@cs.tuiasi.ro

Abstract. Multiagent reinforcement learning problems are especially difficult because of their dynamism and the size of joint state space. In this paper a new benchmark problem is proposed, which involves the need for cooperation, competition and synchronization between agents. The notion of state attractor is introduced, such that agents compute their actions based on the proximity of their current state to the nearest state attractor. A genetic algorithm is used to find the state attractors. This representation can be used as a compact way to define individual or joint policies.

Keywords: multiagent reinforcement learning, benchmark problem, game theory, Nash equilibrium, genetic algorithm, state attractors, policy search.

1 Introduction

A multiagent system is a group of autonomous, interacting entities that share a common environment. While they operate on local knowledge and possess only limited abilities, they are however capable of enacting the desired global behaviors [13].

Reinforcement learning is a convenient way to allow the agents to autonomously explore and learn the best action sequences that maximize their overall value, based on successive rewards received from the environment [11].

A multiagent reinforcement problem adds an additional level of complexity. Since classic algorithms such as *Q-learning* [14] estimate the values for each possible discrete state-action pair, each agent causes an exponential increase of the size of the state-action space. Another challenge is the implicit or explicit need for coordination, since the effect of an agent's action also depends on the actions of the other agents. If their actions are not consistent, the overall goal may be impeded.

Learning should usually exhibit adaptation to the changing behavior of the other agents and stability, i.e. convergence to a stationary policy [3]. Usually, stability assumes the convergence to equilibria, which means that the agents' strategies should eventually converge to a coordinated equilibrium. Nash equilibria are most frequently used for this purpose. Adaptation ensures that performance is maintained or improved as the other agents change their policies as well.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 201–210, 2011.

Several learning algorithms have been proposed for multiagent systems, among which we could mention: *Fictitious Play* [2], *Minimax-Q* [7], *Nash-Q* [5], *Team-Q* [8], *WoLF-PHC* [1], *AWESOME* [4].

The main goals of this paper are to propose a new benchmark problem for cooperative and competitive multiagent reinforcement learning (MARL), provide a game theoretic analysis of rational behavior among competitive agents, and an evolutionary method to detect a compact representation of agent policies.

2 The Proposed Benchmark Problem

One of the main difficulties in dealing with the multiagent version of a reinforcement learning or planning algorithm is the great increase of the state space, since the search for a solution must take into account the combination between all the actions of individual agents. Also, the environment in such a setting is no longer static, in the acception of Russel and Norvig [10], but dynamic, because it is not only the actions of a particular agent that determine the next state of the environment, but the actions of all the other agents. Therefore, it is not enough for an agent to learn how to react best to the environment; it must also adapt to the models of the other agents. In this respect, the dynamism of the problem makes it similar in a way to the moving target learning problem: the best policy changes as the other agents' policies change [3].

Many test problems that deal with multiagent reinforcement learning are simple problems, involving two or only a few agents, in environments with a rather small size. In this section, we will consider a more realistic problem, with 8 agents executing in a 10 x 10 grid world with obstacles. The agents must retrieve and deliver 3 blocks in a designated order.

The proposed problem is presented in figure 1. The agents are marked with circles and the blocks are represented by gray squares marked with rectangles.

The goal of the agents is to move the 3 blocks (*K1*, *K2*, *K3*) to the *Goal* state, in this respective order. There are 3 types of agents: *A*, *B*, *C*, *D* (these types can also be interpreted as roles that these heterogeneous agents can play). There is only 1 agent of type *A*, 2 agents of types *B* and *C*, and 3 agents of type *D*. The blocks can be carried only by a specific combination of agent types: *K1* can be carried only by an *A* together with a *B* agent, *K2* can be carried by 3 agents of types *B*, *C*, and *D*, while block *K3* can be carried by a *B* and a *D* agent.

The world has 2 types of obstacles: a "hard" wall, displayed in dark gray, which no agent can cross, and a "soft" wall, displayed in light gray, which only an agent of type *C* can cross (with a penalty denoting its greater "effort").

The agents can perform 7 actions: move in the four axis-parallel directions (*Left*, *Right*, *Up*, *Down*), *Pick up* an object, *Put down* an object, and *Wait* (perform no action). An important feature is that an "object" can refer both to a block and to another agent. Therefore, agents in this setting can directly act on their peers.

Agents can move into or through a cell occupied by another agent. They execute in a tick-based manner, all in parallel.

There are several rewards given to the agents in different situations: *1)* moving 1 square on the grid: $r = -1$; *2)* moving 1 square on the grid while carrying an object: $r = -2$; *3)* a *C* agent moving through a "soft" wall square: $r = -3$; *4)* hitting an

uncrossable obstacle, including the "edges" of the grid, or a block that is not meant for its type: r = -5; *5)* waiting, picking up or putting down another agent: r = 0; *6)* picking up a block: r = 100; *7)* putting down a block to the *Goal* state: r = 100; *8)* completing the global task (all 3 blocks being delivered): r = 5000 to all the agents.

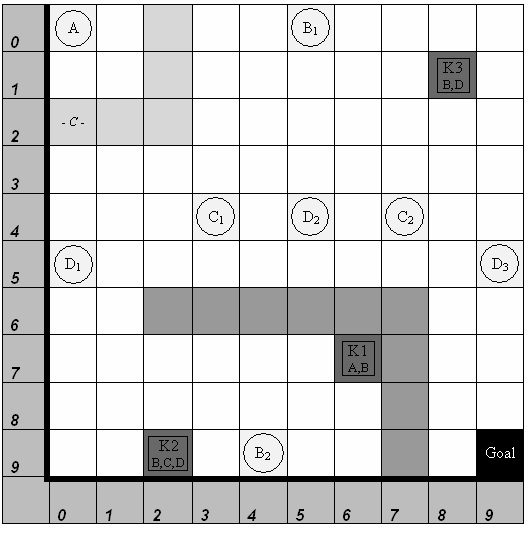


Fig. 1. The initial configuration of the proposed benchmark problem

Since the rewards are subject to a discount factor $\gamma < 1$ (e.g. $\gamma = 0.95$), the maximization of the total value of the agents also implies the minimization of the number of time steps needed to solve the problem.

In order to be solved in a near-optimal way (or at all), the problem requires the agents to make use of several high-level mechanisms, analyzed in the following subsections.

2.1 Cooperation

The problem goal requires agents of different types to cooperate in order to achieve the ultimate reward. Moreover, the environment is set in such as way that some agents must demonstrate entirely altruistic behavior. For example, the move of the first block, *K1*, is only possible if an *A* agent is present to carry it. However, agent *A* cannot come to its location by itself because it cannot go through the "soft" wall. One of the two *C* agents must help agent *A*. If one *C* agent does so, the other C agent will move to get block *K2*, thus obtaining a reward of 100. If no *C* agent helps *A*, there is no solution to the overall problem. The completion reward of 5000 can "motivate" a *C* agent to help *A*.

2.2 Competition

Since there are more agents of the same type, they must analyze the best individual goals based on their locations in the environment. Some goals may yield greater rewards than others (e.g. getting a certain block), but those individual goals may not be obtainable because other agents of the same type can achieve them first (i.e. if they are closer to that particular block).

The best decisions in these situations can be reached with the help of game theory, and an analysis is presented in section 3 for all agents involved.

2.3 Synchronization

A weak form of synchronization appears because two or three agents must come to the location of a block in order to move it. A stronger, more interesting form of synchronization is needed in order to find the solution as quickly as possible. Thus, the moving blocks must arrive to the *Goal* state one after another, if possible. This translates into a synchronization phase between the moving blocks, such that a block should "start" when another block passes through a particular, appropriate location.

2.4 Internal States

There are multiple, consecutive subgoals to be achieved by the agents (move to a block, carry it to the *Goal* state, possibly return for another block, and carry the latter to the *Goal*). The need to carry more than one block appears because all three blocks need a type *B* agent, but there are only two type *B* agents present. Therefore, one *B* agent must eventually carry two blocks.

Especially in this case, it is difficult to find a unique optimal policy that makes the agent move virtually on the same trajectory in opposite directions with different goals. It is more convenient to assume that agents can have internal states that are triggered by certain events, such as picking up an object or dropping it to the *Goal*. Thus the behavior of the agents is no longer based on a first order Markov process, where the current state S_t depends only on the previous state S_{t-1}: $P(S_t \mid S_{t-1}, \ldots, S_0) = P(S_t \mid S_{t-1})$.

3 Game Theoretic Analysis of Competitors' Behavior

In order to reduce the size of the state space when searching for agent policies, we use a game theoretic approach as a heuristic or pre-processing phase. We thus analyze what is the rational meta-action (or decision regarding an individual subgoal) for the agents belonging to the three types that involve competition. There is no competition in the case of type *A*, because there is only one such agent.

3.1 Type *B*

B_1 and B_2 agents are located in opposite sides of the grid, next to *K3* and *K1* or *K2*, respectively. In order to minimize losses from negative rewards associated with movement (r = -1), agents should strive in general to move to their nearest block and get the corresponding reward of 100. One can notice from figure 1 that B_2 is closer to

K2 than to *K1*. This leads to the short-term analysis presented in table 1. The choice regarding which block to pursue is equivalent to a game in normal (or strategic) form where the utilities are represented by the sum of rewards received by each agent. For clarity and simplicity of computation, we will consider that the discount factor $\gamma = 1$.

The cells of table 1 show the utilities received by B_1 and B_2, respectively, while following all the combinations of the 3 actions: pursuing block *K1*, *K2*, or *K3*. The calculation of utilities is based on the reward model described at the beginning of section 2. For example, in the first cell (-16, 96), we consider the situation where both agents pursue block *K1*. Agent B_2 will reach it first, after 4 moves, and on completion, it will get a reward of 100. Therefore, it will get a corresponding utility of $4 \cdot (-1) + 100 = 96$. Agent B_1 will arrive later and thus get a utility of only -16, for moving 16 squares. According to the Nash equilibrium analysis, columns *K1* and *K3* are dominated, because B_2 will get a utility of 98 by choosing *K2*, rather than getting 96 by choosing *K1*, or 82 or -18 by choosing *K3*. If B_2 is rational, it will always choose *K2*, irrespective of what B_1 does. Therefore, B_1 should assume this and try to maximize its own reward under these circumstances. The best B_1 can do when B_2 chooses *K2* is to choose *K3*. In this case, B_1 will receive the maximum reward out of the three possibilities: 84, -14, or 96.

Thus, the pure Nash equilibrium of this game is for B_1 to get block *K3* and for B_2 to get *K2*. The corresponding cell is marked in bold.

Table 1. Short-term utilities of *B* type agents

		B_2		
		K1	*K2*	*K3*
B_1	*K1*	-16, 96	84, 98	84, 82
	K2	86, 96	-14, 98	86, 82
	K3	96, 96	**96, 98**	96, -18

Table 2. Long-term utilities of *B* type agents

		B_2		
		K1	*K2*	*K3*
B_1	*K1*	-	**319, 164**	-
	K2	152, 331	-	-
	K3	**178, 307**	x	-

However, this does not take into account the fact that since block *K1* is not taken, the overall problem cannot be solved. Also, it is easy to see that the *B* agent that gets block *K1* will also have a chance to get the third block, because it will deliver the block to the *Goal* and will be free sooner than the second agent.

The long-term analysis of the situation is presented in table 2. The dominated strategies where an agent can have a negative utility were marked with "-". Also, the strategy that was previously the Nash equilibrium is now marked with "x", because it fails to solve the problem, and thus it is dominated by the strategies leading to the final reward of 5000. The utilities presented in table 2 do not include this final reward.

Similar to the computations for table 1, the resulting utilities are the sums of the rewards for moving on the shortest path to the targets and for picking up the targets. Finding the optimal path is a different problem which can be solved itself by reinforcement learning if the environment is initially unknown. In order to simplify the solution and concentrate on higher level issues, we assume that the environment is

accessible and discrete, and the agents have means to compute shortest paths, e.g. by applying the A* algorithm.

This game has two Nash equilibria, marked in bold. The subgame formed by strategies (*K1*, *K3*) for B_1 and (*K1*, *K2*) for B_2 has a mixed Nash equilibrium where the agents can stochastically choose either actions with probabilities P(B_1, *K1*) = 0.65 and P(B_2, *K1*) = 0.64.

In the following, for our case study, we will consider the pure equilibrium (178,307), because if we assume a cooperative behavior, both the sum of utilities (utilitarian solution) and their product (Nash solution) are greater than for the (319,164) equilibrium. In this case, agent B_1 gets block *K3* and then waits until block *K2* is moved to the *Goal*. The optimal strategy for B_2 is to get *K1*, move it to the goal, then return for block *K2*, and also carry it to the *Goal*.

3.2 Type *C*

Table 3 presents the utilities of the two *C* type agents when the subgoals are either picking up block *K2* or picking up agent *A*. Of course, since agents can pick up any object in their environment, there are many other possible subgoals. However, we chose to analyze these ones because only they can contribute to a solution to the problem. The other meta-actions are irrelevant and could only increase the number of time steps needed to reach the overall goal. We also included the *Wait* action for C_2, because otherwise the Nash equilibrium would force it to accept a negative utility.

Table 3. Short-term utilities of *C* type agents

		C_2		
		A	*K2*	*Wait*
C_1	*A*	-9, -13	-9, 88	-9, 0
	K2	92, -13	92, -12	**92, 0**

In this case, the pair of strategies (*Move to K2*, *Wait*) is the pure Nash equilibrium of the game. This would also mean that no block would be taken to the *Goal*, because agent *A* cannot reach *K1*, and C_2 has no incentive to help it.

However, in the long run, C_2 can benefit from the completion of the joint task, therefore, its dominant strategy becomes *Move to A*.

3.3 Type *D*

Table 4 presents the normal form of the game played by *D* agents. In this case, the Nash equilibrium requires D_1 and D_3 to move to their nearest blocks, and D_2 to *Wait*, because if the other two agents behave rationally, it cannot pick up any of the two blocks *K2* or *K3*. The three tables can be aggregately viewed as a 3D matrix, with one axis corresponding to one of the three agents involved. The utilities displayed in the cells are in the following order: $\left(u(D_1),u(D_2),u(D_3)\right)$.

Table 4. Utilities of *D* type agents

$D_2 - K2$		D_3	
		K2	*K3*
D_1	*K2*	(94, -10, -13)	(94, -10, 95)
	K3	(88, 90, -13)	(-12, 90, 95)

$D_2 - K3$		D_3	
		K2	*K3*
D_1	*K2*	(94, 94, -87)	(94, -6, 95)
	K3	(-12, 94, 87)	(-88, -6, 95)

$D_2 - Wait$		D_3	
		K2	*K3*
D_1	*K2*	(94, 0, -13)	**(94, 0, 95)**
	K3	(88, 0, 87)	(-12, 0, 95)

The game has a pure Nash equilibrium point, marked in bold.

4 Evolving Policies Based on State Attractors

Even with the simplifications provided by the analysis regarding the rational subgoals of the agents, the space of possible joint actions remains large. A classical approach in the reinforcement learning field is to find a way to reduce the size of the state-action mappings (such as the *Q* matrix), e.g. by using a neural network to approximate the learned policy [12].

An alternative way to compress the state-action mapping is to use a *nearest neighbor classifier* to determine the current action based not necessarily on the current state, but on the nearest state with a defined action. In our case, in order to develop individual policies that would perform well when combined in the multiagent system, we resort to the use of *state attractors*, i.e. states when an action is defined, such that an action in another state is computed as the action of the closest state attractor. The state attractors can be viewed as the center points of a Voronoi diagram [15]. When an agent enters the corresponding region, it will follow only the action specified by its center.

A previous article [6] showed, for a single agent scenario dealing with the mountain car problem [9], that classification can be a helpful way to reduce the size of the state-action mapping by up to 90%.

In this paper, a genetic algorithm was used to discover the state attractors. Separate searches were performed for every agent when moving individually. When several agents are jointly involved in moving a block to the *Goal* state, a single joint policy was searched for.

Since there can be more state attractors for a certain policy, a variable length encoding was used. A chromosome is thus formed of a variable number of triplets denoting the attractor position (line and column of the grid, from 0 to 9) and action (from 0 to 6). The structure of a chromosome is displayed in figure 2.

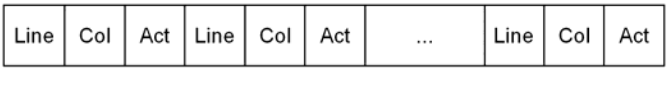


Fig. 2. The structure of a chromosome

The crossover can be performed by choosing two different positions in the two parents, but multiple of 3, and then joining the different ends from the parents in order to form two new children. Mutation can be performed on any position and consists in resetting the gene value to a random value in its definition domain. The selection method is the tournament with 2 individuals. Elitism is used, i.e. the best individual is directly copied into the next generation, in order not to lose the best solution of a generation.

Especially at the beginning of the GA search, the policies are not valid and do not make the agent go to a terminal state (its subgoal). In this case, in order to prevent an infinite loop, e.g. by continuously hitting a wall and keeping the same position, we restricted the application of the policy to a maximum of 200 steps.

The fitness function is given by the total sum of discounted rewards by following the policy:

$$F(\pi) = \sum_{t=0}^{N} \gamma^t \cdot R(S^{\pi;t}), \tag{1}$$

where π is the policy resulting from the actual execution of the agent guided by the state attractors, and N is the number of steps to the terminal state, $N \leq 200$. R is a function giving the reward in state S.

When two chromosomes have the same fitness value, the shorter chromosome is preferred in the tournament. Thus, the GA will try to find the smallest number of state attractors that define an optimal or near-optimal policy.

4.1 Simple State Attractors

The state attractors for the agents involved in reaching the first subgoals (i.e. reaching a block) found by the GA are displayed in figure 3. The dotted lines show the actual trajectories of the agents.

4.2 Joint State and Double State Attractors

As mentioned above, when two or three agents move a block, the GA finds a unique policy for the block movement. This is shown in figure 4 where state attractors have names such as *K1*, *K2*, or *K3*. Also, figure 4 shows the trajectory of B_2 when returning from the *Goal* state in order to get block *K2*.

The double state attractors are more interesting. In this case, the movement of a block also takes into account the position of another. In this way, the system can exhibit a means of synchronization.

Since block *K3* must arrive to the *Goal* just after *K2*, so that the time to completion should be minimized, its state attractors also take into account the state of *K2*. This is achieved by slightly modifying the encoding of the chromosomes to have two positions instead of one, and the crossover being executed onto sets of genes, multiples of 5.

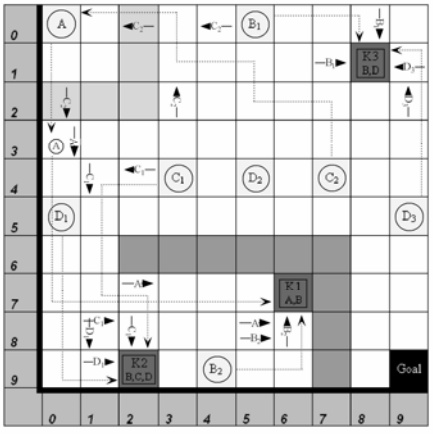


Fig. 3. Simple state attractors and agent trajectories

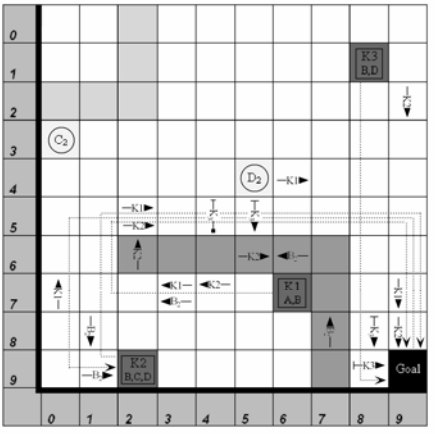


Fig. 4. Joint state and double state attractors and agent trajectories

The state attractors in this case are displayed in table 5.

Table 5. Double state attractors for joint movement of block *K3*

Line *K3*	Column *K3*	Line *K2*	Column *K2*	Action *K3*
1	8	5	4	*Wait*
1	8	5	5	*Down*
8	8	9	9	*Down*
9	8	9	9	*Right*
9	9	9	9	*Put Down*

The effect is that *K3* starts when *K2* approaches and it keeps only a square distance, so that it is delivered to the goal state one tick after *K2* is delivered.

Overall, following these policies, a solution to the problem is found in 89 time steps. Since a GA is used to find the policies, the results are not always optimal. For example, when B_2 returns to get block *K2*, it moves down on column 0 instead of column 1, thus moving 2 additional squares. However, the use of GA has the advantage of simplicity at the expense of execution time to obtain a near-optimal solution.

5 Conclusions

This paper presented a proposed multiagent reinforcement learning problem involving both cooperation and competition. Also, a way to detect a compact representation of policies by using state attractors was described, and a genetic algorithm with variable chromosome length was used to detect them. A game theoretic approach regarding the initial establishment of subgoals among competitive agents was shown to be useful as a preprocessing phase in order to reduce the size of the search space. Due to its

generality and suitability for complex situations, we consider that this method can also be successfully applied to other MARL benchmark problems.

Acknowledgments. This work was supported by CNCSIS-UEFISCSU, project number PNII-IDEI 316/2008, *Behavioural Patterns Library for Intelligent Agents Used in Engineering and Management*.

References

1. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence 136, 215–250 (2002)
2. Brown, G.W.: Iterative solutions of games by fictitious play. In: Koopmans, T.C. (ed.) Activitiy Analysis of Production and Allocation, ch. XXIV, pp. 374–376. Wiley, Chichester (1951)
3. Buşoniu, L., Babuška, R., De Schutter, B.: Multi-Agent Reinforcement Learning: An Overview. In: Srinivasan, D., Jain, L. (eds.) Innovations in Multi-Agent Systems and Applications. SCI, vol. 310, pp. 183–221. Springer, Heidelberg (2010)
4. Conitzer, V., Sandholm, T.: AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In: Proceedings of the 20th International Conference on Machine Learning, ICML 2003, Washington, US, pp. 83–90 (2003)
5. Hu, J., Wellman, M.P.: Multiagent reinforcement learning: Theoretical framework and an algorithm. In: Proceedings of the 15th International Conference on Machine Learning, ICML 1998, Madison, US, pp. 242–250 (1998)
6. Leon, F.: Intelligent Agent Planning with Quasi-Determined States Using Inductive Learning, Bulletin of the Polytechnic Institute of Iaşi, Romania, tome LVI (LX), section Automatic Control and Computer Science, fasc. 3, pp. 27–42 (2010)
7. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th International Conference on MachineLearning, ICML 1994, New Brunswick, US, pp. 157–163 (1994)
8. Littman, M.L.: Value-function reinforcement learning in Markov games. Journal of Cognitive Systems Research 2(1), 55–66 (2001)
9. Moore, A.: Efficient Memory-Based Learning for Robot Control, Ph.D. Dissertation, University of Cambridge (1990)
10. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2002)
11. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
12. Tesauro, G.: Temporal Difference Learning and TD-Gammon. Communications of the ACM 38(3), 58–68 (1995)
13. Vidal, J.M.: Fundamentals of Multiagent Systems with NetLogo Examples (2007), http://jmvidal.cse.sc.edu/papers/mas.pdf
14. Watkins, C.J.C.H.: Learning from Delayed Rewards, PhD Thesis, King's College, Cambridge University (1989)
15. Wikipedia, The Free Encyclopedia: Voronoi diagram (2011), http://en.wikipedia.org/wiki/Voronoi_diagram

Agent Based Simulation of Customers Behavior for the Purpose of Price Distribution Estimation

Marek Zachara and Cezary Piskor-Ignatowicz

AGH University of Science and Technology, Krakow, Poland
{mzachara,ignatow}@agh.edu.pl

Abstract. Price dispersion is an observed variation of price of the same (or similar) product among different vendors. This paper provides a possible explanation of the observed shape of the dispersion, proven by simulations of an agent based client-vendor environment. Proposed models for client and vendor representation are included. It turns out that the observed shape is achieved when some key environmental elements are taken into account; i.e. communication between clients, a limited memory of a client and the cost of crossing the distance between agents. As a benefit of the proposed model, it allows for speculation on how the future commerce may look like - in an Internet world where distances matter little.

Keywords: price dispersion, clients behavior, simulation.

1 Introduction

Price is a key element of virtually any economic activity. Even though its not always explicitly mentioned, its always present - often as a property of cost (e.g. cost of the alternative, cost of supplies, etc.). Price in general is considered to be a ratio of exchange between two types of goods, usually expressed in certain monetary units [3]. Thorough this article two types of price will be used: *a transactional price* (i.e. a price at which transaction between a supplier and a purchaser is carried out) and an *offered price* (i.e. a price that one side of the transaction is willing to accept, but the other might not always do so - in which case the transaction is not concluded).

The transactional price can be determined in various ways. Probably most common is a free market scenario, where suppliers and buyers freely negotiate the price until they come to the point when they are both willing to perform the transaction - or they abandon the negotiations. Another major option is a fixed price scenario, where the price is set (usually by the authorities) and all transactions can be performed only at this set price. The later usually leads however to under/over supply of the traded commodity, or development of an illegal (black) market where the transactions are performed at free market prices [5].

As mentioned above, price is a central issue of economic activities. Economics, like other sciences, strives to build theories that would allow to predict an outcome of conditions not observed before. Yet it lacks the primary scientific tool;

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 211–220, 2011.

i.e. a controlled experiment. For obvious reasons, it is not possible to lock large groups of humans in a controlled environment for extended periods of time to test scientific theories on them. As a result, Economics relies mainly on post-factum data analysis, with modeling and simulation added to the portfolio of tools in recent years. Certainly, all of us are subject to economic experiments - as the authorities continuously try to adjust the environmental conditions of individuals, groups and nations. Yet, this does not meet scientific criteria (controlled variables, stable environment, defined observables), so in reality is useful only for the 'post-factum' analysis.

It is necessary to mention that all Economics' theories are based on sets of data collected through observation of the real world. Modeling and simulation offer some advantage over the traditional analysis of such data in terms of observability, they are however still subject to insufficient input and unproven assumptions.

2 Price Evaluation and Price Dispersion

In a free market economy, the price is usually set by sellers based on the feedback they receive from buyers.

For a client, the following model of perceived attractiveness of a certain seller's offer is proposed:

$$A = aV_q + bV_s - cS - dC_r \ . \tag{1}$$

Where:

- A is the perceived attractiveness
- V_q is the assumed value of the considered item. It is a property of each specific item and may include physical quality, brand value among peers, etc.
- V_s is the expected quality of service from the seller
- S is the transactional price
- C_r is the cost of purchase (e.g. time needed to reach the shop)
- a, b, c, d are factors denoting personal preferences

If the client decides to purchase a particular type of item or service, he or she will choose a seller with the greatest perceived attractiveness, which in turn depends on the offered price (S), but also on other, less easy to measure factors.

Sellers try to maximize the overall profit by adjusting their offer. The profit earned can be defined as:

$$P = n\left(S - C_p\left(n\right)\right) - C_s\left(n\right) - C_c \ . \tag{2}$$

Where:

- P is the total profit for the defined period of time
- S is the transactional price
- $C_p(n)$ is the cost of purchasing or producing the item being sold
- $C_s(n)$ is the cost of sales (e.g. staff, packaging)

- C_c is the constant operating costs (e.g. rent)
- n is the number of transaction during the defined period (e.g. a month)

To maximize the overall profit, sellers can adjust several factors of their equation:

- Adjust their product portfolio. This will change the C_p component of the equation (2), but will also affect the V_q of the client's equation (1)
- Adjust the offered price. This directly changes the profit generated and the perceived offer attractiveness.
- Adjust their quality of the service. That will affect the C_s component of the seller's equation (2) and V_s component of the client's equation (1)
- Relocating the shop. This changes the C_c component and also changes the C_r component of ALL of the potential customers.

Sellers have a number of factors under their control, yet all of them affect both their profit per transaction as well as the perceived attractiveness by customers. Maximizing the profit is therefore a non trivial and non-linear task. Since many of the factors in the equations are unknown (especially in respect to clients' personal preferences), the sellers in real word often probe the market by adjusting their offer parameters and getting the feedback in terms of transactions number increase or decrease.

With the adoption of Internet and on-line transactions, the seller-buyer relation has been altered. Until then, the location of the shop was one of the most important factor, as long distance travel to make a purchase was in most cases unjustified. Thus the competition was limited. At the same time premium location rent was a major cost for the sellers. Internet sales affect the C_s and C_c components of the equation (2) but also the C_r part of the client's equation (1). The later is affected in two ways. First, the cost of a trip to the shop is flattened for all suppliers to the level of parcel's delivery cost. On the other hand, a cost of dissatisfaction is added as the customer must wait for the transaction to complete.

Finally, there is a social and knowledge context of each client's decision. Every client can only choose to buy from a seller he or she is aware of. Even if a client has an extremely cheap and high quality shop just 'around the corner', which in every aspect is best for him, he can only go there if he is aware of its presence. Therefore the clients' knowledge about available sellers and their offer heavily influences their decisions. Also, the social groups a client belongs to may additionally favor or disapprove certain products or sellers.

Considering all the factors involved, its understandable that with many clients and sellers in the market, their personal preferences and locations, even in a completely free market environment, there will be no single optimal price that the whole market will adopt. This is known as a price dispersion and is a well known effect in a free market [1], [6], with good mathematical analysis performed in [2]. Traditional analysis of the price dispersion leads to a conclusion that in most cases the the distribution of prices is Gaussian-like [6]. However, when Internet trade is considered, especially if substitutes are taken into consideration, the distribution becomes more asymmetrical (see Fig. 1) as has been demonstrated in [7].

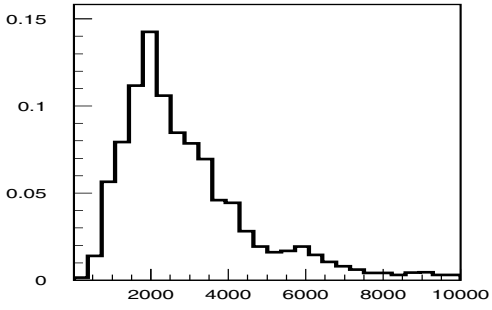


Fig. 1. Example of price dispersion for on-line shops, data gathered from price-comparison sites by authors. The *X axis* represents price offered for tv-sets by on-line shops, while the *Y axis* represent the estimated relative number of transactions. Methodology for the estimation is presented in [7].

3 Simulation Model

The simulation environment used for simulating clients' behavior and price distribution consists of two distinguishable types of agents [4] interacting with each other. Their initial placement in the defined space is random, as well as their initial properties.

3.1 The Vendors

A *Vendor* is represented be a quadruple:

$$V\{L, S, Z, Q\} \ . \tag{3}$$

Where:

- L is the location of the vendor
- S is the price of offered product
- Z is the product offered (represented by a certain abstract quality value)
- Q_s is the quality of the service offered by the vendor

Vendors are focused on one single goal - to maximize their profit generated for each defined period of time (counted in simulation cycles). To achieve this goal, they keep history of profit and price changes made during the recent periods - and adjust their offered price S. The amount of price change is random but limited in value, the direction of price change is however determined by the outcome of previous decisions. If decrease (or increase) in price resulted in increased profit (calculated as per equation (2)) for the evaluation period, the decrease (increase) will be continued. If however, the decrease lowered the profit, the offered price will be increased.

3.2 The Clients

Similarly, a model of *Client* consists of a number of properties:

$$C\{L, K, R, A, E\} \ . \tag{4}$$

Where:

- L is the location of the client
- K is the knowledge possessed by the client
- R is the relationships they have with other clients
- A is their current assets (owned items)
- E is their personality

For the purpose of identifying properties thorough this paper a convention known from object-oriented programming will be used. A hierarchical access to class properties is represented by a '.' (dot) - so $C.E$ means the personality of some client C, while $C.E.F$ means certain aspect of this personality.

Some of these properties may require further explanation:

Knowledge Possessed by a Client. This is limited to knowledge about available vendors, with each piece of knowledge consisting of: vendor location ($K.L$), their product and its price ($K.Z$, $K.S$), and (optionally) the service quality ($K.Q_s$). Each piece of knowledge has certain strength of memory imprint ($K.B$) and can fade with passing time.

$$K = \{V, L, S, Z, Q_s, B\} \ . \tag{5}$$

Relationships with Other Clients. Relationships are of type 'acquaintance' (i.e. certain client knows another client). A client can form new relationships as well as break existing ones (see "Agents' interactions").

Personality of a Client. It consists of several factors that affect each clients behavior. These are: friendliness ($E.F$), price sensitivity ($E.S$) and product adoption ($E.Z$) that denotes how quickly a client gets bored of a newly bought product and starts looking for a new one.

$$E = \{F, S, Z\} \ . \tag{6}$$

3.3 Clients' Actions and Interactions

As it has been described earlier in this section, vendors' behavior is limited to adjusting the offer price to maximize the profit generated. The clients on the other hand have much wider variety of actions available:

Relation Evolvement. With each simulation cycle, each client has a chance to 'meet' another client and develop a relation. A probability of meeting (P_m) depends on the euclidean distance ($d(C_1, C_2)$) and its importance ($X.R_d$) between the clients. $X.R_c$ is a constant simulation factor (thorough the rest of the article, simulation parameters will be prefixed with 'X'):

$$P_m = \frac{X.R_c}{1 + X.R_d * d(C_1, C_2)} \ . \tag{7}$$

If two clients meet, the chance of developing a relation (P_r) depends on the friendliness of each one ($C_n.E.F$) and their current number of friends ($C_n.count(R)$)

$$P_r = \frac{X.R_e * (C_1.E.F + C_2.E.F)}{(1 + X.R_f\,(C_1.count(R))) * (1 + X.R_f\,(C_2.count(R)))} \ . \tag{8}$$

A newly created relation has an initial default strength of 30%, although this value will be subject to change as simulation progresses. This is because with each simulation cycle every relation is tested against the condition:

$$X.R_d * rand() < \frac{C_1.E.F + C_2.E.F}{(1 + X.R_f\,(C_1.count(R))) * (1 + X.R_f\,(C_2.count(R)))} \ . \tag{9}$$

If the outcome is positive, the relation between the clients strengthen, if its negative it weakens. If it falls below a set threshold (i.e. 20%), the relationship is destroyed. This is to simulate real world scenario where people have many relations, with the number of their friends depending on their 'friendliness', yet its hard to maintain too many of them at the same time. The parameter $X.R_f$ denotes the difficulty of maintaining many relations at the same time - and its impact will be considered later on in this paper.

Knowledge Acquisition and Maintenance. There are two methods available for clients to acquire information about vendors and their offers. One is equivalent to marketing/broadcast of information by vendors: a client may came across a marketing message in their living space. A chance of remembering such information (Pm) is reciprocally proportional to the number of vendors the client already knows ($count(C.K)$) to emulate a limited capacity of their memory. The information acquired this way is: $K.V$, $K.L$, $K.S$, $K.Z$. It is assumed that the client does not learn about the quality of vendor's service this way.

The other method is used when a client comes to a decision of buying a new product. In such case they poll their acquaintances for information about their preferences. The client gets from each other client they have relationship with information about the vendor they bought their last products from. In this case the client also gets information about $K.Q_s$ experienced by the other client.

The information about vendors is stored in client's 'memory' which is subject to changes. With every simulation cycle, the $K.B$ property of each information piece is decreased. If it falls below a set threshold, this certain piece of information K is removed from client's 'memory' $K.B$ is however increased by each

received 'marketing message' and by information from client's acquaintances. The probability parameters of these actions are adjusted in such a way that on average a client has knowledge of a dozen or so vendors (although numbers for individual clients may vary significantly).

Buying Decision. At the moment of purchase it is assumed the client is completely satisfied with the product. Yet, with time the satisfaction decreases. The rate of decrease depends on their $E.Z$ value. Once it reaches low enough level (e.g. 20%), the client starts looking for a new product. This includes pooling its acquaintances for information and evaluating vendors' offers. The client also sets their expectations for the quality of the product and the price by examining the products owned by the acquaintances. Each of the analyzed offer attractiveness is calculated as follows:

$$A_n = G_s(C_{ES} - V.S) + G_q(V.Z - C_{EZ}) - X.E_d * d(C, V) - X.E_s * V.Q_s \ . \quad (10)$$

Where:

- A_n is the attractiveness of the offer of vendor n
- G_s is client's price sensitivity (normalized value based on $C.E.S$)
- G_q is client's quality sensitivity which is to be negatively correlated with G_s
- C_{ES}, C_{EZ} are client's expected price and product quality respectively
- $V.S$, $V.Z$, $V.Q_s$ are the properties of vendor's offer
- $X.E_d$, $X.E_s$ are adjusted simulation parameters
- $d(C, V)$ is the distance between the client and the vendor

After assessing the offers of known vendors, the client chooses to buy a certain product - only if the attractiveness is higher than the current satisfaction with the current product. If none of the offers is more attractive than current product's satisfaction, the client does not buy a new product, but decreases the satisfaction of the current one with each new cycle. The satisfaction decreases linearly and can go into negative values, thus guaranteeing that ultimately the customer will choose one of the available offers.

3.4 Simulation Assumptions

There is a number of assumptions and fixed parameters to the model that influence its performance. Most assumptions are simplifications, introduced to allow modeling of complex human behavior with limited number of variables. Some of the parameters are set according to the best knowledge of the authors, as it is often difficult to find a scientifically proven data to back them up. The most important assumptions are:

- There is only one type of product traded. Although the products offered by various vendors may differ in both technical and perceived values (e.g. various photographic cameras), they are all economical substitutes [3] of each other.

- The satisfaction from using a product is linearly decreasing in time (although the ratio of decreasing depends on client's personality).
- All clients perceive quality of products and quality of vendors' service the same way.
- All information is passed between agents error-free.
- The broadcast of information from all vendors is statistically equal, apart from the influence of the distance to the client.
- The distribution of client's personalities is statistically equal. The same applies to clients' and vendors' locations and initial vendors' offers.

Vital model parameters:

- The number of information items about vendors and their offers is limited and on average will reach about 20.
- The number of relationships a client can maintain is limited and depends on client's personality, but in reality will not exceed 20-30.

4 Simulation Results

All the results presented here have been taken for a simulation with the following parameters:

- Number of clients: 2000, number of vendors: 100
- Simulation length: 500 cycles (allowing on average for 10 adjustments of vendors' prices)

A typical result of such simulation can bee seen in Fig. 2. Notably, its quite similar to the real-world results (Fig. 1).

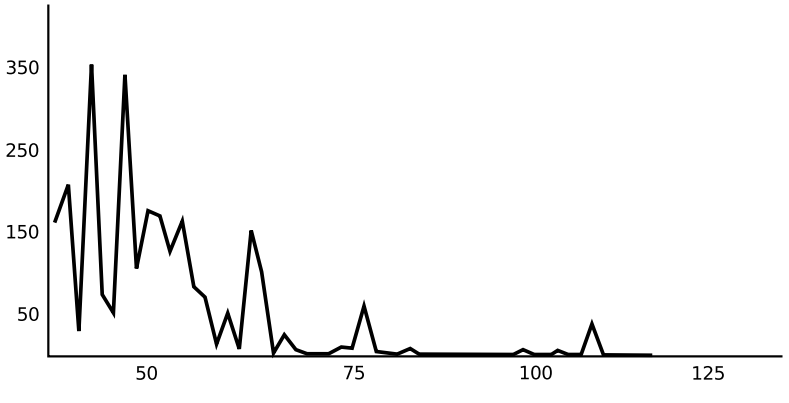


Fig. 2. Price distribution with default parameters. The *Y axis* represents the number of transactions, while the *X axis* represents the transactional price.

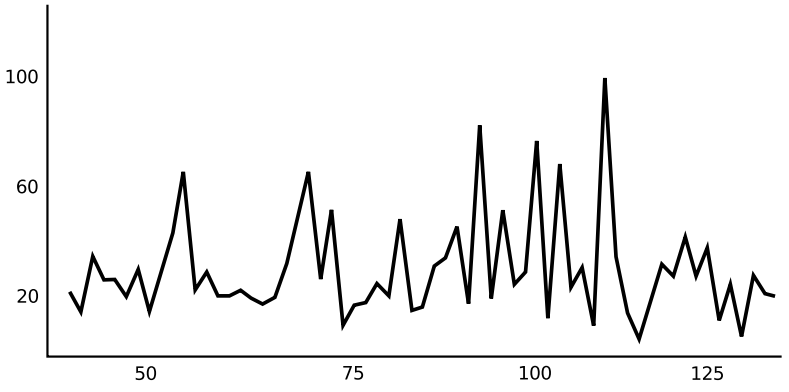


Fig. 3. Price distribution in the 'no relationship' scenario. The *Y axis* represents the number of transactions, while the *X axis* represents the transactional price.

To verify the impact of relations between clients, Fig. 3 demonstrates a price dispersion in an environment with no relationships and no information transfer between clients. As can be expected, the distribution is much more even - reflecting the even randomization of clients' personalities.

Another interesting results can be obtained in case when distances between clients (also distances between clients and vendors) do not matter. This can be considered as a possible 'all-internet' future when all the contacts between people are done remotely. The results of such simulation are presented in Fig. 4. Interestingly, there are only a few spikes in the chart. This is a result of a clusterisation of the clients base - with no distance impact, they tend to form closed groups that average their preferences for product quality. They all then start to buy from the vendor that offers the best deal for the required quality.

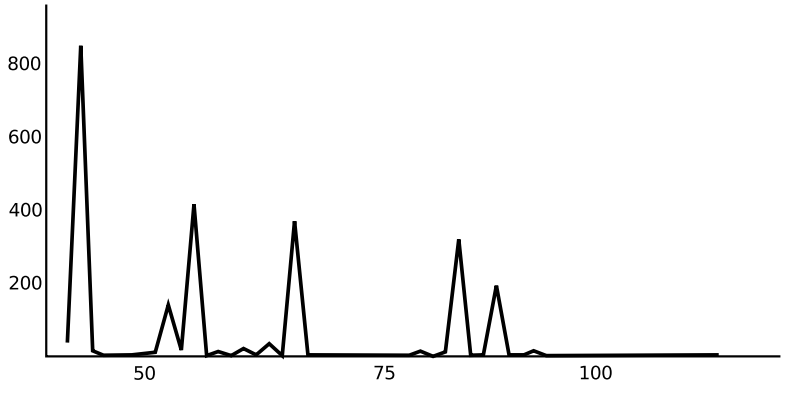


Fig. 4. Price distribution in the 'distances do not matter' scenario. The *Y axis* represents the number of transactions, while the *X axis* represents the transactional price.

5 Conclusions

The results of the experiments, that match real-world observations confirm that the initial assumptions are likely to be correct. Its especially intriguing that removing even one of the key environmental conditions (either clients' relations or the impact of distances) causes the result to significantly differ.

Another interesting observations are the results of the 'no distance world' scenario (Fig. 4). The formation of several clusters with common preferences may be an indication of what can be expected in the future when the generation that uses the Internet as their primary communication tool becomes predominant. It may actually mean that not an increased, but decreased diversity of available experiences might be expected. The effect is so interesting that it prompts further research (although the model will most likely need to be extended for that purpose). Some indication that this scenario may indeed be true is the success of enterprises like Facebook or Groupon - which won their place due to the number of users they attract -which in turn attracts more users. Such self-forming and quickly growing clusters are indeed worth of research attention.

Finally, there is another area that could be investigated with the simulation model presented - and that is an impact of different forms of marketing on the sales. Although it is not possible to come up with an analytical answer on the marketing effectiveness, it should be possible to analyze relative impact of vendor's action in respect to others.

References

1. Baye, M.R., Morgan, J., Scholten, P.: Price Dispersion in the Small and in the Large: Evidence from an Internet Price Comparison Site. Journal of Industrial Economics 52, 463–496 (2004)
2. Baye, M.R., Morgan, J., Scholten, P.: Information, Search, and Price Dispersion. In: Hendershott, T. (ed.) Handbook on Economics and Information Systems, pp. 323–375. Elsevier, New York (2006)
3. Begg, D., Begg, D.K.H., Fischer, S., Dornbusch, R.: Economics. McGraw-Hill, New York (2005) ISBN : 9780077107758
4. Lin, H.: Architectural Design of Multi-Agents System Information Science Reference (2007) ISBN 978-1-59904-108-7
5. Polterovich, V.: Rationing, Queues, and Black Markets. Econometrica 61, 1–28 (1993)
6. Roberts, M.J., Supina, D.: Output Price and Markup Dispersion in Micro Data: The Roles of Producer Heterogenity and Noise. In: Baye, M. (ed.) Advances in Applied Microeconomics. JAI Press, Greenwich (2000) ISBN: 0-7623-0687-4
7. Zachara, M., Piskor-Ignatowicz, C.: Analysis of product price distribution and pricing policy of online merchants. In: Materiały Krakowskiej Konferencji Młodych Uczonych, pp. 659–665, Kraków (2010) ISBN: 978-83-62079-04-9

Evolutionary Sets of Safe Ship Trajectories: Problem-Dedicated Operators

Rafał Szłapczyński[1] and Joanna Szłapczyńska[2]

[1] Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
[2] Gdynia Maritime University, Morska 81-87, 81-225 Gdańsk, Poland
rafal@pg.gda.pl, asiasz@am.gdynia.pl

Abstract. The paper presents the optimization process of the evolutionary sets of safe ship trajectories method, with a focus on its problem-dedicated operators. The method utilizes a customized evolutionary algorithm to solve a constrained optimization problem. This problem is defined as finding a set of cooperating trajectories (a set is an evolutionary individual) of all the ships involved in the encounter situation. The resulting trajectories are safe (meeting optimization constraints) while minimizing the average way loss ratio. When developing a new version of the method, the authors decided to introduce a number of changes. This upgrade enforced redesigning of the optimization process, especially the problem-dedicated collision-avoidance operators.

Keywords: Evolutionary computing, multi-ship encounter situation.

1 Description of the Problem

The main approaches to the problem of planning optimal ship trajectories in encounter situations are based on either differential games [1] or on evolutionary programming [2]. The authors have proposed a new approach, which combines some of the advantages of both methods: the low computational time, supporting all domain models and handling stationary obstacles, typical for evolutionary method [3, 4, 5]), with taking into account the changes of motion parameters (changing strategies of the players involved in a differential game). Instead of finding the optimal own trajectory, an optimal set of safe trajectories of all the ships involved is searched for. The search is performed in real time and assumes most probable behavior of all ships. The method is called evolutionary sets of safe trajectories [6]. It assumes that we are given the following data:

- stationary constraints (such as landmasses and other obstacles),
- positions, courses and speeds of all the ships involved,
- ship domains (a ship domain is an area around the ship that should be free from other ships and obstacles during the voyage),
- times necessary for accepting and executing the proposed manoeuvres.

The goal is to find a set of trajectories, which minimizes the average way loss spent on maneuvering, while fulfilling the following conditions:

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 221–230, 2011.

- none of the stationary constraints are violated,
- none of the ship domains are violated,
- the minimal acceptable course alteration is not lesser than 15 degrees,
- the maximal acceptable course alteration is not to be larger than 60 degrees,
- speed alterations are not to be applied unless necessary (collision cannot be avoided by course alteration up to 60 degrees),
- a ship maneuvers, if and only if, she is obliged to by International Regulations for Preventing Collisions at Sea (COLREGS) [7],
- maneuvers to starboard are favored over maneuvers to port board.

The paper presents a description of the specialized operators and mutation used in the method. In the next section the two phases of the evolutionary process are discussed. Simulation experiments and an example of method's application are presented in Section 3. Finally the summary and conclusions are given in Section 4.

2 Evolutionary Sets of Safe Ship Trajectories

The method solves the constrained optimization task as presented in the previous section and thus provides a solution to a multi-ship encounter situation. The solution is a set of optimal ship trajectories: minimizing the average way loss (goal function) and meeting the optimization constraints such as avoiding land obstacles and other ship domains, fulfilling COLREGS rules, etc. An individual in the method is a set of trajectories, each trajectory for one of the ships involved in an encounter. A trajectory is a sequence of nodes (waypoints), each node containing the data:

- geographical coordinates x and y,
- the speed between the current and the next node.

The single-objective normalized fitness function of the method is given by the formula:

$$fitness = \sum_{i=1}^{n} \frac{trajectory\_fitness_i}{n}, \tag{1}$$

where:

$$trajectory\_fitness_i = trajectory\_economy\_factor_i * scf_i * caf_i * ccf_i. \tag{2}$$

The *trajectory_fitness* takes into account a way loss as a basic economy factor (*trajectory_economy_factor*) and penalties for static constraints violation (*scf*), ship-to-ship collisions (*caf*) and COLREGS violations (*ccf*) as given by formulas (3)-(6).

$$trajectory\_economy\_factor_i = \left(\frac{trajectory\_length_i - way\_loss_i}{trajectory\_length_i} \right), \tag{3}$$

$$scf_i = \left(\frac{trajectory\_length_i - trajectory\_cross\_length_i}{trajectory\_length_i} \right)^2, \tag{4}$$

$$caf_i = \prod_{j=1, j\neq i}^{n} \left(\min(fmin_{i,j}, 1)\right), \tag{5}$$

$$ccf_i = 1 - \sum_{k=1}^{m} \left[COLREGS\_violation\_penalty_k\right], \tag{6}$$

where:

i – the index of the current ship [/],
j – the index of a target ship [/],
k – the index of a registered violation [/],
m – the number of COLREGS violation registered for the current ship[/],
n – the number of ships [/],
$COLREGS\_violation\_penalty_k$ – the penalty for the k-th of the registered COLREGS [7] violation [/],
$fmin_{i,j}$ – the approach factor value [8] for an encounter of ships i and j, if i-th ship is the privileged one, the potential collision is ignored and the approach factor value is equal to "1" by definition [/],
$trajectory\_cross\_length_i$ – the total length of the parts of the i-th ship's trajectory, which violate stationary constraints [nautical miles],
$trajectory\_length_i$ – the total length of the i-th ship's trajectory [nautical miles],
way_loss_i – the total way loss of the i-th ship's trajectory [nautical miles] computed as a difference between the trajectory length and length of a segment joining trajectory's start point and endpoint.

The proposed method handles the problem of ship encounters by focusing on specialized operators. During the method's design process the authors have decided to differ between problem-dedicated specialized operators (subsection 2.1) and typical random mutation (subsection 2.2). The specialized operators, some of them being semi-deterministic [9], are responsible for more conscious improving of trajectories and thus can result in a faster convergence to a solution. As opposed to mixed mutation approach, favored by some researchers [10], the authors make the choice of a particular specialized operator dependent on the present circumstances instead of past experiences. Apart from the specialized operators and mutation the method includes also problem-customized crossover operators which are described in an accompanying paper [11].

Specialized operators and mutation, both use the information returned by evaluation. This information includes fitness function values as well as the data on detected collisions and groundings. Therefore evaluation has to be performed directly preceding the specialized operations and mutation. This resulted in a modified evolutionary algorithm scheme. However, since the evaluation requires collision detection, it is the most time consuming phase. In the modified evolutionary algorithm it would be performed twice: once after reproduction and again after mutation. Therefore doubling the evaluation phase in a cycle increases the total computational time approximately 1.5 times (the extra evaluation after reproduction is done for a population half the size of the one after mutation). To shorten the process,

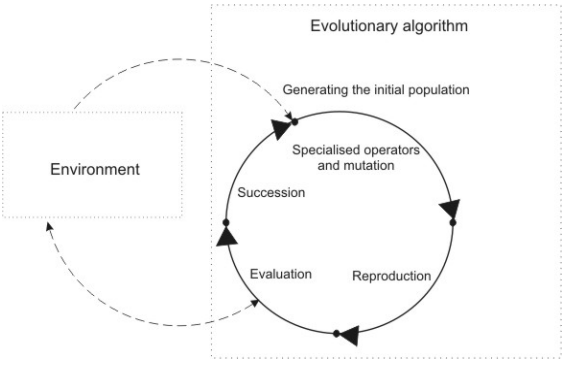


Fig. 1. Modified evolutionary algorithm applied to the method

the authors have decided to change the order of operations within the algorithm. The reproduction phase and specialized operations / mutation phase have changed places with each other, which is shown in Figure 1.

2.1 Specialized Operators

The evolutionary operators used here, can be divided into three groups, with group 1 only applied for restricted waters (waters with presence of static constraints eg. landmasses and other navigational obstacles). On restricted waters, the order of applying collision avoidance operators for collisions is such that operators handling violations of stationary constraints precede operators handling violations of other ships' domains. A violation of stationary constraint must always be handled, since it is disastrous, regardless of other ships' behavior. However, violation of other ships' domains may be no longer valid after violations of stationary constraints have been handled, because the operators responsible for avoiding violations of stationary constraints may have changed the trajectories in such a way that ship-to-ship collisions would not occur. The following operators have been used:

1. Operators avoiding collisions with stationary obstacles (restricted waters only, Figure 2). If a segment of a trajectory crosses a stationary obstacle, depending on the values of a time remaining to collision and a time remaining to reaching the next node, one of the above mentioned five operators is chosen:
    a) segment insertion – if only there is enough time for three course alterations, a new segment is inserted,
    b) node insertion – if there is not enough time for a whole new segment (additional three course alterations), a single node is inserted,
    c) first node shift – if there is not enough time for a node insertion (additional two course alterations) and the collision point is much closer to the first node of a segment, the first node is moved away from the collision point,
    d) second node shift – if there is not enough time for a node insertion (additional two course alterations) and the collision point is much closer to the second node of a segment, the second node is moved away from the collision point,

e) segment shift – if there is not enough time for a node insertion (additional two course alterations) and the collision point is close to the middle of a segment, the whole segment is moved away from the collision point.

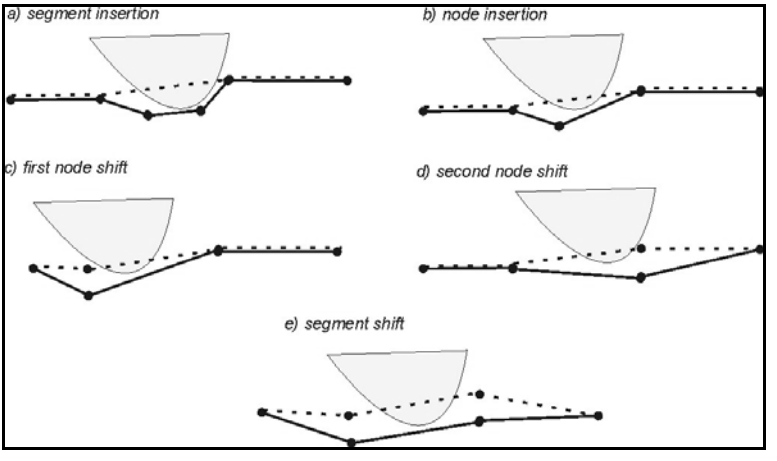


Fig. 2. Specialized operators: avoiding collisions with stationary obstacles

2. Operators avoiding collisions with prioritized ships (Figure 3). If a collision with a prioritized ship is registered, one of those operators is selected depending on the values of a time remaining to a collision and a time remaining to reaching the next node, similarly as for group 1.

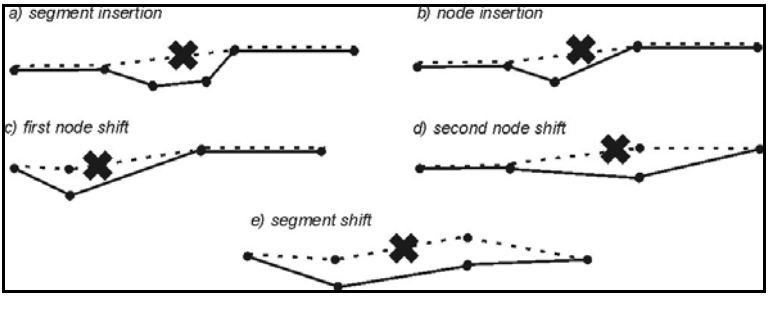


Fig. 3. Specialized operators: avoiding collisions with prioritized ships

3. Validations and fixing. This group includes three operators, shown in Figure 4.
   a) node reduction – its purpose is to eliminate all the unnecessary nodes - if a segment, which bypasses a given node by joining its neighbors, is safe, the node is deleted,
   b) smoothing – if a course alteration is larger than 30 degrees, the node is replaced with a segment to smoothen the trajectory,

c) adjusting maneuvers – each trajectory is analyzed and in case of unacceptable maneuvers (such as course alterations below 15 degrees), the nodes are moved so as to round a maneuver to an acceptable value.

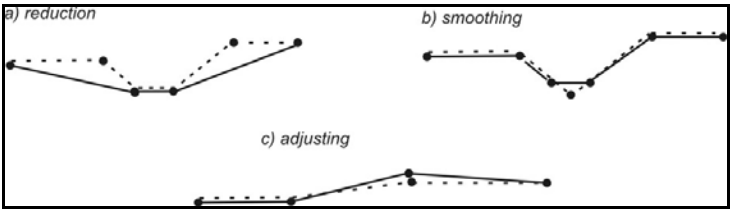


Fig. 4. Validations and fixing operators

The authors have decided, that all the above listed operators would be used whenever needed (fixing probability parameter set to 1), as opposed to mutation. There is no risk of spoiling a high valued individual because overlapping populations are used and the specialized operators work on individuals' copies.

2.2 Mutation

Four types of random mutation operators (Figure 5) have been used, all operating on single trajectories. These random operators are:

a) node insertion: a node is inserted randomly into the trajectory,
b) node joining: two neighboring nodes are joined,
c) node shift: a randomly chosen node is moved (polar coordinates changed),
d) node deletion: a randomly selected node is deleted.

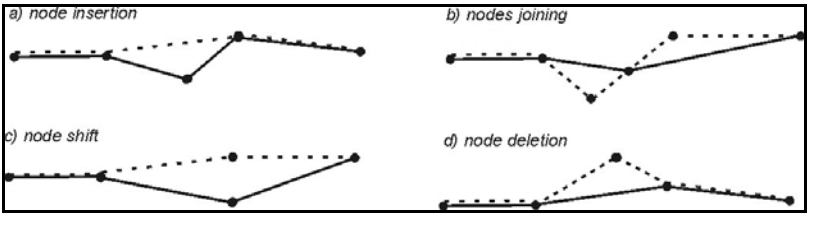


Fig. 5. Random mutation operators

A trajectory mutation probability decreases with the increase of the trajectory fitness value, so as to mutate the worst trajectories of each individual first. In the early phase of the evolution all random operators: the node insertion, deletion, joining and shift are equally probable. In the later phase node shift dominates with its course alteration changes and distance changes decreasing with the number of generations. For node insertion and node shift instead of Cartesian coordinates x and y, the polar coordinates (course alteration and distance) are mutated in such a way that the new maneuvers are between 15 and 60 degrees. The mutation operations are applied to an

individual's copy, but only, if no specialized operator has been used for this individual. The probability of applying mutation operations for each trajectory is:

$$mutation\_probability = basic\_mutation\_probability * (1 - trajectory\_fitness) \quad (7)$$

where:
basic_mutation_probability is a parameter from range [0.05, 0.2], thus larger than typically in genetic algorithms, where mutation is usually secondary to crossover [12].

The mutation probability [13, 14] was made dependent on the fitness function value, as opposed to uniform or dependant on generation number, to only slightly shift the best trajectories in a set, while strongly shake the weaker ones.

3 Simulation Experiments

Specialized, problem-dedicated operators are only useful if they considerably improve ship trajectories. It has already been stated that evolutionary scheme had to be changed, to avoid doubling the evaluation phase. Thus a question arises: are these operators more effective than basic random mutation and worth the changes in the evolutionary algorithm? Also, even without additional evaluation, an iteration of the cycle with the operators takes approximately 1.05 of the time spent on the same iteration without them. Therefore it is worth checking which option returns better solutions. Such comparison is presented below. The numbers of generations have been set to such values that the total computational time is the same. The 91 randomly generated test scenarios cover open and restricted waters, all typical combinations of courses with the number of ships ranging from 2 to 6.

3.1 The Method with and without the Operators: Statistical Results

The Table 1 presents test parameters for the method's with and without specialized operators. The test results are given in Table 2.

Table 1. General parameters for the method's test

Test scenarios	Runs for each scenario	Generations	Population size	Basic mutation probability	Fixing probability
91	5	100/105, 200/210	100	0.05	1.00

Table 2. Average fitness function values for various versions of the method

The method	Number of generations	Average fitness function
Method with all operators	100	0.9725
	200	0.9741
Method without specialized operators (basic mutation only)	105	0.9585
	210	0.9618

The method with specialized operators is superior to the one without them. Even with the number of generations doubled, the version with only the mutation cannot reach the same results (Table 2). The superior version returns much better results for multiple encounters on restricted waters, as shown in the next section.

3.2 The Method with and without the Operators: An Example of Results

Below a randomly generated scenario is presented. The motion parameters of the ships are given by Figure 6. Figures 7 and 8 present the results of running the method with all operators and without the specialized operators respectively.

Zero longitude 20.5 Zero latitude 58.5

Ship Parameters

	Initial pos	Goal pos	Velocity [kn]	Turn [deg./sec.]
Ship1	(80.01 ; 160.45)	(103.99 ; 143.55)	14.67	1.00
Ship2	(98.95 ; 163.27)	(90.73 ; 143.08)	10.90	1.00
Ship3	(103.88 ; 157.78)	(85.75 ; 148.55)	10.17	1.00
Ship4	(102.23 ; 159.43)	(84.10 ; 150.20)	10.17	1.00
Ship5	(83.85 ; 150.95)	(105.86 ; 155.42)	11.23	1.00
Ship6	(82.16 ; 152.57)	(104.16 ; 157.04)	11.23	1.00

Fig. 6. Parameters of six ships involved in an encounter

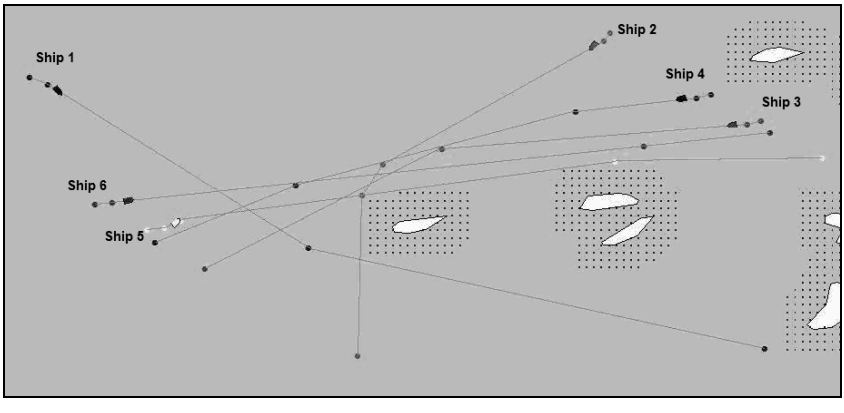


Fig. 7. Result trajectories for the six ships generated by the method with all operators, running for 100 generations

According to COLREGS the ships should act as follows:

- Ship 1 should give way to ships 5 and 6,
- Ship 2 should give way to ships 1, 5 and 6,
- Ship 3 should give way to ships 1, 2 ,4, 5 and 6,
- Ship 4 should give way to ships 2, 5 and 6,

- Ship 5 is a privileged ship (given way to by all other ships),
- Ship 6 should give way to ship 5.

In the solutions shown in Figures 7 & 8 all ships perform the maneuvers recommended by COLREGS passing each other and all obstacles in safe distances. The resulting fitness value for the scenario with all operators applied is 0.9739, while for the one without the operators is 0.9236. The larger fitness function value for Figure 7 (all operators applied) results from the lesser value of total way loss. Especially, the privileged ship 5 takes a shorter trajectory in Figure 7 than in Figure 8.

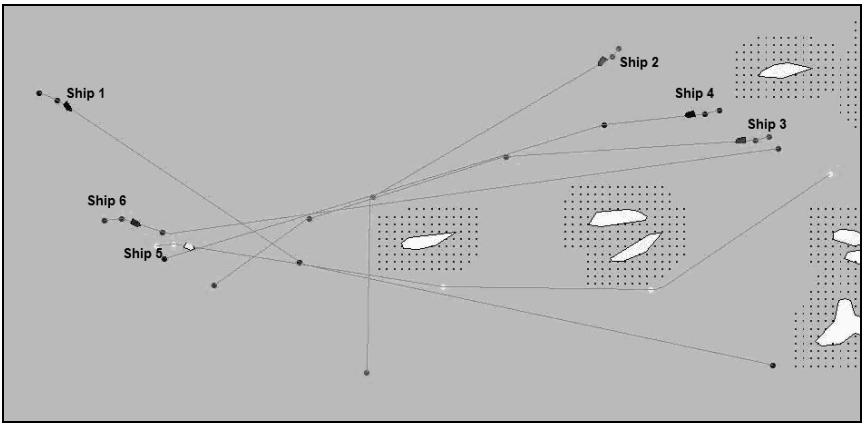


Fig. 8. Result trajectories for the six ships generated by the method without the operators (mutation only), running for 105 generations

4 Summary and Conclusions

In the paper a method of solving ship encounter situations has been discussed, with the focus on its specialized operators. The method is a generalization of evolutionary trajectory determining. A set of trajectories of all ships involved, instead of just the own trajectory, is determined. The method avoids violating ship domains and stationary constraints, while obeying the COLREGS and minimizing average way loss computed over all trajectories. It benefits from a set of problem-dedicated operators, all of which have been described here. Using these operators enforced the changes of the traditional evolutionary cycle. The final version of the method has been compared with the basic one by means of a series of simulations. The results prove superiority of the extended version over the one devoid of specialized operators, even if the latter is run for a much larger number of generations. Other aspects of the method have been described in the accompanying paper [11]. The further research of the authors is focused on additional evaluation rules and operators, which would allow directly applying the method to the Traffic Separation Scheme areas.

Acknowledgments. The author thanks the Polish Ministry of Science and Higher Education for funding this research under grant no. N N516 186737.

References

1. Lisowski, J.: Dynamic games methods in navigator decision support system for safety navigation. In: Proceedings of the European Safety and Reliability Conference, vol. 2, pp. 1285–1292 (2005)
2. Smierzchalski, R., Michalewicz, Z.: Modeling of a Ship Trajectory in Collision Situations at Sea by Evolutionary Algorithm. IEEE Transactions on Evolutionary Computation 4(3), 227–241 (2000)
3. Beyera, H.G., Schwefela, H.P., Wegenerb, I.: How to analyse evolutionary algorithms. Theoretical Computer Science 287, 101–130 (2002)
4. Eiben, A.E., Schoenauer, M.: Evolutionary computing. Information Processing Letters 82, 1–6 (2002)
5. Michalewicz, Z., Fogel, D.B.: How To Solve It: Modern Heuristics. Springer, Heidelberg (2004)
6. Szlapczynski, R.: Solving multi-ship encounter situations by evolutionary sets of cooperating trajectories. In: Marine Navigation and Safety of Sea Transportation, pp. 437–442. CRC Press / Taylor & Francis Group / Balkema (2009)
7. Cockroft, A.N., Lameijer, J.N.F.: A Guide to Collision Avoidance Rules. Butterworth-Heinemann Ltd., Butterworths (2003)
8. Szłapczyński, R.: A unified measure of collision risk derived from the concept of a ship domain. Journal of Navigation 59(3), 477–490 (2006)
9. Jozwiak, L., Postula, A.: Genetic engineering versus natural evolution: genetic algorithms with deterministic operators. Journal of Systems Architecture 48, 99–112 (2002)
10. Dong, H., He, J., Huang, H., Hou, W.: Evolutionary programming using a mixed mutation strategy. Information Sciences 177, 312–327 (2007)
11. Szlapczynski, R.: Evolutionary Sets of Safe Ship Trajectories: improving the method by adjusting evolutionary techniques and parameters, submitted to ICCCT (2011) (in prep.)
12. De Falco, I., Della Cioppa, A., Tarantino, E.: Mutation-based genetic algorithm: performance evaluation. Applied Soft Computing 1, 285–299 (2002)
13. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. Applied Mathematics and Computation 193, 211–230 (2007)
14. Zhao, X., Gao, X.S., Hu, Z.C.: Evolutionary programming based on non-uniform mutation. Applied Mathematics and Computation 192, 1–11 (2007)

Evolutionary Sets of Safe Ship Trajectories: Improving the Method by Adjusting Evolutionary Techniques and Parameters

Rafał Szłapczyński

Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
rafal@pg.gda.pl

Abstract. The paper presents some of the evolutionary techniques used by the evolutionary sets of safe ship trajectories method. In general, this method utilizes a customized evolutionary algorithm to solve a constrained optimization problem. This problem is defined as finding a set of cooperating trajectories (here the set is an evolutionary individual) of all the ships involved in the encounter situation. The resulting trajectories are safe, taking into account the International Regulations for Preventing Collisions at Sea (COLREGS), and economical - due to minimization of the average way loss ratio (the goal function). While developing a new version of the method, the author decided to introduce a number of changes, e.g. focusing on COLREGS compliance. The upgrade to the method led the author to experiments with various evolutionary mechanisms which resulted in a much more effective evolutionary process. These mechanisms are thoroughly discussed here.

Keywords: Evolutionary computing, multi-ship encounter situation.

1 Introduction

A desired solution to a multi-ship encounter situation would include a set of planned, optimal trajectories for all the ships involved in an encounter. These trajectories should have properties of no collision or domain violations when the ships follow them. There is a number of approaches to solving this problem. Two basic trends are utilization of differential games [1] and evolutionary computing [2]. The evolutionary method has also been applied by other researchers for finding an optimal path [3] as well as optimal collision avoidance maneuver [4]. The author has proposed a new approach, which combines some of the advantages of both methods: the low computational time and flexibility (typical for evolutionary methods [5, 6, 7]), with taking into account the changes of motion parameters (changing strategies of the players involved in a game). Instead of finding the optimal own trajectory for the unchanged courses and speeds of targets, an optimal set of safe trajectories of all the ships involved is here searched for. The method is called evolutionary sets of safe trajectories [8].

The paper presents a description and a discussion of the evolutionary techniques used in the current version of the method, with a focus on improving some of the

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 231–240, 2011.

evolution's phases. The rest of the paper is organized as follows. In the next section the optimization problem and constraint violations detection are described. Then selected evolutionary phases are presented: reproduction (Section 3), evaluation (Section 4) and selection (Section 5). Example results are presented in Section 6 and finally the summary and conclusions are given in Section 7.

2 Optimization Problem and Constraint Violations Detection

It is assumed that we are given the following data:

- stationary constraints (such as landmasses and other obstacles),
- positions, courses and speeds of all ships involved,
- ship domains (area around the ship that should be free from other objects),
- times necessary for accepting and executing the proposed maneuvers.

The goal is to find a set of trajectories, which minimizes the average way loss spent on maneuvering, while fulfilling the following conditions:

- none of the stationary constraints and ship domains are violated,
- the acceptable course alterations should be between 15 and 60 degrees,
- speed alteration are not to be applied unless necessary,
- a ship only maneuvers, when she is obliged to,
- maneuvers to starboard are favored over maneuvers to port board.

The conditions are imposed either by COLREGS [9] and good marine practice or by the economics. An additional, computational constraint is the fact, that due to the optimization being done in real time (with the ships approaching each other and the obstacles), the solution should be returned within a short time specified by the operator of the system. By default, one minute is assumed. A description of how the particular optimization constraints are handled in the proposed method is presented below.

The current version of the method utilizes a vector map of a given area, which is not processed directly, but used for generating bitmap of an area. Then, when the method is running, each bitmap cell, which the trajectory of a ship traverses, is read and checked for belonging to landmass, water or safety isobate. For a bitmap, whose detail level reflects this of a given vector map, the computational time is proportional to the number of traversed cells.

The algorithm for detecting ship-to-ship collisions is as follows. Each ship's trajectory is checked against all other ships. For each pair of ships, the start time and end time of each trajectory's segments are computed. If two segments of the two trajectories overlap in time, they are checked for geometrical crossing. In case of a crossing, the approach factor value, proposed in [10] by the author, is computed. Then, if the approach factor value indicates collision, the type of an encounter (head-on, crossing or overtaking) is determined on the basis of the ships' courses and it is

decided, which ship is to give way (both ships in case of head-on). The collision is only registered for the give way ship.

As for the COLREGS [9] violations, three types of those are of interest here:

- a ship does not give way, when it should,
- a ship gives way, when it should not, because it is a stand-on ship,
- a ship maneuvers to port-board when it should maneuver to starboard.

Each of these three situations may happen on either open or restricted waters. The COLREGS violations detection rules applied in the method are:

1. On open waters, COLREGS violation is registered if
   a) a ship, which is not obliged to give way to any other ship, maneuvers,
   b) a ship, which is obliged to give way, does not perform a maneuver,
   c) a ship maneuvers to port board.
2. On restricted waters: each maneuver has assigned additional information on its reason: land or other stationary obstacle avoidance, target avoidance or accidental maneuver generated by evolutionary mechanisms. Based on this, COLREGS violations are registered if:
   a) a ship, which does not initially have to give way to any target has its first maneuver's reason other than static constraint violation avoidance,
   b) a ship maneuvers to port board for reason other than static constraint violation avoidance.

3 Individuals and Their Reproduction

Each individual is a set of trajectories, each trajectory corresponding to one of the ships involved in an encounter. A trajectory is a sequence of nodes, each node containing the following data:

- geographical coordinates x and y,
- the speed between the current and the next node.

In the reproduction phase pairs of individuals (parents) are crossed to generate new individuals (offspring). Two types of crossover operators (Figure 1) have been used:

- an offspring inherits whole trajectories from both parents,
- each of the trajectories of the offspring is a crossover of the appropriate trajectories of the parents.

Due to problem specifics there is no guarantee that offspring of two highly valued parents will be highly valued itself. In case of trajectory inheriting (Figure 1b), the resulting trajectories may not fit to other trajectories (collisions between ships). Therefore, to make sure that the best individuals will not be lost (the parents might be better fitted than their offspring), the overlapping populations are used.

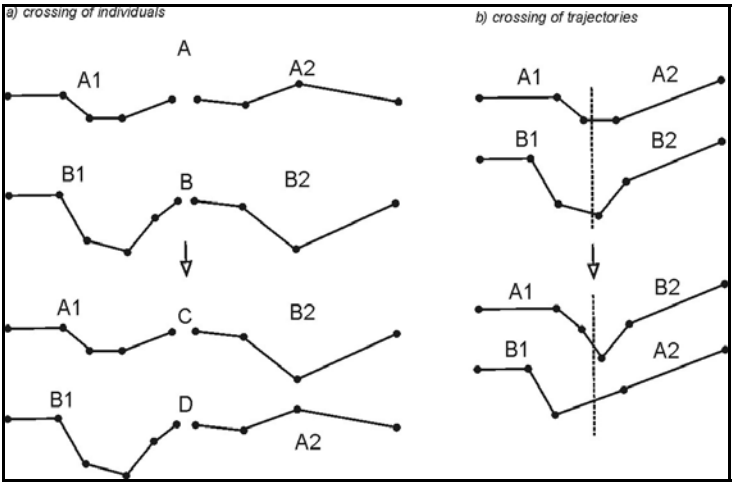


Fig. 1. Reproduction: inheriting whole trajectories (a) and crossover of trajectories (b)

4 Fitness Function

In the described method all individuals (sets of trajectories) are evaluated by the fitness function, reflecting given optimization criteria and constraints. The basic criterion is the economic one – minimizing way losses of trajectories in a set. For each of the trajectories, *a trajectory_economy_factor* is computed according to the formula (1).

$$trajectory\_economy\_factor_i = \left(\frac{trajectory\_length_i - way\_loss_i}{trajectory\_length_i} \right), \tag{1}$$

where:

i – the index of the current ship [/],
$trajectory\_length_i$ – the total length of the i-th ship's trajectory [nautical miles],
way_loss_i – the total way loss of the i-th ship's trajectory [nautical miles] computed as a difference between the trajectory length and length of a segment joining trajectory's start point and endpoint.

The *trajectory_economy_factor* is always a number from a (0,1] range.

After the *trajectory_economy_factor* has been computed the static constraints are handled by introducing penalties for violating them. For each trajectory its static constraint factor scf_i is computed. The static constraints are always valid and their violations must be avoided at all cost, therefore penalties applied here are the most severe – hence the square in the formula (2).

$$scf_i = \left(\frac{trajectory\_length_i - trajectory\_cross\_length_i}{trajectory\_length_i} \right)^2, \tag{2}$$

where:
$trajectory\_cross\_length_i$ – the total length of the parts of the i-th ship's trajectory, which violate stationary constraints [nautical miles].

Analogically to the static constraint factor, collision avoidance factor caf_i is computed to reflect the ship's collisions with all other privileged ships as shown by (3).

$$caf_i = \prod_{j=1, j\neq i}^{n} \left(\min(fmin_{i,j}, 1)\right), \tag{3}$$

where:

n – the number of ships [/],
j – the index of a target ship [/],
$fmin_{i,j}$ – the approach factor value [10] for an encounter of ships i and j, if i-th ship is the privileged one, the potential collision is ignored and the approach factor value is equal to "1" by definition [/].

The collision avoidance factor is a number from a [0,1] range, where "1" value means no ship domain violation and "0" means a crash with at least one of the targets.

The COLREGS violations are secondary to static constraint violations and to collisions with other ships. Therefore the author has decided to penalize it moderately, to make sure that constraints from the previous two points are met first. COLREGS compliance factor ccf_i is computed according to the formula (4).

$$ccf_i = 1 - \sum_{k=1}^{m} \left[COLREGS\_violation\_penalty_k\right], \tag{4}$$

where:

m – the number of COLREGS violation registered for the current ship as has been described in section 2 [/],
k – the index of a registered violation [/],
$COLREGS\_violation\_penalty_k$ – the penalty for the k-th of the registered COLREGS violation [/].

The penalty values for all registered COLREGS violations are configurable in the method and are set to 0.05 by default.

Once all aforementioned factors have been computed, the normalized fitness function value is calculated. The fitness function keeps the needed high resolution of evaluation: minor stationary constraints violations are penalized similarly as major collisions with other ships and minor collisions with other ships are penalized similarly as multiple COLREGS violations.

$$fitness = \sum_{i=1}^{n} \frac{trajectory\_fitness_i}{n}, \tag{5}$$

where:

$$trajectory\_fitness_i = trajectory\_economy\_factor_i * scf_i * caf_i * ccf_i. \tag{6}$$

It must be noted here, that while fitness function values are normalized, a single trajectory fitness function value may be equal to 1.0 only for a stand-on ship in lack of obstacles on his way. The global fitness function value of 1.0 is only possible when none of the ships manoeuvre, that is, when there are no encounters (situations, which are not considered here). The minimal assumed course alteration manoeuvre is 15 degrees, the minimal time for accepting and executing a manoeuvre – 6 minutes. Thus, for a ship, which was supposed to cover a distance of 12 nautical miles with a speed of 12 knots but performed one minimal course alteration manoeuvre, the trajectory's fitness function value would be approximately 0.98. For more complex scenarios the maximum possible value of fitness function computed over all trajectories would be even smaller. Therefore, while the precise value cannot be determined analytically, it is reasonable to assume that for a randomly generated multi-ship encounter situation the maximum possible value of fitness function would be below 0.98, which can be considered a better practical reference value than 1.0.

5 Selection

The specialized operators and mutation, which operate on the individuals prior to their evaluation and succession have been described in detail in the accompanying paper [11] and therefore this subject is not touched here. Thus the next phase of interest is selection, which is used separately for reproduction (pre-selection) and succession (post-selection) and purposes. The author has decided to use various selection methods [12, 13] for pre-selection and post-selection.

5.1 Post-selection

The following types of selections have been tested for succession purposes:

a) threshold - only the given upper percentage is included for selection,
b) random proportional: probability is proportional to fitness function value,
c) modified random proportional: probability is proportional to fitness function value less the lowest fitness function value over population.

Additionally, various values of elite size have been tested for random proportional and modified random proportional selection. The test parameters and the test results have been gathered in Tables 1- 5.

For 100 generations (Tables 2 - 3) basic threshold selection turned out to be more effective than most variants of other selection methods. The most probable reason is the superiority of fast convergence over diverse population for such a small number of generations. Very few variants of random selections obtained better results than threshold selection and all of them featured large elite sizes, which practically meant an approach very similar to threshold selection. The best results (0.9731) were returned by random proportional selection with an elite consisting of 80 individuals. The average fitness function value obtained for truncation succession for 100 generations was 0.9725.

Table 1. Test parameters

Test scenarios	Runs for each scenario and each succession method	Generations	Population size	Basic mutation probability	Fixing probability
91	5	100/200	100	0.05	1.00

Table 2. Average fitness function values (random proportional succession) for 100 generations

Elite size	Threshold			
	50% (only better half allowed)		100% (all allowed)	
	Elite returned		Elite returned	
	Yes	No	Yes	No
0	0.9386		0.8298	
5	0.9656	0.9652	0.9618	0.9625
10	0.9659	0.9685	0.9645	0.9667
20	0.9664	0.9700	0.9673	0.9699
40	0.9682	0.9717	0.9697	0.9703
60	0.9680	0.9722	0.9727	0.9728
80	0.9719	0.9730	0.9731	0.9730

Table 3. Average fitness function values (modified random proportional succession) for 100 generations

Elite size	Threshold			
	50% (only better half allowed)		100% (all allowed)	
	Elite returned		Elite returned	
	Yes	No	Yes	No
0	0.9436		0.8931	
5	0.9638	0.9665	0.9619	0.9632
10	0.9648	0.9678	0.9640	0.9665
20	0.9670	0.9698	0.9660	0.9676
40	0.9681	0.9710	0.9682	0.9688
60	0.9707	0.9725	0.9695	0.9699
80	0.9721	0.9728	0.9709	0.9713

Table 4. Average fitness function values (random proportional succession) for 200 generations

Elite size	Threshold			
	50% (only better half allowed)		100% (all allowed)	
	Elite returned		Elite returned	
	Yes	No	Yes	No
0	0.9476		0.8290	
5	0.9677	0.9699	0.9673	0.9560
10	0.9696	0.9710	0.9701	0.9710
20	0.9706	0.9722	0.9710	0.9719
40	0.9718	0.9739	0.9716	0.9731
60	0.9725	0.9740	0.9727	0.9747
80	0.9730	0.9741	0.9737	0.9748

Table 5. Average fitness function values (modified random proportional succession) for 200 generations

Elite size	Threshold			
	50% (only better half allowed)		100% (all allowed)	
	Elite returned		Elite returned	
	Yes	No	Yes	No
0	0.9559		0.9201	
5	0.9702	0.9700	0.9652	0.9649
10	0.9706	0.9714	0.9671	0.9672
20	0.9717	0.9722	0.9700	0.9705
40	0.9722	0.9736	0.9709	0.9729
60	0.9727	0.9739	0.9719	0.9740
80	0.9730	0.9741	0.9725	0.9741

For 200 generations (Tables 4 - 5) basic threshold selection was also competitive and the best results were reached by selections, whose elite sizes exceeded half of their populations. The highest fitness function value was again obtained for random proportional selection with elite of 80 individuals. This time however, the results indicate that with the growing number of generations the elite should not be returned to the population. The average fitness function value obtained for truncation succession for 200 generations was 0.9741.

Generally, this comparative simulation has shown that for this particular optimization problem, where operating in real time drastically limits the number of possible generations to about 100-200, the simplest, threshold selection is highly competitive. It is also more flexible, since it does not need adjusting the values of parameters depending on the situation.

5.2 Pre-selection

The following types of selections have been tested for crossover purposes:

a) threshold,
b) random proportional,
c) modified random proportional,
d) uniform – all individuals have the same chance of being selected as parents.

Similarly as in post-selection, various values of the elite size have been tested in case of uniform, random proportional and modified random proportional selections. The differences between results for various methods were insignificant and therefore it has been decided to apply uniform selection for pre-selection, which enabled resigning of population's additional evaluation directly preceding the crossover.

6 Example Results

In Figure 2 an example result returned by the method is presented. The method was run for 100 generations and for population size of 100. All the resulting trajectories are not only safe but also obeying the COLREGS rules. It is worth noticing that in this

particular situation there is only a minor way loss due to collision avoidance maneuverings, i.e. only ships 3, 5 & 6 have to extend their routes to avoid collision and ships 1, 2 & 4 perform course alterations mostly to avoid the shallow waters (dotted area) around the islands.

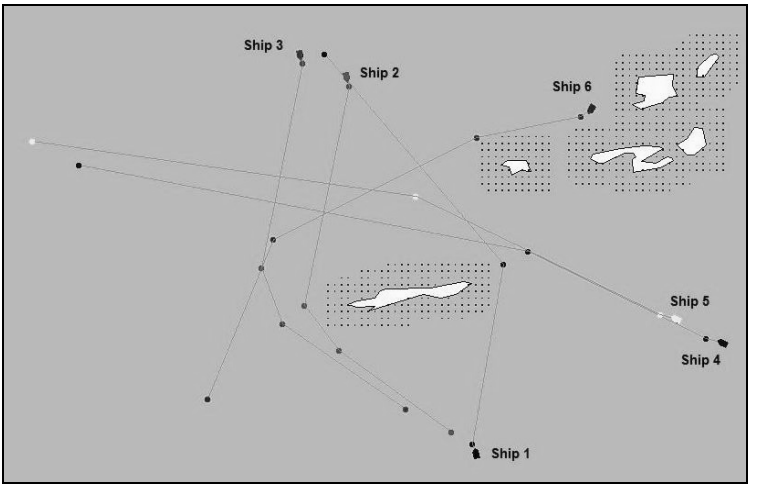


Fig. 2. Result trajectories for the encounter of six ships

7 Summary

The extended functionality of the Evolutionary Sets of Safe Ship Trajectories method has resulted in a much longer evaluation phase due to detection of constraints violations. This led to the necessity of boosting up the effectiveness of evolution. The improved evolutionary phases described in the paper are: reproduction (section 3) including selection for reproduction purposes (section 5.2) and selection for succession purposes (section 5.1). When put together, these upgrades contribute to the significantly better overall performance of the method. In particular, for the problem presented in the paper it turned out that some very generic selection methods achieved better results than more refined ones.

Acknowledgments. The author thanks the Polish Ministry of Science and Higher Education for funding this research under grant no. N N516 186737.

References

1. Lisowski, J.: Dynamic games methods in navigator decision support system for safety navigation. In: Proceedings of the European Safety and Reliability Conference, vol. 2, pp. 1285–1292 (2005)

2. Smierzchalski, R., Michalewicz, Z.: Modeling of a Ship Trajectory in Collision Situations at Sea by Evolutionary Algorithm. IEEE Transactions on Evolutionary Computation 4(3), 227–241 (2000)
3. Zeng, X.: Evolution of the safe path for ship navigation. Applied Artificial Intelligence 17, 87–104 (2003)
4. Tsou, M.-C., Kao, S.-L., Su, C.-M.: Decision Support from Genetic Algorithms for Ship Collision Avoidance Route Planning and Alerts. The Journal of Navigation 63, 167–182 (2010)
5. Beyera, H.G., Schwefela, H.P., Wegenerb, I.: How to analyse evolutionary algorithms. Theoretical Computer Science 287, 101–130 (2002)
6. Eiben, A.E., Schoenauer, M.: Evolutionary computing. Information Processing Letters 82, 1–6 (2002)
7. Michalewicz, Z., Fogel, D.B.: How To Solve It: Modern Heuristics. Springer, Heidelberg (2004)
8. Szlapczynski, R.: Solving multi-ship encounter situations by evolutionary sets of cooperating trajectories. In: Marine Navigation and Safety of Sea Transportation, pp. 437–442. CRC Press / Taylor & Francis Group / Balkema (2009)
9. Cockroft, A.N., Lameijer, J.N.F.: A Guide to Collision Avoidance Rules. Butterworth-Heinemann Ltd., Butterworth (2003)
10. Szłapczyński, R.: A unified measure of collision risk derived from the concept of a ship domain. Journal of Navigation 59(3), 477–490 (2006)
11. Szłapczynski, R., Szłapczyńska, J.: Evolutionary Sets of Safe Ship Trajectories: problem dedicated operators, submitted to ICCCT (in prep., 2011)
12. Cantú-Paz, E.: Order statistics and selection methods of evolutionary algorithms. Information Processing Letters 82, 15–22 (2002)
13. Wiese, K.C., Goodwin, S.D.: Keep-Best Reproduction: A Local Family Competition Selection Strategy and the Environment it Flourishes in. Constraints 6, 399–422 (2001)

Comparison of Selection Schemes in Evolutionary Method of Path Planning

Piotr Kolendo, Bartosz Jaworski, and Roman Śmierzchalski

Gdansk University of Technology, Faculty of Electrical and Control Engineering,
G. Narutowicza 11/12, 80-233 Gdansk, Poland
kolendo.piotr@wp.pl, the_bart@o2.pl, romsmier@pg.gda.pl

Abstract. This article compares an impact of using various selection schemes on the quality of the solution for the problem of planning the path for a moving object using the evolutionary method. In study case problem of avoiding collisions at sea is analyzed. The modelled environment includes static constraints (lands, canals, etc.) and dynamic objects (moving ships). Article analyses behaviour of selection schemes in two similar environments which differ in number of dynamic objects (highly congested areas). Research has proven that application of specific selectors improves results of study case evolutionary path planning situation.

Keywords: evolutionary algorithm, path planning, selection schemes.

1 Introduction

The problem of path planning occurs in numerous technical applications, such as, for instance, motion planning for mobile robots [12, 8], ship weather routing in ocean sailing or safety path planning for a ship in a collision situation at sea [5, 7, 8]. The problem is defined in the following way: having given a moving object and the description of the environment, plan the path for an object motion between the beginning and end location which avoids all constraints and satisfies certain optimization criteria. The problem can be divided into two basic tasks: an off-line task, in which we look for the path of the object in a steady environment, and an on-line task, in which the object moves in the environment that meets the variability and uncertainty restrictions. The on-line mode of the path planning relates to the control of the moving object in the non-stationary environment, in which parts of some obstacles reveal certain dynamics.

The main goal of the present paper is to compare various selection schemes in the evolutionary method of path planning and its impact on the quality and the computation time necessary for solving the problem. In this paper we consider a particular instance of the path planning problem as the navigation problem of avoiding collision at sea [4, 6]. By taking into account certain boundaries of the manoeuvring region, along with the navigation obstacles and other moving ships, we reduce the problem to the dynamic optimization task with static and dynamic constrains. We consider this an adaptive evolutionary task of estimating the ship path in the unsteady environment.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 241–250, 2011.

Based on the Evolutionary Planner/Navigator (EP/N) planning concept [3, 10, 11], we develop a modified version (υEP/N++) of the system [4, 7, 8] which takes into account specific nature of the process of avoiding collisions. The present version of the system uses different types of static and moving constraints to model the real environment of moving targets and their dynamic characteristics.

The paper is arranged in the following way: Section 2 gives the definition of (υEP/N++) ship navigation environment and the discussion of selected aspects of the evolutionary path planning. The selection schemes are discussed in Section 3. Section 4 describes simulation Environment, while Section 5 presents results of each selector application. Section 6 concludes the paper.

2 Evolutionary Method Path Planning

When determining the safe trajectory for the so-called own ship, we look for a trajectory that compromises the cost of a necessary deviation from a given route, or from the optimum route leading to a destination point, and the safety of passing all static and dynamic obstacles, here referred to as strange ships (or targets). In this paper the following terminology is used: the term own ship means the ship, for which the trajectory is to be generated, and strange ship or target mean other ships in the environment, i.e. the objects which are to be avoided. All trajectories which meet the safety conditions reducing the risk of collision to a satisfactory level constitute a set of feasible trajectories. The safety conditions are, as a rule, defined by the operator based on the speed ratio between the ships involved in the passing manoeuvre, the actual visibility, weather conditions, navigation area, manoeuvrability of the ship, etc.

Other constraints resulting from formal regulations (e.g., traffic restricted zones, fairways, etc) are assumed stationary and are defined by polygons – in a similar manner to that used in creating the electronic maps. When sailing in the stationary environment, the own ship meets other sailing strange ships/targets (some of which constitute a collision threat).

It is assumed that a dangerous target [7] is a target that has appeared in the area of observation and can cross the estimated course of the own ship at a dangerous distance. The actual values of this distance depend on the assumed time horizon. Usually, the distances of 5-8 nautical miles in front of the bow, and 2-4 nautical miles behind the stern of the ship are assumed as the limits for safe passing. In the evolutionary task, the targets threatening with a collision are interpreted as the moving dangerous areas having shapes and speeds corresponding to the targets determined by the ARPA system.

The path S is safe (it belongs to the set of safe paths) if any line segment of S stays within the limits of the environment E, does not cross any static constraint and at the times t determined by the current locations of the own ship does not come in contact with the moving representing the targets. The paths which cross the restricted areas generated by the static and dynamic constrains are considered unsafe or dangerous paths.

The safety conditions are met when the trajectory does not cross the fixed navigational constraints nor the moving areas of danger. The actual value of the safety cost

function is evaluated as the maximum value defining the quality of the turning points with respect to their distance from the constraints.

The EP/N is a genetic algorithm [11, 12] incorporating part of the problem maritime path planning specific knowledge into its structures. In general, the attractiveness of the use of the evolutionary techniques is connected with the fact that [8]:

- random search is believed to be the most effective in dealing with NP-hard problems and in escaping from local minima,
- parallel search actions not only secure high speed but also provide opportunities for interactions between search actions, all this acting in favour of better efficiency of the optimization,
- intelligent behaviour can be treated as a composition of simple reactions to a complex world,
- a planner can be much more simplified, and still much more efficient and flexible, and increase the quality of the search if it is not confined to the action within a specific map structure,
- it is better to equip the planner with the flexibility to change the optimization goals than the ability to find the absolute optimum solution for a single particular goal.

The EP/N realises all the above ideas by incorporating part of the problem specific knowledge into the evolutionary algorithm. What is evenly important and not quite obvious, due to the unique design of the chromosome structure and genetic operators the EP/N does not need a discretized map for search, which is usually required by other planners. Instead, the EP/N "searches" the original and continuous environment by generating paths with the aid of various evolutionary operators. The objects in the environment can be defined as collections of straight-line "walls". This representation refers both to the known objects as well as to partial information of the unknown objects obtained from sensing. As a result, there is little difference for the EP/N between the off-line planning and the on-line navigation. In fact, the EP/N realises the off-line planning and the on-line navigation using the same evolutionary algorithm and chromosome structure.

A crucial step in the development of the evolutionary trajectory planning systems was made by introducing the dynamic parameters: time and moving constraints (vEP/N++). Chromosome consists of path nodes (turning points), that are described by own vessel course, speed and coordinates of actual position. In the evolutionary algorithm used for trajectory planning eight genetic operators were used, which were: soft mutation, mutation, adding a gene, swapping gene locations, crossing, smoothing, deleting a gene, and individual repair [8]. The level of adaptation of the trajectory to the environment determines the total cost of the trajectory, which includes both the safety cost and that connected with the economy of the ship motion along the trajectory of concern. One can set the algorithm to weighed the particular element of the path cost so that i.e. the safest paths will be evaluated as the ones with the highest fitness even though they will at the same time represent the longest route. The level of adaptation of the trajectory to the environment determines the total cost of the trajectory, which includes both the safety cost Safe_Cost(S) and that connected with the economy Econ_Cost(S) of the ship motion along the trajectory of concern. The total cost of the trajectory (Fitness function) is defined as [7]:

$$Total\_Cost(S) = Safe\_Cond(S) + Econ\_Cond(S) \tag{1}$$

The safety conditions are met when the trajectory does not cross the fixed navigational constraints, nor the moving areas of danger. The actual value of the safety cost function *Safe_Cost*(*S*) is evaluated as the maximum value defining the quality of the turning points s_i with respect to their distance from the constraints:

$$Safe\_Cond(S) = w_c * clear(S) \tag{2}$$

where: $clear(S) = \max_{i=1}^{n} c_i$, w_c is the weight coefficient, c_i is the length difference between the distance to the constraint (the closest turning point s_i) and the safe distance *d*. The trajectory cost connected with the economic conditions *Econ_Cost*(*S*) includes: the total length of the trajectory *S* consisting of *n* line sections s_i, the function of the maximum turning angle between particular trajectory sections at turning points s_i , the time needed for covering the trajectory *S*. The total cost of the trajectory adaptation to the environment, resulting from the economic conditions, is equal to:

$$Econ\_Cond(S) = w_d\, dist(S) + w_s\, smooth(S) + w_t\, time(S) \tag{3}$$

where: w_d, w_s, w_t are the weight coefficients. Weight coefficients are determined by navigator preferences, concerning actual sailing conditions.

3 Selectors

In this section most popular types of selection schemes will be discussed [1] [3].

3.1 Roulette Wheel Selector

Roulette Wheel Selector chooses an individual from the population based on magnitude of its fitness score f_i relative to the whole population. The probability *p* of each individual being chosen is described in (4). Likelihood of selection is proportionate to the fitness score.

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{4}$$

3.2 Tournament Selector

In Tournament Selector user defines a number of individuals that are being chosen to the tournament (using Roulette Wheel selector). Winner of the tournament (Individual with the highest fitness function value) proceeds to the next generation. This selector chooses higher graded individuals more often than Roulette Wheel. By changing number of individuals in the temporary population (contestants in tournament), selective pressure can be controlled. Low numbers of contestants proved to achieve better results [1]. In implemented example the tournament will consists of 2 individuals.

3.3 Rank Selector

Rank Selector consists of two stages. In the first one, all individuals are ranked by their fitness function, which can be done in two ways. In the first case, individuals with the same fitness score have different ranks, while in the second one they are ranked identically. In the second stage we use function which defines the probability of each individual being chosen. In current work, the rank selector always returns the best individual as it is suggested in [9]. In case there is more than one individual with best fitness score, we choose one of them at random.

3.4 Deterministic Sampling Selector (DS)

This selector also consists of two stages. The expected representation of each individual is calculated in the first stage. Individuals with highest expected numbers get their number of expected copies in temporary population. Then the population is being sorted by decimal parts of each individual representation and those with highest values are fulfilling rest of the temporary population. In the second stage, Uniform Random Selector chooses individuals to be transferred to the next population.

3.5 Stochastic Reminder Sampling Selector (SRS)

SRS consists of two stages. In the first stage, expected representation of each individual is calculated similarly as it was in DS. Individuals with highest expected numbers get their amount of expected copies in temporary population. Then population is sorted by its decimal parts, where the decimal part is the probability of each individual being chosen. Rest of the temporary population is being filled with the individuals based on their decimal probability. In the second stage, Uniform Random Selector chooses individuals to be transferred to the next population.

3.6 Uniform Selector

In this selector every individual has the same probability p of being chosen. p is equal to 1 divided by the population size.

$$p_i = \frac{1}{n} \tag{5}$$

4 Simulation Environment

The simulation was performed in an environment populated with static and dynamic constraints. The static constraints were represented by black polygons, while the dynamic objects (characterised by their own course and speed) were marked by grey hexagons. The positions of the dynamic objects are displayed for the best route, which is bolded in the figures. Simulation scenario are presented in Figure 1. Figure 2 shows only the dynamic objects (targets), that reveal a potential point of collision [8] with own ship.

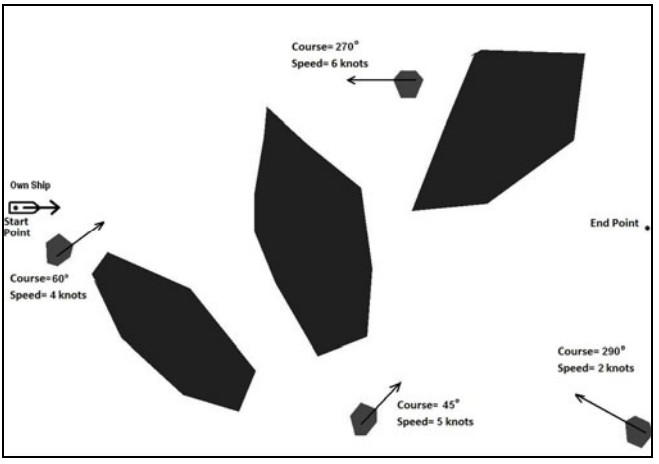


Fig. 1. Simulation environment - Scenario

Performance of Evolutionary Algorithm (EA) was tested for each selector for 7 random initial populations. Population size was 100, i.e. the evolutionary system processed 100 paths. Selectors use raw objective score in calculations. Probability of crossover and mutation is constant during all simulations. Steady State Algorithm [9] with overlapping population was used, where replacement percentage is 20%. In study case all selectors 3.1-3.6 were compared.

5 Simulations and Results

In this section results of selector types described in 3.1-3.6 were discussed. In scenario 7 initial populations were generated. Results of experiments are presented in Table 1. Based on the experiments, the single run terminal condition was set to 500 generations. As far as Table 1 are concerned, fitness value means the value of the best individual after 500 generations, number of generations means number of generations after which EA finishes it's exploration phase and starts exploiting local minima. Niche number means the way in environment that path goes through (Figure 1).

In 3 of 7 scenarios fitness function concentrate around one value for all selectors. In the remaining 4, Tournament selector achieved much better results compared to the other selectors. As we can deduce from Table 1, Roulette Wheel Selector and SRS performs worse than Tournament, however general fitness values are on acceptable levels. Rest of the selectors behave unstable, as they achieve worse value more often.

Final solutions for the first random population are presented in Figure 2. As it is depicted in the Figure 2a, Tournament selector found optimal route under criteria of distance, clearance and smoothness. Roulette Wheel Selector performs a bit worse in criteria of clearance however final result is acceptable. Rest of the selectors has fitness value even twice as large as Tournament's.

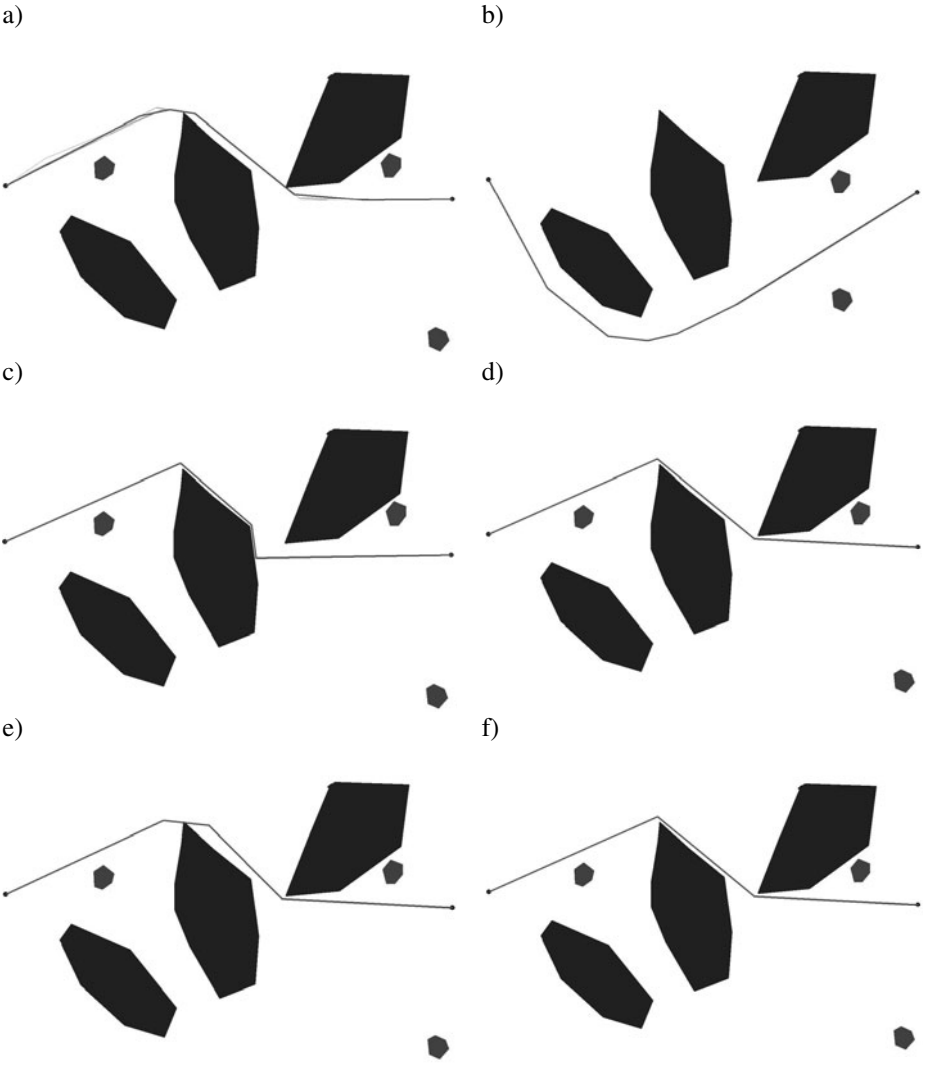


Fig. 2. Simulation results for Scenario, 1st initial population: a) RW b) Tournament c) Rank d) DS e) SRS f) Uniform

As far as number of generations of initial exploitation phase is concerned, average diversity level varied around 20 gen. As it was mentioned in Section 3, Tournament selector chooses highest individual more often than Roulette Wheel Selector. That is why the number of generations is lower than average. In study case Rank selector starts the exploitation phase very quickly and seems to block on the first good individual. Because of that applying Rank Selector (in this form) is not a good choice. Uniform selector needs the highest number of generations to reach the exploitation phase and there is no guarantee that better result will be achieved.

Table 1. Simulation Results for the environment - Scenario:

Random initial population	Selector Type	Fitness value	Number of generations	Niche number
1st	Roulette Wheel	270.56	12	1
	Tournament	207.713	15	3
	Rank	426.99	8	1
	DS	413.175	26	1
	SRS	414.991	25	1
	Uniform	413.175	70	1
2st	Roulette Wheel	204.348	20	2
	Tournament	202.249	10	2
	Rank	203.83	6	2
	DS	201.432	12	2
	SRS	200.901	16	2
	Uniform	203.009	40	2
3st	Roulette Wheel	285.476	20	2
	Tournament	248.873	10	1
	Rank	434.111	8	1
	DS	413.175	15	1
	SRS	260.081	16	1
	Uniform	413.175	20	1
4st	Roulette Wheel	292.971	20	3
	Tournament	215.994	10	3
	Rank	293.228	7	3
	DS	293.195	20	3
	SRS	292.855	23	3
	Uniform	208.852	30	3
5st	Roulette Wheel	200.763	30	1
	Tournament	204.172	12	3
	Rank	203.672	7	1
	DS	201.829	30	1
	SRS	201.599	40	1
	Uniform	203.213	50	1
6st	Roulette Wheel	411.998	34	1
	Tournament	211.708	16	3
	Rank	427.097	11	3
	DS	412.58	33	1
	SRS	211.013	25	3
	Uniform	278.398	90	1
7st	Roulette Wheel	292.971	20	3
	Tournament	215.994	10	3
	Rank	293.228	7	3
	DS	293.195	20	3
	SRS	292.855	23	3
	Uniform	208.852	30	3

6 Conclusions

As one can see from the performed experiments, the noticeable difference between selection types was found in the situation with not very congested environments. In the highly congested ones, only few of them presented better results, but as it was mentioned in previous section, it appears a random situation rather than a trend.

Experiments in Scenario show that Tournament Selector, Roulette Wheel Selector and Stochastic Reminder Sampling perform well in study case. The best, Tournament selector, gives better results in almost all cases, that makes value of the final result stable. Rest of the compared selectors provide unstable final results, by achieving values that differ much from the average value of final solutions more often.

In the performed experiments lengthening of exploration phase did not guarantee better final fitness value or finding other niches to be exploited.

EA almost always concentrates around the same niches for all of the selectors depending on which population was used as an initial one. In few cases EA found path around other niche. As one can deduce from the experiments, the choice of other niche did not provide better final fitness function value, so this situation has to be treated as a random one, not significant while comparing selection schemes.

In further research Tournament selector should be used due to the fact that in congested areas it performs similarly to the other selectors, while in not congested it significantly improves achieved results.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education under Grant N514 472039.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
2. Kolendo, P., Smierzchalski, R., Jaworski, B.: Experimental research on evolutionary path planning algorithm with fitness function scaling for collision scenarios. In: 9th International Symposium on Marine Navigation and Safety of Sea Transportation, TransNav 2011, Gdynia (2011)
3. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Heidelberg (1996)
4. Śmierzchalski, R.: Trajectory planning for ship in collision situations at sea by evolutionary computation. In: Proceedings of the IFAC MCMC 1997, Brijuni, Croatia (1997)
5. Śmierzchalski, R.: Ships' domains as collision risk at sea in the evolutionary method of trajectory planning. Computer Information and Applications II, 117–125 (2004)
6. Śmierzchalski, R., Michalewicz, Z.: Adaptive Modeling of a Ship Trajectory in Collision Situations at Sea. In: Proccedings of the 2nd IEEE World Congress on Computational Intelligence, ICEC 1998, Alaska, USA, pp. 364–369 (1998)
7. Śmierzchalski, R., Michalewicz, Z.: Modeling of a Ship Trajectory in Collision Situations at Sea by Evolutionary Algorithm. IEEE Transaction on Evolutionary Computation 4(3), 227–241 (2000)

8. Śmierzchalski, R., Michalewicz, Z.: Innovations in Machine Intelligence and Robot Perception. In: Path Planning in Dynamic Environments, pp. 135–154. Springer, Heidelberg (2005)
9. Mathew, W.: GAlib: A C++ Library of Genetic Algorithm Components. MIT, Cambridge (1996)
10. Xiao, J., Michalewicz, Z.: Soft-Computing and Mechatronics. In: Hirota, K., Fukuda, T. (eds.) An Evolutionary Computation Approach to Planning and Navigation, pp. 118–136. Physica, Heidelberg (1999)
11. Xiao, J., Michalewicz, Z., Zhang, L., Trojanowski, K.: Adaptive Evolutionary Planner/Navigator for Mobile Robots. IEEE Transactions on Evolutionary Computation 1(1), 18–28 (1997)
12. Yap, C.-K.: Algorithmic Motion Planning, In Advances in Robotics. In: Schwartz, J.T., Yap, C.-K. (eds.) Algorithmic and Geometric Aspects of Robotics, vol. 1, pp. 95–143. Lawrence Erlbaum Associates, Mahwah (1987)

Evidence Representation and Reasoning in Selected Applications

Włodzimierz Filipowicz

Gdynia Maritime University, Faculty of Navigation
81-225 Gdynia, Morska 81/83, Poland
wlofil@am.gdynia.pl

Abstract. Theory of evidence can be used to model uncertainty in many applications. Representation of relationship between evidence and hypothesis space is to be defined in practical cases. The relationship is usually expressed by an evidential mapping. The evidential mapping is a sort of belief assignment discussed in many papers. The Dempster-Shafer theory is used for further exploration of the relation between mentioned spaces. In the paper the concept is followed, explored and exploited in order to solve selected practical problems. One of them is floating objects detection. Mappings can be extended to include fuzzy sets. The extension is presented and exploited for position fixing in maritime navigation.

Keywords: evidence representation, Dempster-Shafer theory of evidence, detection problem, position fixing.

1 Introduction

Mathematical Theory of Evidence also called as Dempster-Shafer theory is able to handle uncertainty and ignorance. It is suitable to model imprecision expressed by intervals as well as fuzziness. Uncertainty and erroneous evidence is ubiquitous in maritime navigation. Possibility of detection of a floating object by a monitoring station can be considered as an uncertain fact described by an interval value. The theory offers embedded inference engine. Thanks to this mechanism conclusions regarding joined ability of detection can be drawn based on combination of several pieces of evidence delivered by monitoring stations that cover given area.

Position fixing engages fuzziness. Measurements taken in navigation are random values. Their mean error estimates are imprecise due to variety of applied methods and conditions during testing. Isolines or lines of position, curves of equal measured value, are used when graphical fixing is made [6]. Measurements errors affect isolines deflections. The relation: observation error – line of position deflection, depends on isolines gradients. All the mentioned issues contribute to an overall evidence to be considered once vessel's position is being fixed. Traditional approach is limited in its ability of considering mentioned factors. In order to include whole available evidence into a calculation scheme one has to engage Mathematical Theory of Evidence extended for fuzzy environment.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 251–260, 2011.

For above mentioned reasons crisp and fuzzy valued representation of evidence is considered in the paper. At first evidential mappings with binary sets are introduced. Then, form of uncertain rule is included. Modus ponens, inference pattern based on probability assignment, is presented next. The pattern is used to proof result obtained with complete evidence mapping introduced in [8]. Second part is devoted to fuzzy representation of evidence. Position fixing can be treated as reasoning on hypothesis space given measurements evidence. Scheme of combination as well as proofing final conclusions are depicted.

2 Evidence Crisp Representation

Mathematical Theory of Evidence operates on evidential mappings and exploits belief with plausibility measures. The evidential mapping represents relations among elements of two universes or frames of discernment respectively: evidence, denoted by Ω_E and hypothesis, indicated by Ω_H. For each element e_i within Ω_E space there exists a set of "subset of the power set of Ω_H - mass" pairs denoted by $m(e_i)$. It can be seen as a function $m: e_i \in \Omega_E \to 2^{\Omega_H} \to [0,1]$. The mapping takes the form as specified by Formula (1).

$$m(e_i) = \{(H_1, f(e_i \to H_1)), \cdots, (H_n, f(e_i \to H_n))\}. \tag{1}$$

In the above formula constraints (2) should be satisfied. Term belief structure is used for a mapping for which all below presented conditions are observed.

$$\begin{aligned} &1.\quad \Omega_H = \bigcup_{j=1}^{n} H_j \\ &2.\quad H_j \neq \varnothing \\ &3.\quad m_{ij} = f(e_i \to H_j) \geq 0 \\ &4.\quad \sum_{j=1}^{n} f(e_i \to H_j) = 1 \end{aligned} \tag{2}$$

Evidential mappings can be presented in a table. Each row is devoted to a single piece of evidence. Titles of columns refer to hypothesis frame subsets. Contents of cells are mass functions that are used for calculating credibility attributed to each of hypothesis subsets within given piece of evidence.

Two assignments referring to the same universes can be combined in order to increase their informative context. Combination of two mappings described by Formula (1) yields result structure that simplified version presents Formula (3).

$$m_c(e_c) = \{(H_{c1}, m_{c1}), \cdots, (H_{cl}, m_{cl})\}. \tag{3}$$

Within result structure elements m_{ci} are calculated based on Formula (4). Combination of two subsets $H_{1j} \cap H_{2k}$ can result in null set if conjunction operation is applied. Therefore greater than zero mass is assigned to empty set what means occurrence of inconsistency. Inconsistency violates constraints (2). In such case result assignment is

a pseudo belief structure and should be converted to its normal state by applying a normalization procedure. Several approaches to pseudo structures conversion have been suggested. One of them was proposed by Dempster, in the approach masses assigned to not null sets are increased by a factor calculated based on the sum of inconsistencies. The idea is embedded into Formula (4).

$$m_{ci} = K \cdot \sum_{H_{1j} \cap H_{2k} = H_{ci} \neq \emptyset} m_{1j} m_{2k}$$
$$K = \frac{1}{1 - \sum_{H_{1j} \cap H_{2k} = \emptyset} m_{1j} m_{2k}} \quad . \tag{4}$$

Association of two mappings can be carried out using combination table. Content of the table is subject of grouping and normalization in order to obtain combined belief structure. The result structure can be combined with the third mapping then with the next one until all cases are exhausted.

Belief and plausibility measures are calculated based on belief structure. Formula (5) delivers calculation details. To calculate belief for given set one has to add masses assigned to all subsets of the set. Calculation of plausibility engages all sets that intersection with the considered set is not empty. It was proved that plausibility is not less than belief measure if calculated for belief structure, assignment that satisfies constraints (2). The statement might not be true when calculations engage pseudo belief structure [11].

$$bel(H_i) = \sum_{H_{cj} \subseteq H_i} m_{cj}$$
$$pl(H_i) = \sum_{H_{cj} \cap H_i \neq \emptyset} m_{cj} \quad . \tag{5}$$

2.1 Modus Ponens Inference Pattern

Uncertain fact can be characterized by a range, for example [*a, b*]. The lower bound of the interval represents belief of the corresponding fact. The upper bound expresses plausibility value. Evidence mapping associated with uncertain fact P considered in $\{P, \neg P\}$ universe is in the form of pairs set as specified by Formula (6).

$$m(e) = \{(P, a), (\neg P, 1-b), (\Omega, b-a)\} . \tag{6}$$

Uncertain rule usually takes the form of: *if E then H* with uncertainty [*c*, *d*]. The rule is interpreted as relation between two universes, the same as evidential mapping. Thus it can be modeled by probability assignment. Adopted model depends on logical interpretation of the rule; implication and conditional probability interpretations are considered [7]. Under the latest interpretation interval limits show degrees to which E supports H. Therefore evidence mapping associated with uncertain rule $R{=}\neg E \vee H$ considered in $\{R, \neg R\}$ universe takes the form of pairs set as specifies Formula (7).

$$m(e) = \{(\neg E \vee H, c), (\neg E \vee \neg H, 1-d), (\Omega, d-c)\} . \tag{7}$$

Modus ponens inference procedure enables inferring on consequent Q from known antecedent P and a rule $P \rightarrow Q$. The rule states that P implies Q. Schematically the modus ponens inference procedure can be presented as [10]:

uncertainty of the rule *if … then …*:		$[c, d]$
uncertainty of the antecedent P	:	$[a, b]$
uncertainty of the consequent Q	:	$[x, y]$

The inference scheme engages two premises: the rule that is proved to be true inside certain interval and the antecedent that appeared to be true within given range. From these two premises, the range for the consequent of the conditional claim is calculated.

Two pieces of evidence with mappings (6) and (7) can be combined using the Dempster-Shafer scheme of association in order to calculate belief and plausibility of the consequent. Table 1 shows combination details.

Table 1. Modus ponens combination matrix

	$\neg P \vee Q$ $m_{11} = c$	$\neg P \vee \neg Q$ $m_{12} = 1\text{-}d$	Ω $m_{13} = d\text{-}c$
P $m_{21} = a$	$P \wedge Q$ $m_{c11} = ac$	$P \wedge \neg Q$ $m_{c21} = a(1\text{-}d)$	P $m_{c31} = a(d\text{-}c)$
$\neg P$ $m_{22} = 1\text{-}b$	$\neg P$ $m_{c12} = (1\text{-}b)c$	$\neg P$ $m_{c22} = (1\text{-}b)(1\text{-}d)$	$\neg P$ $m_{c32} = (1\text{-}b)(d\text{-}c)$
Ω $m_{23} = b\text{-}a$	$\neg P \vee Q$ $m_{c13} = (b\text{-}a)c$	$\neg P \vee \neg Q$ $m_{c23} = (b\text{-}a)(1\text{-}d)$	Ω $m_{c33} = (b\text{-}a)(d\text{-}c)$

Content of Table 1 enables direct calculation of belief and plausibility since there is no conflicting information included. Conflicting data refers to assignments to null sets. Therefore belief and plausibility of Q are as presented by Formula (8) [7].

$$bel(Q)=ac,\ pl(Q)=1 - a(1\text{-}d). \tag{8}$$

Based on belief and plausibility evidential mapping for the consequent can be easily obtained. Result assignment is shown by Formula (9).

$$m(e)=\{(Q, ac), (\neg Q, a - ad), (\Omega, 1 - a - ac + ad)\}. \tag{9}$$

In works devoted to knowledge representation and propagating beliefs [8], [9] an idea of complete evidential mapping is introduced. The mapping involves two power sets related to evidence and hypothesis spaces, it is defined as a function: $m: \ 2^{\Omega_E} \rightarrow 2^{\Omega_H} \rightarrow [0,1]$. The way of constructing the complete evidential mapping is also presented in the mentioned papers. Based on delivered rational inferring on consequent Q from known antecedent P and a rule $P \rightarrow Q$ can be carried out. The obtained result is the same as specified by Formula (9).

2.3 Detection Problem

Coastal waters are monitored by radar stations in order to enforce traffic regulations. Monitoring is also helpful during search and rescue operations. Allocation of radar stations should assure floating objects detection at predefined level all over supervised waters. Credibility that small crafts can be spotted even during bad weather conditions should be greater than stipulated value. To meet mentioned requirements detection and coverage optimization problem is defined [2]. In order to solve it regions located at the outskirts of ranges covered by the stations are to be explored. Mesh of points spanned over such areas is to be considered. For each of the points predefined ability of detection is to be granted for given class of objects and sea conditions. It is assumed that the stations operate autonomously and each of them is a source of independent evidence.

Let us consider a point covered by two stations, which ability of detection is uncertain and characterized by data gathered in Table 2.

Table 2. Set of detection characteristics

	detected	*¬detected*	*uncertainty*
station 1	0.5	0.2	0.3
station 2	0.2	0.2	0.6

There are two frames of discernment regarding ability of detection and magnitude of objects - respectively:

$\Omega_E = \{small, \neg small\}$, $\Omega_H = \{detected, \neg detected\}$.

Data collected in Table 2 give two evidential mappings regarding detection abilities by each of the stations:

$m(e_1)$ = {(*detected,* 0.5), (*¬detected*, 0.2), (*uncertain,* 0.3)} $m(e_2)$ = {(*detected,* 0.2), (*¬detected*, 0.2), (*uncertain,* 0.6)}

Joint ability of detection is to be calculated. The Dempster-Shafer scheme of combination is applicable to achieve the goal [2]. Association results embrace cases of conflicting assignments. Therefore pseudo belief structure is obtained and Formula (4) is to be applied in order to calculate cumulated and normalized masses. Next expression (5) is to be used to achieve belief and plausibility measures for joint detection characteristics. Final result is:

$m_c(e)$ = {(*detected,* 0.53), (*¬detected*, 0.26), (*uncertain,* 0.21)}

Assuming classification of a floating object in form of:

$m(e_3)$ = {(*small,* 0.8), (*¬small*, 0.1), (*uncertain,* 0.1)}

Possibility of detection obtained with Formula (9) is as follows:

$m(e)$ = {(*detected,* 0.45), (*¬detected*, 0.18), (*uncertain,* 0.37)}

3 Evidence Fuzzy Representation

Uncertain evidence involves fuzzy sets that embrace grades expressing possibilities of belonging to the set for consecutive hypothesis items. Each of the fuzzy sets has assigned credibility mass. Therefore fuzzy evidence mapping consist of "fuzzy set – mass" pairs. The mapping is described by Formula (10).

$$m(e_i) = \{(\mu_{i1}(x_k),\, f(e_i \to \mu_{i1}(x_k))), \cdots, (\mu_{in}(x_k),\, f(e_i \to \mu_{in}(x_k)))\}. \tag{10}$$

Constraints that are to be observed are specified by expression (11).

$$\begin{array}{ll} 1. & \mu_{ij}(x_k) = g(\{x_k\} \to p_{ik}) \neq \varnothing \\ 2. & \max\limits_{k} \mu_{ij}(x_k) = 1 \end{array}. \tag{11}$$

Fuzzy sets are represented by membership functions that reflect relations between the two universes. Membership function converts the hypothesis space into power set of [0, 1] interval: $\mu : \{x_k\} = \Omega_H \to 2^{[0,1]}$. Each piece of evidence is accompanied by a set $\{p_i\}$, therefore membership functions reflects relation between elements belonging to hypothesis space and sets attributed to elements of the evidence frame (see item 1 in constraints specification). Membership functions for nautical applications are discussed in author's previous papers [3], [4].

Item 2 in constraints specification stipulates normality of fuzzy sets. Normal sets should include highest grade equal to one. Apart from these two limitations typical for fuzzy mapping, additionally requirements 3 and 4 from Formula (2) are to be observed. Consequently belief structure should include all normal fuzzy sets and total sum of their masses is to be one.

Two fuzzy assignments can be combined in order to increase their informative context. Combination of two mappings specified by Formula (10) result in structure that simplified version presents Formula (12).

$$m_c(e_c) = \{(\mu_{c1}, m_{c1}), \cdots, (\mu_{cl}, m_{cl})\}. \tag{12}$$

Table 3. Combination of two evidence mappings with fuzzy sets

	μ_{11} $m_{11} = f(e_{11} \to \mu_{11})$	…	μ_{1m} $m_{1m} = f(e_{1m} \to \mu_{1m})$
μ_{21} $m_{21} = f(e_{21} \to \mu_{21})$	$\mu_{c11} = \mu_{11} \wedge \mu_{21}$ $m_{c11} = m_{11} \cdot m_{21}$		$\mu_{cm1} = \mu_{1m} \wedge \mu_{21}$ $m_{cm1} = m_{1m} \cdot m_{21}$
μ_{22} $m_{22} = f(e_{22} \to \mu_{22})$	$\mu_{c12} = \mu_{11} \wedge \mu_{22}$ $m_{c12} = m_{11} \cdot m_{22}$		$\mu_{cm2} = \mu_{1m} \wedge \mu_{22}$ $m_{cm2} = m_{1m} \cdot m_{22}$
…			
μ_{2n} $m_{2n} = f(e_{2n} \to \mu_{2n})$	$\mu_{c1n} = \mu_{11} \wedge \mu_{2n}$ $m_{c1n} = m_{11} \cdot m_{2n}$		$\mu_{cmn} = \mu_{1m} \wedge \mu_{2n}$ $m_{cmn} = m_{1m} \cdot m_{2n}$

Association of two fuzzy mappings can be carried out using combination matrix as presented in Table 3. In order to obtain result in form presented by Formula (12) one has to transform content of the table using expression (13). Obviously, obtained sets represented by membership function μ_{ci} might be empty or not normal.

$$m_{ci} = \sum_{\mu_{1j} \wedge \mu_{2k} = \mu_{ci}} m_{cjk} \ . \tag{13}$$

Combination of two fuzzy assignments is a transformation carried out on Cartesian product of two power sets of [0, 1], it takes the form $c : 2^{[0,1]} \times 2^{[0,1]} \rightarrow 2^{[0,1]}$. The result count can reach value close or equal to the product of combined sets counts. Result of association of two normal fuzzy sets can be empty or subnormal in case of applying null producing operators. In presented application conjunction operator is used therefore in each cell of combination table minimum values are selected as specified by Formula (14). Subsequently subnormal and empty fuzzy sets appear.

$$\mu_{cij}(x_k) = \mu_{1i}(x_k) \wedge \mu_{2j}(x_k) = \min(\mu_{1i}(x_k), \mu_{2j}(x_k)) \ . \tag{14}$$

In order to normalize obtained result calculations specified by Formula (15) are to be carried out. Note that m_{ci} is short for $m(\mu_{ci}(x_k))$. There are two goals to be achieved through normalization. They are conversion of subnormal sets and modification of their masses. Presented procedure was proposed by Yager [11] it can be seen as extension to fuzzy environment of the original Dempster transformation (see Formula (4)).

$$\begin{aligned} h_i &= \max_k(\mu_{ci}(x_k)) \\ m_{ci}^D &= \frac{m_{ci} \cdot h_i}{\sum_j m_{cj} \cdot h_j} \\ \mu_{ci}^D(x_k) &= \frac{\mu_{ci}(x_k)}{h_i} \end{aligned} \ . \tag{15}$$

Final belief structure consists of a family of fuzzy sets $\{ \mu_k(x_i) \}$ and a collection of masses assigned to each of the sets $\{ m(\mu_k(x_i)) \}$. Given these data sets support for hypothesis represented by a set of $\mu_A(x_i)$ is sought. It is fundamental ability of the evidence theory to reason on certain hypothesis based on associated evidence. Formula (16) defines belief and plausibility for proposition described by $\mu_A(x_i)$, which are embedded in collection of sets $\{ \mu_k(x_i) \}$ [1].

$$\begin{aligned} bel(\mu_A(x_i)) &= \sum_{k=1}^{n} m(\mu_k(x_i)) \min_{x_i \in \Omega}(\mu_A(x_i) \vee \neg\mu_k(x_i)) \\ pl(\mu_A(x_i)) &= \sum_{k=1}^{n} m(\mu_k(x_i)) \max_{x_i \in \Omega}(\mu_A(x_i) \wedge \mu_k(x_i)) \end{aligned} \ . \tag{16}$$

3.1 Position Fixing

Mathematical Theory of Evidence appears to be flexible enough to be used for reasoning on the fix. Contrary to the traditional approach, it enables embracing knowledge into calculations. Knowledge regarding position fixing includes: characteristics of random distributions of measurements as well as ambiguity and imprecision in obtained parameters of the distributions. Relation between observations errors and lines of position deflection is also important. Uncertainty can be additionally expressed by subjectively evaluated masses attributed to each of observations.

Let us consider three ranges related to three isolines as shown in Figure 1. Within ranges six strips were distinguished. Widths of the strips are calculated based on measurement errors and the isoline gradients. Each strip has fuzzy borders depending on imprecision in estimations of the isoline errors distribution. It is assumed that theoretical or empirical probabilities of containing the true isoline within each of the strips are given. Having particular point and all before mentioned evidence, support on representing fixed position for given point is sought. This is quite different from traditional approach where single point is sought and available evidence hardly exploited – for details see recent book in navigation [6].

The scheme of the position fixing is as follows:

Given: available evidence obtained thanks to nautical knowledge and experience

Question: what is a support that particular point can be considered as fixed position of the ship?

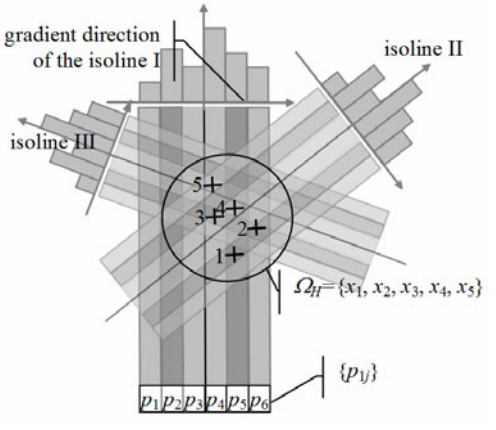


Fig. 1. General scheme of position fixing

Figure 1 shows common area of intersection of three areas associated with three isolines. Six strips were selected around each isoline, the strips were numbered as shown in the figure. Number 1 refers to the farthest section, number 6 indicates closest range according to gradient direction and regarding observed landmark(s).

Table 4 contains preliminary results of the example analysis. The table contains hypothesis points fuzzy locations within selected belts, locations are given with reference to the first of the isolines. Last, all one set (μ_{1u}), represents uncertainty. Weight assigned to this set expresses doubtfulness attributed to the measurement. Empirical probabilities assigned to each of the strips are included in last column.

Table 4. Fuzzy evidential mapping related to the first measurement

	x_1	x_2	x_3	x_4	x_5	$m(..)$
μ_{11}	{0	0	0	0	0}	0.05
μ_{12}	{0	0	0	0	0}	0.15
μ_{13}	{0	0	0	0	0}	0.30
μ_{14}	{0	0	1	0	1}	0.35
μ_{15}	{1	0	0	1	0}	0.10
μ_{16}	{0	1	0	0	0}	0.05
μ_{1u}	{1	1	1	1	1}	0.20

Presented mapping is a pseudo belief structure. Location vectors are not normal since their highest grade is less than one. The sum of all masses within presented mapping is not equal to one. Therefore pseudo belief structure was converted to its normal state with expression (15), the result is shown in Table 5.

Table 5. Normalized, fuzzy evidential mapping related to the first measurement

	x_1	x_2	x_3	x_4	x_5	$m^D(..)$
μ_{14}	{0	0	1	0	1}	0.47
μ_{15}	{1	0	0	1	0}	0.13
μ_{16}	{0	1	0	0	0}	0.07
μ_{1u}	{1	1	1	1	1}	0.33

Normalized belief structures are subject of combination in order to obtain knowledge base enabling reasoning on the position of the ship. Belief and plausibility of the proposition represented by a fuzzy vector included in collection of result sets can be calculated using Formula (16). In position fixing distances and/or bearings to the observed landmarks, from each hypothesis space element, are examined. Search space point with highest belief and plausibility measures is assumed to be fixed position of the ship. Reference set that represent hypothesis on, for example, point number 2 as being a fixed position is: $\mu_A(x_2) = \{0, 1, 0, 0, 0\}$. Thus general formula for belief and plausibility calculation can be reduced due to unique kind of the reference set. Formula (17) specifies details.

$$\begin{aligned} bel(x_l) &= \sum_{k=1}^{n} m(\mu_k(x_i)) \min_{x_i \in \Omega;\, i \neq l} (\neg \mu_k(x_i)) \\ pl(x_l) &= \sum_{k=1}^{n} m(\mu_k(x_i)) * \mu_k(x_l) \end{aligned} \quad . \tag{17}$$

Table 6. Final belief and plausibility measures

	x_1	x_2	x_3	x_4	x_5
belief	0.01	0.07	0.24	0.27	0.26
plausibility	0.05	0.15	0.36	0.35	0.32

It should be noted that belief cannot be considered as primary factor in considered position fixing problem. For discussion on this topic refer to paper [5]. Final belief and plausibility measures regarding considered points are included in Table 6.

4 Summary

In the first part of the paper binary evidential mapping was presented. Representations of uncertain facts and rules were considered. Modus ponens inference pattern was used for concluding on consequent given uncertain rule and its antecedent. Obtained result was the same as the outcome of the solution engaging complete evidential mapping proposed in [8], [9].

Mappings involving fuzzy sets were considered in the second part of the paper. Measurements taken in navigation deliver pieces of evidence with fuzzy location vectors. Each measurement enables creation of a single belief structure. Belief structures can be used for position fixing. Their combination results in a sort of knowledge base that should be explored in order to make a fix. Formulas enabling exploration of the base were presented. A point, within hypothesis space, with the highest plausibility and belief measures is selected as the ship's position.

References

1. Denoeux, T.: Modelling Vague Beliefs using Fuzzy Valued Belief Structures. Fuzzy Sets and Systems 116, 167–199 (2000)
2. Filipowicz, W., Neumann, T.: Detection and Coverage Problem. Zeszyty Naukowe Akademii Morskiej 7, 10–18 (2008)
3. Filipowicz, W.: Belief Structures in Position Fixing. In: Mikulski, J. (ed.) Communications in Computer and Information Science 104, Transport Systems Telematics, pp. 434–446. Springer, Heidelberg (2010)
4. Filipowicz, W.: New Approach towards Position Fixing. Annual of Navigation 16, 41–54 (2010)
5. Filipowicz, W.: Belief Structures and their Application in Navigation. Methods of Applied Informatics 3, 53–82 (2009)
6. Jurdzinski, M.: Principles of Marine Navigation. WAM, Gdynia (2008)
7. Lee, E.S., Zhu, Q.: Fuzzy and Evidence Reasoning. Physica, Heidelberg (1995)
8. Liu, W., Hughes, J.G., McTear, M.F.: Representing Heuristic Knowledge and Propagating Beliefs in Dempster-Shafer Theory of Evidence. In: Federizzi, M., Kacprzyk, J., Yager, R.R. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 441–471. John Willey and Sons, Inc, New York (1992)
9. Liu, W., Hughes, J.G., McTear, M.F.: Representing Heuristic Knowledge in D-S Theory. In: UAI 1992 Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco (1992)
10. Rutkowski, L.: Methods and Techniques of the Artificial Intelligence. PWN, Warsaw (2009)
11. Yager, R.: On the Normalization of Fuzzy Belief Structures. International Journal of Approximate Reasoning 14, 127–153 (1996)

Application of Artificial Intelligence Methods for the Diagnosis of Marine Diesel Engines

Adam Charchalis and Rafał Pawletko

Mechanical Faculty, Gdynia Maritime University,
81-87 Morska St. 81-225 Gdynia, Poland
pawletko@am.gdynia.pl

Abstract. The paper presents a diagnostic system for marine diesel engine based on an expert system model. The research relevant to knowledge acquisition for this system was done, knowledge base was built and general structures of the expert system was proposed. Basic sources of knowledge which can be used for construction of knowledge base are also identified. The basic knowledge related to the diesel diagnostic was undertaken from experts and diagnostic database. The paper questionnaire was used to the knowledge acquisition from experts. The basic knowledge related to the marine diesel exploitation was undertaken. The rule induction algorithms was used to knowledge acquisition from database. During the experiment efficiency of LEM induction algorithms was compared to new MODLEM and EXPLORE algorithms. Training and test data were acquired from experiment on marine engine Sulzer 3AL 25/30.

Keywords: technical diagnostic, expert system, marine diesel engines.

1 Introduction

The development of diagnostic systems for marine diesel engines is vital for both ship safety and economic reasons. Nowadays, many diagnostic systems have been created by both research laboratories and engine producers. Typical disadvantage of most systems is their completeness. This means that diagnostic algorithms of technical conditions, adopted during system creation, cannot be updated or modified during later operation.

The solution to the problem could be an expert system in ship engine diagnosis. Modular system structure, and above all, the separation of knowledge base from remaining program, enables creation of diagnostic system of open type, where diagnostic knowledge can be updated and cumulated.

This paper presents diagnostic system concept, for marine diesel engine, basing on expert system model. The relevant knowledge base was created with the use of collected diagnostic data.

Diagnostic data were collected from experts (ship engine professionals) and from diagnostic databases.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 261–270, 2011.

2 Marine Diesel Engine Diagnosis Systems

Modern diagnostic systems for marine diesel engines can be divided into two main groups. The first group of these are systems offered by major manufacturers of marine engines, such as MAN, Wärtsilä and Mitsubishi. These are typically complex systems, which are part of comprehensive engine room management systems [4, 9, 10].

In addition to the assessment of technical state of engines, these systems are responsible for the overhauls and operations management and spare parts managements.

Undoubtedly the greatest potential for assessing the technical condition of marine engines currently offer systems produced by major engine manufacturers. It should be noted here primarily CoCoS-EDS system, developed by MAN Diesel & Turbo and FAKS2i offered by Watrsilla.

CoCoS-EDS system belongs to the most complex marine diesel engines diagnosis systems. This system is the result of a joint project company MAN B&W and SEMT Pielstick, which uses experience gained during the development of systems-Geadit MODUS - for a four-stroke engines, and CAPA - for a two-stroke engines.

The CoCoS system is designed as a modular expert system, which aims to comprehensively manage the operation of the engine.

Knowledge base of the system includes the manufacturer's many years of experience gained during the design, manufacture and operation. CoCoS-EDS provides archiving, monitoring and trend analysis on the parameters recorded during the operation. Diagnosis is based on the symptoms of engines faults. This includes not only detect damage to the engine, such as uses and filters pollution, but also detection of sensors damages.

Similar capabilities to the system CoCoS-EDS offers FAKS2i system, which is a powerful tool to assess the technical state of engines. Like the CoCoS-EDS - FAKS2i is part of a comprehensive system to support the engine room operation, called RCOM. The RCOM also includes subsystems: ELDOC (Elektronic Documentation System), MAMA Pro (Maintanance Management System and a Remote Export (Multimedia Comunication System).

The heart of the system is FAKS2i rules expert system, which is the result of years of operating experience of specialists of the company. FAKS2i can include trend analysis of the recorded parameters, signaling the need for adequate maintenance activities and to detect defects in the early stages of development. The data source for the system can be integrated monitoring systems, such as the WECS (Wartsila Engine Control System) or MoniTrend. Evaluation of technical FAKS2i system is based on a mathematical model of the engine. Depending on external conditions and the current load the reference value is calculated for each diagnostics parameter. If any parameter exceeds the limit, the system automatically checks the deviation and gives the probable diagnosis. Presented is a list of all possible failures, and each is evaluated in terms of the most "probable".

Apart from the described complex systems for marine diesel engines diagnosis, there are groups of diagnostics systems offered by the manufacturers of measuring instruments and automation system producers. Mean Effective Pressure (MIP) Calculators are the most popular in these group. These systems allowing to measure the cylinder pressure curves on the basis of which set out a series of parameters

relating to the conduct of the combustion process. MIP Calculators offers the ability to measure additional parameters, such as the course of the injection pressure, pressure and temperature of charge air. Representative systems of this group are Autronica NK series, DIESELTUNE IV, Kyma Diesel Analyzer, CYLDET-CM Diagnostic System and CDS's Iowa.

Despite the obvious benefits of marine systems for diagnosing diesel engines, they are still of limited use on ships, due to high costs and lack of data on real economic benefits for their use.

3 Research Results

Most of the currently operating so called marine diagnostics systems are in fact measuring systems. They are not equipped with the faults location modules and the task of assessing the technical state of engines should continue to engineer. The quality of diagnosis therefore depends largely on human knowledge and experience.

A number of factors, such as the complexity and multiplicity of engine design and operation of the variable load condition meant that to date there is no universal, objective methods of assessing the technical condition of marine engines.

It seems necessary to try to build systems using the new developments of technical diagnostics and information technology including artificial intelligence methods to assist the evaluation of the technical condition of marine engines.

Expert systems represent a new generation of diagnostic equipment. By separating the knowledge base from the rest of the program, it can be easily modified without the need for intervention in the structure of the whole system. Such systems allow a combination of numerous team of specialists in the certain knowledge field. It is possible to simultaneously represent knowledge from different sources, such as experts, diagnostic database and simulation models.

The study attempts to acquire diagnostic knowledge for assessing the ship's engine, and the development of expert system with the use of this knowledge.

3.1 Sources of Knowledge

Diagnostic knowledge can be acquired from human experts or from diagnostics databases. The experts should play an important role especially in the initial stage of creating an expert system [1].

Given the incompleteness and lack of systematic knowledge for diagnosis of marine engines attempt to acquire it from the experts was done.

The development of information systems and their widespread availability, resulted in that they are increasingly being used in marine engine rooms. Modern control systems, in addition to measuring the number of parameters of the engine room, also allow automatic registration. As a result of these systems are created large collections of data, whose analysis and correct interpretation of increasingly exceeds the capacity of human. Therefore, there has been development of methods and tools supporting the process of acquiring knowledge from databases (called Knowledge Discovery).

Methods for knowledge acquisition from databases are an alternative for less effective methods of obtaining knowledge from experts. According to [5], these methods can be divided into two groups: machine learning (for examples of pre-classified) and the discovery in databases (for examples of non-scheduled). In studies that are the subject of work is limited to the methods of machine learning [5] .

3.2 Collecting Knowledge from Experts

The survey was to collect declarative opinions, useful in technical analysis of engine condition. Collection was made in a fashion where knowledge base programmer played a vital role [1]. The programmer was responsible for the interpretation and aggregation of collected expert opinions.

The survey was to obtain operational knowledge information in the form of diagnostic relations, like "damage – damage symptoms". Along with operational knowledge, collected information included basic appliance instructions, necessary for storage format of diagnostic report. Basic knowledge included dictionaries of object names, object property names, as well as, terms indispensable for data storage.

Expert data were collected during Questionnaire Interviews [2]. The questionnaire questions were tabled. Malfunction listing was created on the basis of problems addressed in professional literature [3, 4]. Questionnaire used open questions with an option of entering new malfunction names by the respondent. Survey experts were chosen on the basis of having at least second mechanic license with 2 years occupational experience. 36 experts took part in the survey.

The questionnaire included malfunctions of the following engine components:

- Fuel system,
- Crank-piston system,
- Combustion chamber,
- Turbocharging system,
- Starting-reversing system,
- Cooling system,
- Lubrication system.

Each expert was to name malfunction symptoms in relevant blank space of his questionnaire form.

Results made up 35 diagnostic rules. The rules included combined premises concluding on technical conditions of particular engine system.

Below presented is an example of such a rule:

```
Rule R1    system : injection    detail:  injector
Damage: injector nozzle seizing (injector open)
Symptoms:
  a) Mean effective pressure - drop
  b) Max. combustion pressure - drop
  c) Fumes colour change – smoking
  d) Combustion gases temperature in remaining
cylinders - growing
  e) Max. injection pressure - drop
```

Obtained sets of rules were verified in compliance with the procedures proposed by by W. Moczulski [1]. Single rules were verified by experts, by ranking them with degree of confidence on their validity.

3.3 Collecting Knowledge from Database

The target of the survey was to obtain rules enabling diagnostic of technical conditions of a marine diesel engine on the basis of exploitation information, available in database.

Each set of rules was determined with automatic induction method. LEM2 [5] classical algorithm results were compared with MODLEM [5] and EXPLORE [5] algorithms results.

The assessment of rule induction algorithms was made with the use of data obtained in active experiment on a laboratory engine.

The engine tested was a four stroke Sulzer 3Al 25/30, of nominal power Nn=408kW and n=750 rpm revolution. The engine was fitted with a measurement system enabling reading of basic operational parameters, like pressure and temperature of combustion gases, charged air, coolant and lubricator. Additionally, taken were fast-changing pressure measurements in engine cylinders and fuel lines. All parameters were automatically saved in a database integrated into measurement system.

The research program was conducted according to active experiment principles. During the experiment, each time, one level of particular malfunction was simulated and all parameters were measured, with engine working within 50 to 250 kW range. The experiment excluded simultaneous presence of multiple malfunctions, as well as, different ranges of particular malfunction.

The following malfunctions were taken into account:

- Air compressor efficiency drop,
- Turbocharger filter contamination,
- Air charger cooler contamination,
- Exhaust duct contamination,
- Injection pump leakage,
- Diminished injector opening pressure,
- Clogged injector nozzle,
- Badly calibrated injector nozzle,
- Leaky cylinder head.

Measurement results were saved in the database and converted to a decision table. Such a form of data presentation is required by adopted rule-induction algorithms. In such a situation, instructing examples are presented in the tabled verses, together with a set of attributes. One of the attributes is decisive, and qualifies a particular example to particular class of decisiveness [5].

The obtained table includes 454 instructing examples, where each of them is represented with 43 numerical attributes. Examples related to 9 simulated engine malfunctions.

Because algorithms LEM2 and EXPLORE should not be applied directly to numerical data, initial discretization was employed.

Investigated induction algorithms were applied both to non-discretized and discretized data.

Survey software, named ROSE2 was prepared Institute of Intelligent Systems of Decision Support at Poznan College of Technology [6, 7].

Evaluation of rule sets was made with regard to classification. It means that verified classifier was each time created basing on rules. Presented in table 1 are the rules and right classification choices obtained with 10-fold cross validation technique for investigated rule-induction algorithms. The results of classification are presented in Table 1.

Table 1. Classification results obtained with different algorithms

No.	Initial discretization	Induction Algorithm	Number of obtained rules	Percentage of correctly classified examples [%]	Percentage of incorrectly classified examples [%]	Percentage of non-classified examples [%]
1.	None	LEM2	178	24	32	44
2.		MODLEM	35	87	2	11
3.		EXPLORE	5	21	76	3
4.	Local Method	LEM2	56	91	9	0
5.		MODLEM	46	91	9	0
6.		EXPLORE	300	74	26	0

Obtained results indicate high efficiency of MODLEM algorithm in case of non-discretized data. The obtained classification accurateness, estimated with 10-fold cross validation technique, was 87 %. Classification accuracy obtained with LEM2 algorithm was, in this case, 24 %, while in case of EXPLORE algorithm, in 21 %. In the case of initial digitalization conducted with help of LEM2 and MODLEM algorithms, identical results were obtained. The lowest accuracy was obtained with EXPLORE algorithm.

One doubtless advantage of MODLEM algorithm, comparing with LEM2, is direct numeric data use, without the need for employing initial discretization. On one hand it simplifies the process of data gathering, on the other, improves readability and interpretation of the created rules. In such a situation, the Expert System user previews parameter values included in rule reasoning. Additional advantage of MODLEM algorithm, in spite of its slightly lower classification effectiveness, is low percentage of incorrectly classified examples (in case of initially non-digitalized data). Doubtful examples are then dropped without any classification.

3.4 System Concept

The following general assumptions were adopted in relation to the manner of operation of Expert Diagnostic System for ship engine:

- System user (ship mechanic) feeds data into a computer in form of answers to system-generated questions.
- System may also use data from ship database automatically.
- System generates diagnosis statements on engine qualification to the class of particular conditions.

Basic role of Expert System is to produce statement diagnosis while taking into account input data (fed directly by user or taken automatically from database).

Adopted architecture of the system is of module type. This enables, among other things, easy system updating by adding new elements and making multiple configurations. The system comprises of the following main modules:

- Database (changeable and constant data),
- Knowledge base,
- Knowledge obtaining module,
- Conclusion module,
- Interface module.

The structure of the system is presented in Figure 1.

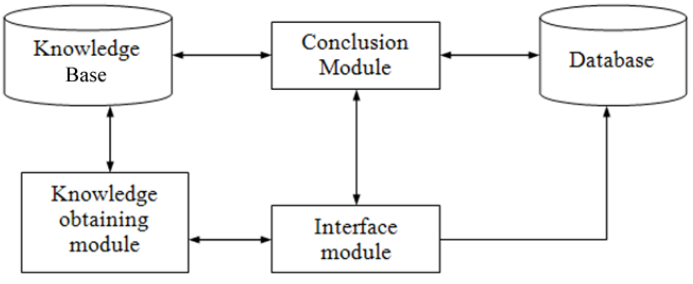


Fig. 1. The structure of the expert system

The most important detail of the whole system is Conclusion Module. It is responsible for the reasoning process and choosing relevant diagnosis. Conclusion Module and Interface Control use CLIPS language. CLIPS enables building basic system elements in homogenous environment. Conclusion Machine, Interface Control and Knowledge Database perform their duties in CLIPS environment.

Knowledge Management Module. Most of modern tools for expert systems, including CLIPS environment, lack dedicated software for knowledge base management. Knowledge base is typically included in a system in the form of a text file saved in the format accepted by particular expert system. During our system development, the necessity arose to create dedicated software for diagnostic

information management. The application was created in DELPHI environment and, as an independent module, is included in applications making up Expert System of Combustion Engine Diagnosis.

Knowledge Management Module performs the following actions:

- Installing and editing attribute name dictionaries, objects and malfunctions
- Updating, tracing and editing rules already saved in knowledge base
- Assessing rules saved in knowledge base
- Importing rules into knowledge base from ROSE2 environment
- Exporting knowledge base to CLIPS environment format

Knowledge Base. In knowledge base, saved is the basic information on application sector and operational instructions enabling engine diagnosis. Basic information includes object description and object classes, object attribute description, as well as, term dictionaries (of objects, attributes, malfunctions and symptoms).

Operational knowledge is included in rules, enabling assessments of engine conditions. Operational knowledge comprises two sub-basses. This is due to the different methods of recording expert information and automatically generated information. Expert information is of quality type. While collecting data from experts, it was fund that they tend to use expressions like "high temperature of combustion gases" or "low pressure of charged air". However, they have problems when it comes to expressing quantitative values of such attributes. In case of data obtained with inductive methods, existent rules relate to quantitative values of attributes. The decision to divide the knowledge base in order to convert it to uniform representations, was made for the following reasons:

- Quality information from experts, converted to quantity relations, would demand determining nominal engine and quantitative definition of terms like "high temperature" and "low pressure". The second part seems to be particularly difficult. The definition of "high temperature" may assume different meanings for different experts, and it may strongly depend on the kind of diagnosed engine.
- Quality relations seem to be the most suitable for recording general diagnostic relations and useful for diagnosing various kinds of engines. What is more, this kind of representation enables easy information update, because it intuitively corresponds with the reasoning shown by experts.
- It is generally possible to obtain quality information in an inductive way with the use of initial digitalization, however, it substantially complicates the process of the automatic information collecting. Discretization also diminishes readability of thus obtained rules and makes the dialogue with user difficult.

Taking into account the above difficulties of uniform representation of expert-obtained information and inductively obtained one, each source was stored independently. Expert-obtained information, of general nature, was presented in the form of qualitative rules. Inductively obtained information was saved in the form of quantitative rules (strict).

Quality information was obtained in expert interviews. The knowledge base includes 36 rules enabling diagnosing of chosen engine systems.

All rules, saved in qualitative knowledge base, were assessed by experts. The detection of malfunctions in the said engine systems with use of these rules is possible at expert-determined certainty level.

Another independent information knowledge base of Expert System makes the base with automatically obtained data. Here MODLEM algorithm was used. The rules saved in the knowledge base are quantitative in nature (strict). The knowledge base contains 35 rules enabling detection of chosen malfunctions of injection system, serviceable parts changing system and combustion chamber system.

The presented knowledge base is open. It can be developed and modified in any way.

4 Summary

Complex diagnostic systems for marine diesel engine diagnosis face limited application in ships, particularly due to their high cost. Ship engines are fitted with assorted indicators and measurement tools enabling control of many operational parameters, as well as, storing such measurements in databases. Technical condition verdict is however still the responsibility of the engine operator, and here comes the room for IT systems, which could facilitate such processes.

The expert system application may substantially enhance abilities of monitoring systems presently existent in power rooms, in respect of ship engine diagnosis. Such system enables saving valuable, operational knowledge for later use. Additional advantage, represents the opportunity of automatic collection of diagnostic information with machine learning methods. The usefulness of such methods for creation of diagnostic rules was proved on the basis of examples stored in database.

The expert system enables integration within a single frame of both information collected from experts and automatically collected one. A doubtless advantage of expert system is the opportunity of updating and developing the content recorded in the knowledge base. Due to this feature, the effectiveness of the system may grow during engine operation and facilitate gaining new experience.

References

1. Moczulski, W.: Methods of knowledge acquisition for machinery diagnostics, ZN Pol. Śląskiej nr 1382, Seria: Mechanika z. 130, Gliwice (1997) (in Polish)
2. Sobocki, M.: Introduction to methodology of pedagogical research. Oficyna Wydawnicza IMPULS, Kraków (2001) (in Polish)
3. Grzywaczewski, Z.: Reliability of ships. Wydawnictwa Przemysłu maszynowego WEMA, Warszawa (1988) (in Polish)
4. Żółtowski, B., Cempel, C.: Machinery Diagnostic Engineering, Polskie Towarzystwo Diagnostyki Technicznej, Instytut Technologii Eksploatacji PIB Radom, Warszawa, Bydgoszcz, Radom (2004) (in Polish)
5. Stefanowski, J.: Decision rules induction algorithms to discover knowledge, Wydawnictwo Politechniki Poznańskiej, Rozprawy nr 361, Poznań (2001) (in Polish)
6. Predki, B., Slowinski, R., Stefanowski, J., Susmaga, R., Wilk, S.: ROSE - Software Implementation of the Rough Set Theory. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 605–608. Springer, Heidelberg (1998)

7. Predki, B., Wilk, S.: Rough Set Based Data Exploration Using ROSE System. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, pp. 172–180. Springer, Heidelberg (1999)
8. Cholewa, W.: DT3D100 diagnostic advisory system. Organisation of the process of inference, Partial report nr DT6D131 of the project PBZ-038-06, KPKM Pol. Śląskiej, Gliwice (in Polish)
9. Łosiewicz, Z., Pielka, D.: Possible applications of artificial intelligence methods for diagnosis of marine diesel engine, Zeszyty Naukowe Akademii Marynarki Wojennej, Rok XLVI nr 162 K/2, Gdynia (2005) (in Polish)
10. Witkowski, K.: Status of technical diagnostics of marine piston engines. Diagnostyka 34 (2005) (in Polish)

Scalable System for Protein Structure Similarity Searching

Bożena Małysiak-Mrozek[1], Alina Momot[1], Dariusz Mrozek[1], Łukasz Hera[1], Stanisław Kozielski[1], and Michał Momot[2]

[1] Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
{bozena.malysiak,alina.momot,dariusz.mrozek,stanislaw.kozielski}@polsl.pl,
lukasz.hera@gmail.com
[2] Institute of Medical Technology and Equipment,
Roosevelta 118, 41-800 Zabrze, Poland
michal.momot@itam.zabrze.pl

Abstract. One of the advantages of using multi-agent systems in solving many problems is their high scalability adequately to the demand for computing power. This important feature underlies our agent-based system for protein structure similarity searching. In this paper, we present the general architecture of the system, implementation details, communication between agents, distribution of databases, and user interface. Moreover, presented results of numerical experiments show that distributing the computational procedure across multiple computers results in significant acceleration of the search process.

Keywords: proteins, protein structure, similarity searching, multi-agent system, distributed computing.

1 Introduction

Protein structure consists of one or more polypeptide chains which are folded into complicated 3D forms [1]. Different structures of proteins affect their functions and activities in biological processes and it can be stated that 3D structure of a protein largely determines its functional properties. Therefore, knowledge of protein 3D structures can yield useful information about their functional properties. For example, having an unknown protein structure we can conclude about its functional properties by comparing the structure to other known structures and finding mutual structural similarities.

Quantitative assessment of the similarity between two protein structures requires performing structure-to-structure comparisons. This is very time-consuming process, especially, when we want to compare a given, input structure, representing a result of some biological experiment, to the database of protein structures. Existing methods of similarity searching, like VAST [3], DALI [4], CE [5], FATCAT [6], allow to seek the 3D structural patterns in the database of protein structures. However, in recent years a great amount of new protein structures have been discovered. Many of them were stored in a well-known Protein Data Bank (PDB)

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 271–280, 2011.

[7]. Nowadays, the PDB contains 73 823 protein structures (Jun 14, 2011), and therefore finding similarity between protein structures constitutes a major challenge to maintain efficiency. For the last three years our group has been working on the system distributing the computational procedure to many autonomous agents and this way, providing the results in shorter time, proportionally to the scaling of the system horizontally by adding more working agents [8]. In this way, we join a community of researchers that use various techniques, like distribution of computational procedure [9] or cloud computing, in order to make the process of protein structure similarity searching more efficient.

This paper presents the application of the hierarchical multi-agent system to a large-scale comaprison of protein structures. We also show results of performed numerical experiments exploring the dependence of searching time on the number of working agents, as well as on the size of data transmitted between agents for sample proteins.

2 Architecture of Hierarchical Multi-agent System

Since 1996, when Franklin and Graesser [11] defined the essence of being an agent, the use of multi-agents systems for analysis, design, and development of complex domains is growing rapidly [12,13,14]. Composed of multiple interacting agents, multi-agent system (MAS) offers both the ability to increase computation speed (through agents operating asynchronously and concurrently) and reliability (because failure of one or several agents does not necessarily lead to the malfunction of the whole system and failure in achieving the intended goal) [15].

For these reasons, while developing our platform supporting the protein structure similarity searching, we decided to use a hierarchical multi-agent system. The theoretical basis of the hierarchical multi-agent system underlying our platform is described in [8]. The structure of the hierarchical multi-agent system consists of three types of agents: the **Supervisory Agent** responsible for communication with users, which distributes data for further processing over Control Agents and merges the results of the similarity searching process received from Control Agents; **Control Agents** responsible for communication with Supervisory Agent and distribution of searching tasks over Searching Agents, and finally, **Searching Agents**, which evaluate the similarity measures between two given proteins.

The main idea of the hierarchical multi-agent system is as follows. The user of the platform sets the query protein, for which he wants to find similar molecules. He also sets the method of protein structure similarity searching in selected databases and the length of the similarity ranking list B showing database proteins, which are the most similar to the input, query protein. Getting the parameters through the Graphical User Interface (GUI) (see fig.1), the Supervisory Agent divides the whole range of candidate proteins and assigns a range of data to search to each of Control Agents, which passes the task of searching to the Searching Agents dividing its range into smaller parts (packages). Each of the Searching Agents returns a list of similarity searching results and the lists are

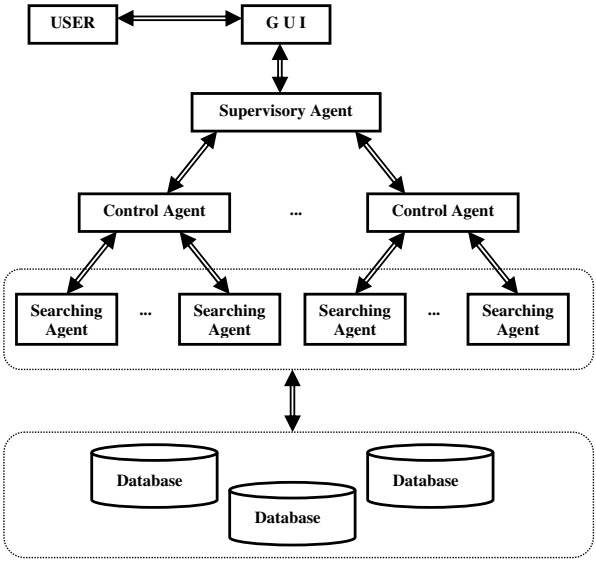


Fig. 1. Schema of the hierarchical multi-agent system architecture

merged into new sorted lists of the length B by each of Control Agents. Finally, the sorted lists are returned to the Supervisory Agent, which performs the last merging of the lists from all Control Agents into one sorted list and the list is returned to the User.

3 Implementation Details

The hierarchical multi-agent system described above provides the theoretical basis for the system and web application supporting protein structure similarity searching. The system was implemented using JADE (Java Agent DEvelopment Framework) [10], the open source platform for peer-to-peer agent based applications. We decided to use JADE, since it simplifies the construction and administration of agents and is compatible with FIPA (Foundation for Intelligent Physical Agents) standards. Agents of the JADE platform can run on remote computers, communicate with one another using messages and their behavior in a distributed environment can be controlled by implementing appropriate classes.

Web application, which forms a Graphical User Interface, is placed on the JBoss application server (`http://www.jboss.org/`). The JBoss server is released under the Lesser General Public License, is designed based on Enterprise JavaBeans (EJB) and implements the full set of Java EE (J2EE) services. There is also an exhaustive documentation of the server, a number of thematic forums and written advices [16].

For storing data of the application we use the Microsoft SQL Server 2005 Express Edition. The database provides storage of the results of comparisons for individual proteins, data requests made by users of the web application and its settings.

4 Communication between Agents

Proper communication between agents is one of the most important aspect of our system supporting protein structure similarity searching. Agents of the JADE platform exchange special messages carrying any text content. These messages can be divided according to the information conveyed by them and according to the direction of transmission. Nevertheless, one of the characteristic features of the communication system applied in the presented architecture is that the communication is mainly vertical, i.e. looking at the hierarchical architecture presented in figure 1, messages are sent from agents operating at higher levels to agents at lower level and opposite. Messages sent from Searching Agents to Control Agent are the following:

- message indicating the appearance of a new agent (newly created agent sends its data to the Control Agent),
- message indicating the completion of processing the package together with the results of this processing,
- message indicating that the agent passed in idle state,
- message stating the agent had died (if it happened during the processing package, this message will also contain details of the package).

Messages sent from the Control Agent to Searching Agents:

- message instructing the Searching Agent to process the package of proteins,
- message storing all parameters needed to perform the similarity searching.

In figure 2 we can observe sample flow of messages while new Searching Agents connect and register in the system. When a new Searching Agent connects to the system, it sends the message to the Control Agent reporting its readiness to act. The Control Agent registers the new agent and adds it to the list of agents remaining under its control. Then, the Control Agent sends a package of proteins to be processed by the Searching Agent. The package contains a list of protein identifiers that should be compared to the input molecule. The identifier of the input protein and parameters of the comparison process are sent as well.

When a Searching Agent completes processing a package, it sends results of the comparison process to the Control Agent. The Control Agent updates the status of the processed package to *completed*, and sends the next package of proteins to the Searching Agent. Afterwards, the Control Agent updates the ranking of similar molecules based on the results obtained from the Searching Agent. This situation is presented in figure 3.

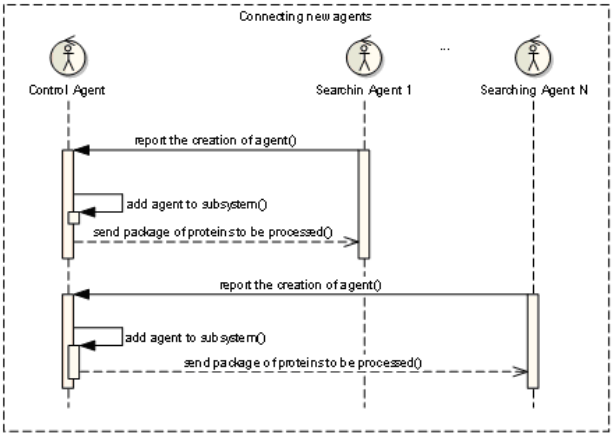


Fig. 2. Message flow during registration of new Searching Agents in the system

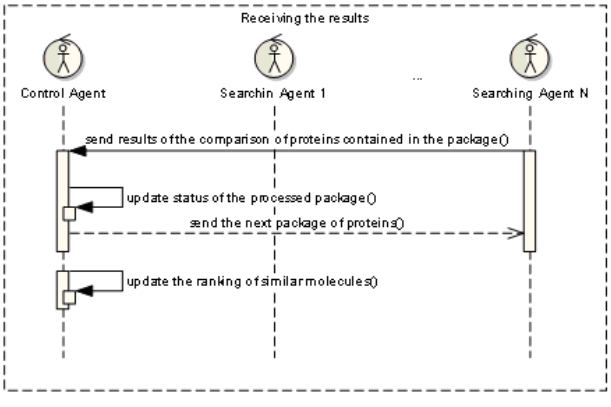


Fig. 3. Message flow when Searching Agent completes a package processing

5 Data Sources

Exploration of protein structures in such a distributed system requires appropriate arrangement of data sources in order to avoid the bottlenecks while retrieving structural data needed in the processing. For this reason, we decided to use various data sources:

- many local data repositories collecting subsets of protein structures replicated from the Protein Data Bank, accessed by Searching Agents while they perform comparison of structures,
- remote Protein Data Bank repository available through the FTP site, accessed by Searching Agents in case, when they cannot find a particular structure in the associated local repository,

– local database storing results of comparison processes, accessed by Supervisory Agent and Control Agents when they want to save partial or complete results of the similarity searching.

The term *remote* means that the repository remains remote to our system, outside the local network, where agents act. Similarly, the term *local* means the repositories are close to the developed system and are available through the fast network connections. These local repositories constitute a kind of cache for agents operating on protein structures.

6 Web Application

The system supporting protein structure similarity searching that we have developed is available through a dedicated web site. On the site users can submit input protein structures by specifying their PDB ID identifiers. They also choose the alignment algorithm, which will be used in the process (currently available are CE and FATCAT) - figure 4.

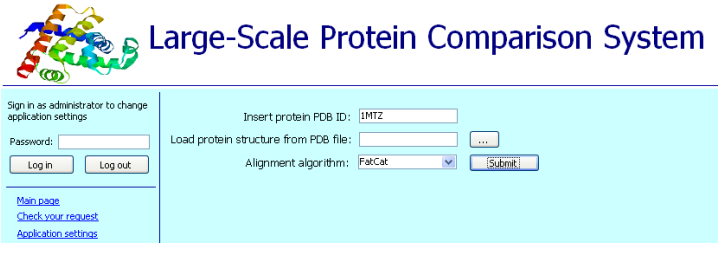


Fig. 4. Main site of the web application

Afterwards, the web site sends the search request to the main processing server and the procedure of similarity searching begins by distributing the work to agents: first the Supervisory Agent, next Control Agents and Searching Agents. The results can be available in a few minutes up to a few days, depending on the number of requests submitted. The new request is always inserted into a FIFO queue (First In Fist Out) and users can check the status of their requests by entering the token number, which was generated automatically when they submitted input protein structure. Users are also able to view partial results of the similarity searching during the process, even if it is not completed at all, and full results when the process is finished (figure 5).

Results contain PDB identifiers of molecules claimed to be similar, values of similarity and identity measures in percents, and the alignment time in seconds. Users can also see a detailed alignment of the input protein structure and the chosen database structure.

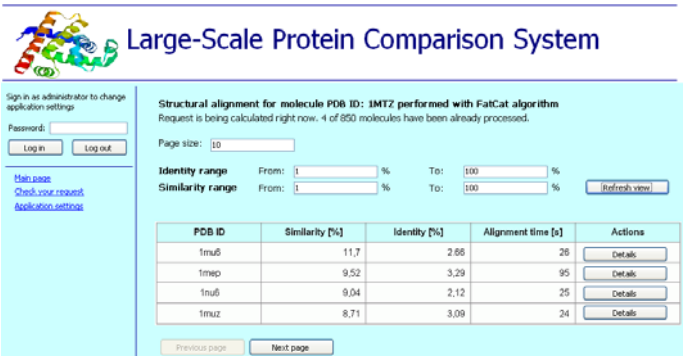


Fig. 5. Results of searching similarity for sample molecule

7 Numerical Experiments

In this section we present results of numerical experiments investigating the efficiency of the developed system working for different application settings and various numbers of Searching Agents.

All agents run at PC computers with different, but similar, hardware components (CPU and RAM) and managed by the Windows XP Professional or Windows 7 operating systems. Computers used in experiments were equipped with dual-core processors. On the basis of observations of CPU utilization there were established two agents on every used computer except the host computer, where the Control Agent resided. In the performed experiments there was only one Control Agent, therefore the Supervisory Agent was not needed and its tasks were realized by the Control Agent. The number of Searching Agents varied from 1 to 30. All tests were carried out from beginning to end on a fixed number of agents, i.e. during the calculations there were not joining or disconnecting Searching Agents.

The FTP server of the Protein Data Bank, located at `ftp.wwpdb.org/pub/pdb/data/structures/all/` contains tens of thousands of PDB files (73 823 files on Jun 14, 2011), representing different biological molecules. Invoking algorithms for comparing all of the files in this location would be extremely time-consuming - for a smaller number of agents it could take even a few days. For this reason most of the studies were carried out for the much smaller repository containing only 100 files. These files were chosen randomly and among them, there are proteins both with higher and lower complexity.

We present results just for two sample PDB proteins: 1qqa (Purine Repressor Mutant-Hypoxanthine-Palindromic Operator Complex) and 2o68 (Crystal Structure of Haemophilus influenzae Q58L mutant FbpA), which are presented in figure 6.

The first step was to check how long it would take to calculate similarities between the chosen protein and all proteins from the 100-element repository,

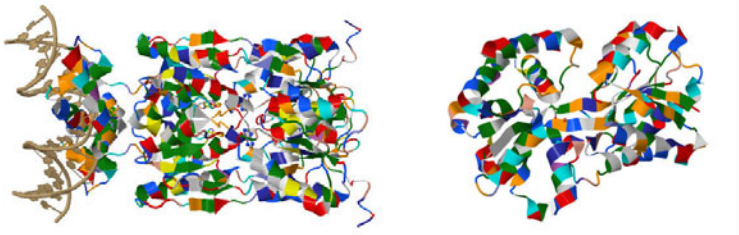


Fig. 6. PDB proteins: 1qqa (left) and 2o68 (right)

using one computer without the JADE agent platform. Using the FatCat algorithm, the time of similarity searching was 1595[s] for protein 2o68 and 1756[s] for protein 1qqa. The next step was to run the application, progressively increasing the number of agents connected to the JADE platform. On each subsequent computer there were run two agents. Table 1 presents the similarity searching time depending on the number of agents in a few selected cases.

Table 1. Dependence of searching time on the number of agents

Number of agents	Time for 2o68[s]	Time for 1qqa[s]
2	1524	1701
4	799	509
8	378	407
16	252	211

Figure 7 presents full results of dependence of searching time on the number of agents for 1qqa and 2o68 proteins. The number of agents varied from 2 to 20. As can be seen for small number of agents, acceleration growth is almost proportional to the increasing number of agents, but with further increasing the number of agents, the proportion of the value disappears. One of the reasons of this is that in case of the large number of agents more time is devoted to communication between them.

Further experiments investigated the effect of packet size, i.e. number of proteins to be processed, which is sent to a single Searching Agent. It was stipulated that the smaller package will increase the time needed to perform the entire searching process, because of the higher number of messages that must be sent between agents. However, performed experiments showed little dependency of the packet size on searching time. For example, comparing the 2o68 protein to 856 other proteins using the FatCat algorithm, for 20 agents, when the packet size was selected as 2, 5, and 10, the time was 1635, 1509 and 1590 seconds, respectively.

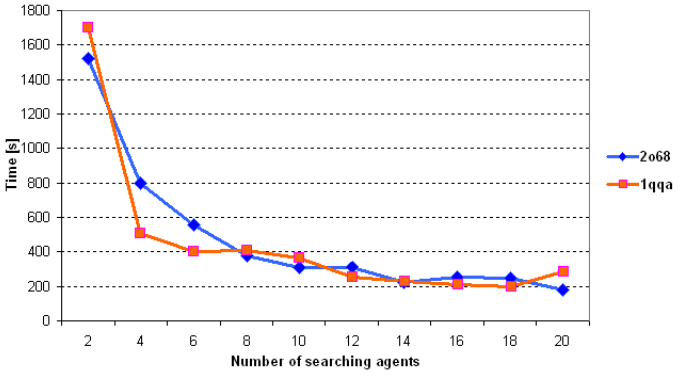


Fig. 7. Dependence of searching time on the number of agents

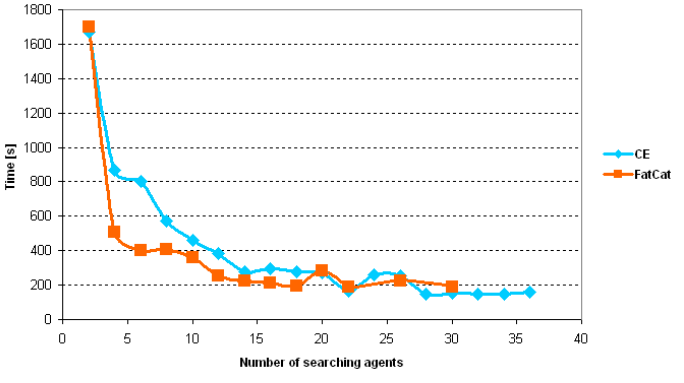


Fig. 8. Dependence of searching time on type of used algorithm

We also performed experiments studying how the searching time depends on the used algorithm. For the CE and the FatCat algorithms and for the 1qqa input protein, the results are presented in figure 8. As can be observed the execution times obtained using both algorithms are similar.

8 Concluding Remarks

Protein structure similarity searching is so time-consuming, that we tested presented system for small local repositories of molecules just to obtain the idea of how to configure the system for the future use. At the moment, we prepare to make the system available for the community of users, with the possibility of searching for similarities against the entire PDB database and all its resources. For this reason, we also tested the time of the similarity searching for the sample 1qqa protein against all PDB proteins using 30 Searching Agents (the packet size was 5 and the chosen algorithm was CE). The process took 89 157 seconds (about 25 hours), which we find as a promising result.

Acknowledgments. This scientific research was supported by the Ministry of Science and Higher Education, Poland in years 2008-2011, Grant No. N N516 265835: *Protein Structure Similarity Searching in Distributed Multi Agent System.*

References

1. Lodish, H., Berk, A., Zipursky, S.L., et al.: Molecular Cell Biology, 4th edn. W. H. Freeman and Company, NY (2001)
2. Branden, C., Tooze, J.: Introduction to Protein Structure. Garland, New York (1991)
3. Gibrat, J.F., Madej, T., Bryant, S.H.: Surprising similarities in structure comparison. Curr. Opin. Struct. Biol. 6(3), 377–385 (1996)
4. Holm, L., Sander, C.: Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 233(1), 123–138 (1993)
5. Shindyalov, I.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein. Eng. 11(9), 739–747 (1998)
6. Friedberg, I., et al.: Using an alignment of fragment strings for comparing protein structures. Bioinformatics 23(2), 219–224 (2007)
7. Berman, H.M., et al.: The Protein Data Bank. Nucleic Acids Res. 28, 235–242 (2000)
8. Momot, A., Małysiak-Mrozek, B., Kozielski, S., Mrozek, D., Hera, Ł., Górczyńska-Kosiorz, S., Momot, M.: Improving Performance of Protein Structure Similarity Searching by Distributing Computations in Hierarchical Multi-Agent System. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS (LNAI), vol. 6421, pp. 320–329. Springer, Heidelberg (2010)
9. Folino, G., et al.: Towards a Distributed Framework for Protein (Structure) Comparison, Knowledge, Similarity and Information (ProCKSI). In: UK eScience All Hands Meeting (AHM-2008), Edinburgh, UK, pp. 1–2 (2008)
10. Bellifemine, F., et al.: JADE, A White Paper (2003), http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf
11. Franklin, S., Graesser, A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: Jennings, N.R., Wooldridge, M.J., Müller, J.P. (eds.) ECAI-WS 1996 and ATAL 1996. LNCS, vol. 1193, pp. 21–35. Springer, Heidelberg (1997)
12. Wobcke, W., Pagnuccco, M., Zhang, C. (eds.): Agents and Multi-Agent Systems Formalisms, Methodologies, and Applications. LNCS (LNAI), vol. 1441. Springer, Heidelberg (1998)
13. Jain, L.C., Chen, Z., Ichalkaranje, N. (eds.): Intelligent agents and their applications. Physica, Heidelberg (2002)
14. Ghose, A., Governatori, G., Sadananda, R. (eds.): PRIMA 2007. LNCS (LNAI), vol. 5044. Springer, Heidelberg (2009)
15. Jennings, N.R., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems 1, 7–38 (1998)
16. Richards, N., Griffith, S.: JBoss: A Developer's Notebook. O'Reilly, Sebastopol (2005)

Efficient Algorithm for Microarray Probes Re-annotation

Pawel Foszner, Aleksandra Gruca, Andrzej Polanski,
Michal Marczyk, Roman Jaksik, and Joanna Polanska

Silesian University of Technology, Institute of Informatics,
Akademicka 16, 44-100 Gliwice, Poland
{pawel.foszner,aleksandra.gruca,andrzej.polanski,michal.marczyk,roman.jaksik,joanna.polanska}@polsl.pl
http://www.polsl.pl

Abstract. Systems for re-annotations of DNA microarray data for supporting analysis of results of DNA microarray experiments are becoming important elements of bioinformatics aspects of gene expression based studies [10]. However, due to the computational problems related to the whole genome browsing projects, available services and data for re-annotation of microarray probes are still quite sparse. The difficulty in developing systems of re-annotations of microarray probe sets is mainly in huge sizes of probe set data.

In our research we have created an efficient re-annotation method by combining the well known gene search tool BLAT with appropriately designed database. The elaborated system extends possibilities of existing re-annotation tools in the sense that: (1) by tuning parameters of all steps of re-annotation procedure any Affymetrix microarray standard chip can be automatically re-annotated in few hours, (2) several Affymetrix microarray chip standards are already precomputed.

Keywords: re-annotation, microarrays, expression data, affymetrix, classification.

1 Introduction

In the Affymetrix microarrays, the gene intensity is estimated/calculated on the basis of signals obtained from gene probes consisting of 25-mer oligo-nucleotides. Procedures for merging and normalizing signals from probes aim at obtaining reliable estimates of values of expressions of genes. In many cases, however, the estimated value does not match the true value of the gene expression. Technical sources of measurement noise introduce random error with controlled amount of variation. Important source of the mismatch between estimated and true values of expression is in the design procedure of microarray probes. One source of the design error is the presence of single nucleotide polymorphisms inside the oligo - sequence of the chip. More important errors coming from the design procedure of the microarray are related to assigning the probe to the locus in the genome

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 281–289, 2011.

different than desired. These errors are related to imperfections of assembly processes of genomes of organisms, like false sequences, gaps etc. The process of assembly of genomes is, however, continued and successively improved assemblies of genomes are published. Successive improvement of the quality of available data on genomic sequences opens the possibility of improving the quality of microarray measurements by re-annotating microarray probes, i.e., re-targeting all microarray probe sequences to newest versions of genomic sequences and modifying microarray definition files (CDF files) according to the results of this procedure. Researches focused on such possibility were already published in the literature, e.g., [10].

In this paper we report results of our work on creating and implementing an efficient algorithm for accurate targeting strings of nucleotide sequences, coming from microarray probes, to genomic databases. We have developed a methodology which provides information about how the probes are aligned to the latest built of the human genome. We used publicly available tools and databases for genomic sequences, BLAT [8], EST, mRNA, RefSeq databases [7] and the UniGene database [3]. Using mRNA and EST sequences databases we searched for matches to the genome, for each probe of the DNA microarray.

We have embedded the databases and technologies in the PHP and MySQL environment. The elaborated methodologies and tools allow the user both to access the re-annotation tools as a standalone program and as a web service. Our aim was also to use the elaborated tools for studying the improvement in gene probes definition obtained in successive assemblies of genomes, namely we researched statistics of status of gene probes. Status of the gene probe is determined by results of alignment of the gene probe against genomic databases, which enable verifying whether the gene probe (1) correctly and uniquely targets the desired gene, (2) miss-targets the gene, (3) targets the desired gene but does not fulfill the uniqueness property [10].

Results of using our algorithms for Affymetrix DNA microarrays are presented at two levels. The higher lever involves statistics stemming from comparisons of the status of original and re-annotated probes and the lower one involves comparing estimates of expression profiles, for exemplary data, between original and re-annotated microarrays. Contemplating the results reported in our research one can check how incorporating the latest improvements in human DNA assembly in DNA microarra data processing algorithms, can improve estimation of values of gene expressions.

2 Genomic Databases

All collected data concern human genome version 19. The entire human genome in this version has been downloaded from the UCSC website (University of California Santa Cruz) in 2bit format [7]. A .2bit file stores multiple DNA sequences, where each nucleotide is stored on 2 bits. This file was a database used by the BLAT program [8] for searching for matches of microarray probes. All the matches found by BLAT were linked to EST and mRNA sequences and further

assigned to corresponding genes. Therefore, there was a need to have a database of EST and mRNA sequences and database with information about the assignment of these sequences to genes. Databases that include sequences were downloaded from UCSC website [7] (files all_mrna.txt.gz and all_est.txt.gz). Database that contains information about assignment a sequence to the gene, was UniGene database. Downloaded version was 228, from the NCBI servers [3]. The last part of the algorithm verifies if the aligned probe does not belong to the non-coding region.

2.1 Pre-processing

Data obtained from the UCSC, were flat files representing MySQL tables. These are very large tables without any indexes, and without any possibility of creation of any unique index. This causes that the simplest search on a table with millions of rows becomes very costly in time. Given that microarrays have hundreds of thousands of probes, this results in unacceptable execution time.

BLAT [8] as a result of the calculation returns the name of the chromosome on which the match was found, and the starting and ending index where the match was found. Thus, in the SELECT statement we had two variables of type numeric and one string. Elimination of this last variable was a key element of the time optimization process. The main table has been divided into many tables, where every single table contains data on one chromosome. This resulted in the elimination of a string variable from select statement, which accelerated its execution.

At each stage of the algorithm, the structure of the database was changed for the most expensive queries. This fragmentation also allowed us to create multiple unique indexes, which again contributed to the time optimization.

3 Implementation

As a first step, we search for matching sequences in the human genome using the BLAT program. One of the pre-processing steps which BLAT program performs, is creation of a map of indexes for the searched database. This map is placed entirely in computer memory, and all search operations are performed on it. The map of the entire human genome build 19 uses around 6GB of memory, thus it was necessary to recompile BLAT program for use on x64 system architecture. The results of the program after compilation has been checked for consistency with the 32-bit version.

BLAT call parameter were default parameters for the standalone version, optimized for short sequences with maximum sensitivity [7] that is:

```
blat -stepSize=5 -repMatch=1000000 -minScore=0 -minIdentity=0 19.2bit
query.fa output.psl
```

The above parameters have the following interpretations: 19.2bit is a file including the data of the human genome build 19. The next file is a specially prepared FASTA file with probes sequences. It includes all the probes and each

of them is given a unique name. This name consists of a symbol of a probeset to which probe belongs and the coordinates defining its position on array. Below we present beginning of the file used for the re-annotation array HG-U133A:

```
> 1007_s_at:467:181
CACCCAGCTGGTCCTGTGGATGGGA
> 1007_s_at:531:299
GCCCCACTGGACAACACTGATTCCT
> 1007_s_at:86:557
TGGACCCCACTGGCTGAGAATCTGG
```

The last call parameter is the file which stores the results. After the execution of the BLAT program, we save only such results, which have no more than two mismatches and we remove all other probes from further analysis.

As the input the system takes results of the BLAT program and a list of probes with their unique names used in the FASTA file. In the next step, for all of the matches, we search for EST and mRNA sequences. Therefore, for each match we obtain a list of sequences that may belong to a gene. Annotation of a sequence to a particular gene is verified using the UniGene database and we keep only those that represent a gene according to the UniGene database. The last step of the algorithm verifies whether found match is in the coding or non-coding region.

The result of the program is a report including information about analyzed sequences. For each sequence we provide information about the matching gene, and whether it represents either a single or many genes. In addition, there is an information whether the probe is located in the coding or non-coding region.

Finally, we create a CDF file which includes only those probes that represent single genes and are located in the coding regions of these genes.

4 Re-annotation

To verify the quality of the performed re-annotation, we analyzed how it affects the expression data after normalization process. For this purpose, we compared expression values computed for two different CDF files. The CDF file describes the layout for an Affymetrix GeneChip array [1]. First of them was the original Affymetrix CDF file downloaded from the official web site and it was used as a reference file. The second one was generated on the basis of our method. This example re-annotation and CDF files were prepared for the HG-U133A array.

In the table 1 we present some statistic for the the CDF files for the HG-U133A matrix. The first column of the table includes statistics for the original Affymetrix CDF file and the second one represents the results after our re-annotation algorithm.

4.1 Microarray Data

The data were downloaded from the NCBI GEO DataSets database [3]. The example set has accession number GSE10072. This set contains data for the set

Table 1. A table showing the array HG-U133A statistics

	Affymetrix CDF file	Our CDF file
Number of probesets	22 216	11 708
Number of probes	22216	10954
Unigene build	133	228
Unigene release date	April 20, 2001	October 1, 2010
Human genome build	31	37.1

of samples of human tissues classified as lung cancer tissues or healthy tissues, described in the paper [9]. The study includes 107 tissues (58 tumor and 49 non-tumor tissues) from 74 patients. Each microarray experiment (tissue sample) is stored as a separate CEL file.

4.2 Microarray Expression Normalization

The raw data files (CEL files), were normalized using RMAExpress (Version: 1.0.5 Release) [4]. Normalization parameters were assumed as their default values, which are:

```
- Background adjust = yes
- Normalization = Quantile
- Summarization Method = Median Polish
- Store Residuals = false
```

After normalization we obtained two text files containing normalized expression measurements. Each of these files included expression values computed using different CDF file.

Further analysis were performed for three different expression groups: (1) the entire data set, (2) healthy tissues and (3) tumor tissues. The expression files were loaded into Matlab [2], and expression measurements were averaged in each of the group separately. Finally, we obtained three scatter plots, each for the different expression group.

4.3 Results of Expression Analysis

To analyze if, and how the values of expression levels differ before and after re-annotation, we created scatter plots, where each point represents normalized expression values. During the process of re-annotation we changed an arrangement of the probe sets, and created a completely new, based on alignment a probes to genes. Probes matched to the same gene were grouped into sets.

The plots in figures below show, for each gene, comparison the level of expression of the same data normalized by two different annotation methods. We took into account only the genes common to both CDF files. In our CDF file each gene is represented by a single probeset as opposed to original CDF file, where several probesets can represent one gene. Annotation of original Affymetrix probesets to

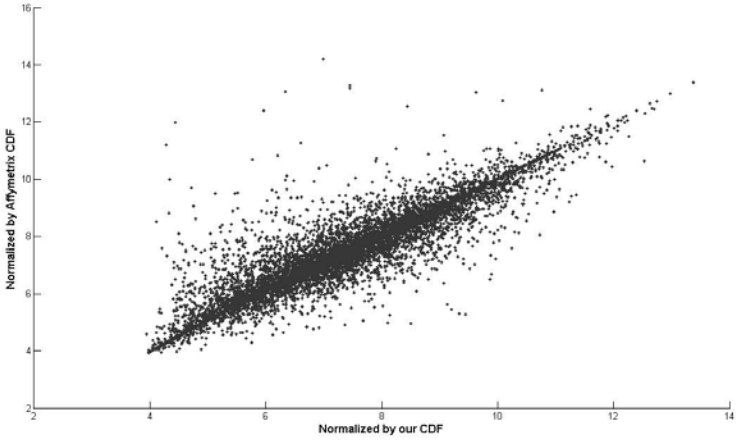


Fig. 1. Scatter plot for all 107 tissues

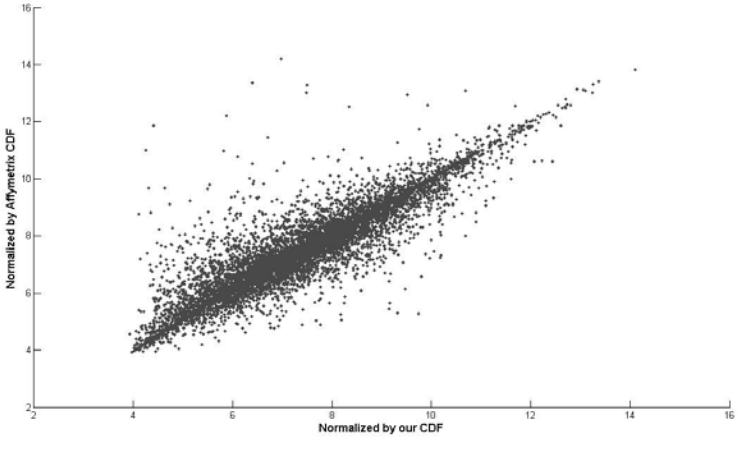


Fig. 2. Scatter plot for subset of normal tissues

genes were taken from HG-U133A Technical Support Documentation on official web site. We used NetAffx Annotation file, release 31 (August 23, 2010).

Scatter-plots were calculated for three subsets of data consisting of 107 microarray experiments. Figure 1 represents analysis of expression values for the entire data set. For each probeset, we averaged expression levels, and after that we compare this averaged values between data normalized using different CDF files. Figures 2 and 3 presents the expression values calculated for healthy and cancer 3 tissues respectively.

We also created two scatter-plots representing expression values for normal and tumor tissues. Fig 4 represents expression values obtained for original Affymetrix CDF files while the figure 5 shows the expression values computed for our CDF file.

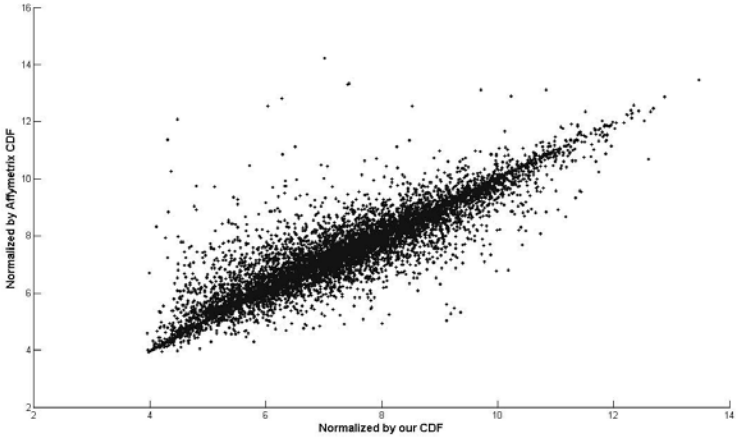


Fig. 3. Scatter plot for subset of tumor tissues

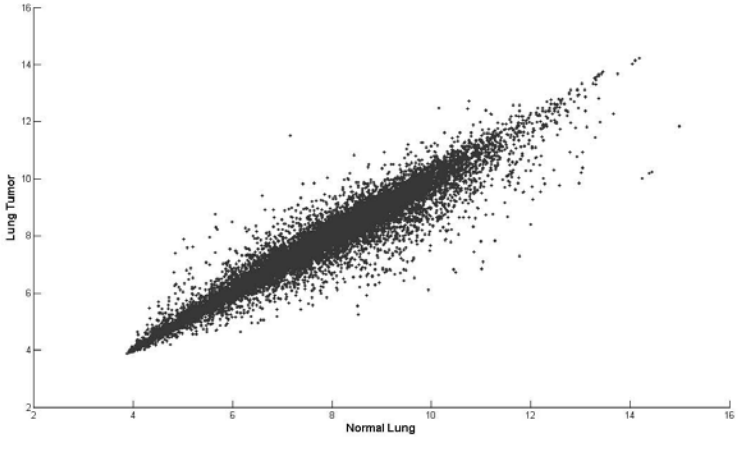


Fig. 4. Tumor tissues vs normal tissues – expression values obtained for original Affymetrix CDF file

Analysis of the results presented in figures 1 – 3 shows that expression values obtained using our method differs significantly from the expression values computed for original Affymetrix CDF file. By analyzing presented figures we can also observe that the differences between annotation methods are bigger than differences between normal and tumor tissues presented in figures 4 and 5. Such differences may affect the results of classification, clustering or many other operations performed on microarray expression data.

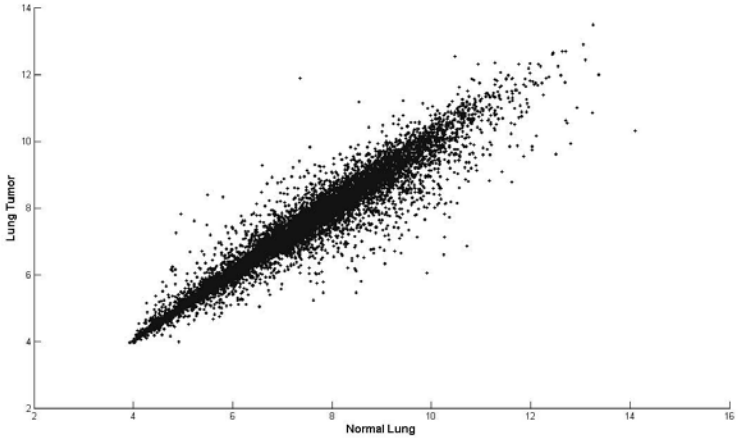


Fig. 5. Tumor tissues vs normal tissues – expression values obtained for our CDF file

5 Conclusions

In this paper we have presented new efficient re-annotation algorithm for Affymetrix DNA microarray probes. By using our algorithm we are able to target all probes of a microarray (of numbers of orders 100,000 - 800,000) to their true genomic location. Then, by reading annotations of their matches we can redesign procedures for computing values of expressions of genes of the microarray. The algorithm uses several available and widely used tools for genomic alignments, the most important one is the BLAT system. The algorithm also links to appropriate genomic databases such as EST, mRNA, RefSeq. These tools and services are combined with fast data parsing system and a local database. The project of the local database is optimized with respect to computational time.

Consistently to already published papers [10] our results show substantial differences between original projects of DNA microarray chips (as an example Affymetrix chip HGU133A is used) and the true locations of probesets. These results are shown in table 1. Moreover, when logarithms 2 of expression values are computed, by using standard RMA normalization procedure, for original project of the chip and for the chip with re-annotated probesets then again substantial differences are observed. These differences are presented, as scatter-plots, for real DNA expression data in figures 1 – 3. Importantly, by visual comparisons of figures 1 – 3 with the scatter-plots in figures 4 and 5, where data on healthy tissues for samples from [9] are compared with data on cancer tissues, stronger differences are observed for re-annotation effect than for the effect of difference between cancer and normal.

6 Further Research

In the further research we plan to re-annotate more Affymetrix arrays for different microarrays and organisms. We will also compare our result with normalized expression data obtained using different re-annotation methods, for example Ferrari [6] or Dai [5].

We also plan to analyze how the re-annotation can affect the quality of classification by comparing misclassification rates for classification of microarray data obtained using the official affymetrix CDF files and CDF files created by us. The information obtained from re-annotations reflects our current biological knowledge and thus application of updated CDF files can significantly improve the classification results.

Acknowledgments. This work was supported by the European Community from the European Social Fund.

References

1. Affymetrix official website, http://www.affymetrix.com
2. Matlab - the language of technical computing, http://www.mathworks.com/products/matlab/
3. National center for biotechnology information, http://www.ncbi.nlm.nih.gov
4. Bolstad, B.M.: RMAExpress Users Guide. 1.0.5 Release (2010), http://rmaexpress.bmbolstad.com/
5. Dai, M., Wang, P., Boyd, A.D., Kostov, G., Athey, B., Jones, E.G.: Evolving gene/transcript definitions significantly alter the interpretation of genechip data. Nucleic Acids Res. 33, e175 (2005)
6. Ferrari, F., Bortoluzzi, S., Coppe, A., Sirota, A., Safran, M., Shmoish, M.: Novel definition files for human genechips based on geneannot. BMC Bioinformatics 8, 446 (2007)
7. Fujita, P.A., Rhead, B., Zweig, A.S., Hinrichs, A.S., Karolchik, D., Cline, M.S., Goldman, M., Barber, G.P., Clawson, H., Coelho, A., Diekhans, M., Dreszer, T.R., Giardine, B.M., Harte, R.A., Hillman-Jackson, J., Hsu, F., Kirkup, V., Kuhn, R.M., Learned, K., Li, C.H., Meyer, L.R., Pohl, A., Raney, B.J., Rosenbloom, K.R., Smith, K.E., Haussler, D., Kent, W.J.: The ucsc genome browser database: update 2011. Nucleic Acids Research 38, D876–D882 (2010)
8. Kent, W.J.: Blat-the blast-like alignment tool. Genome Res. 12, 656–664 (2010)
9. Landi, M.T., Dracheva, T., Rotunno, M., Figueroa, J.D.: Gene expression signature of cigarette smoking and its role in lung adenocarcinoma development and survival. PLoS One 3(2), e1651 (2008)
10. Nurtdinov, R.M., Mikhail, O.V., Ershova, A.S., Lossev, S.I., Karyagina, A.S.: Plandbaffy: probe-level annotation database for affymetrix expression microarrays. Nucleic Acids Research 38, D726–D730 (2010)

Learning Method for Co-operation

Ewa Dudek-Dyduch and Edyta Kucharska

Institute of Automatics, AGH University of Science and Technology
al. Mickiewicza 30, PL 30-059 Krakow
edd@ia.agh.edu.pl, edyta@agh.edu.pl

Abstract. The aim of the paper is to present a new heuristic method for determining intelligent co-operation at project realization. The method uses local optimization task of a special form and is based on learning idea. The paper presents a formal approach to creation of constructive algorithms that use a local optimization and are based on a formal definition of multistage decision process. It also proposes a general conception of creation local optimization tasks for different problems as well as a conception of local optimization task modification on basis of gathering information. To illustrate the conceptions, the learning algorithm for NP-hard scheduling problem is presented as well as results of computer experiments.

Keywords: learning algorithms, local search techniques, optimization of co-operation, project management, scheduling problem, multistage decision process.

1 Introduction

The paper is related to the development of a new method within computational intelligence, applied to co-operation algorithms. Intelligent co-operation is defined as an accurate assignment of executors to tasks that form a project. On the one hand, a particular executor is supposed to realize tasks that it is best suited for, on the other, the selection of tasks and their ordering needs to take into account optimal (suboptimal) realization of tasks by other executors.

The aim of the article is to present a new heuristic method for determining intelligent co-operation at project realization. The method uses local optimization task of a special form and is based on learning ideas.

A project is composed of a number of tasks that need to be performed by various executors. The term executor may refer to people and/or machines characterized by a number of differentiating features. Executors should aim towards achieving a common target, which is the realization of a project at minimal costs, and meet restrictions which most often refer to execution time, costs and ordering of particular tasks. The execution time of particular tasks most frequently depends on the assigned executor and the extent of the tasks performed so far (the set of performed tasks). This is caused by the fact that some of the already realized tasks may facilitate or accelerate other task realization. The completed tasks may become the resources for realization of other tasks. Thus, such problems can be treated as scheduling problem with increasing resources which depend on the project state.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 290–300, 2011.

One must determine such an ordering of tasks and assignment of these to particular executors which would minimize the total project cost and meet the restrictions.

A lot of combinatorial problems correspond to the co-operation problem presented above, e.g. task scheduling for multiple machines, especially task scheduling with state dependent resources. Researches in this area are conducted in two main directions: developing algorithms for particular problems and developing general solution methods. The latter includes research devoted to metaheuristics and software tools implementing them. The paper also belongs to the second group of research as well.

The paper[6] presents a conception of software tools based on agent team approach and mainly improvement algorithms. Improvement algorithms can be applied only when the solution type can be defined a priori (for example, it is a permutation, set, vector, etc.). However, there are a number of problems for which type of solution cannot be determined a priori and consequently it is impossible to apply the improvement algorithm (see an example below). Then, it is necessary to apply a constructive approach.

The authors' research is related to methods based just upon this approach. The paper presents developing of ideas given in [3], [4], [8]. Its aim is 3-fold:

- to present a formal approach to creation of constructive algorithms that use a local optimization and are based on a formal definition of a multistage decision process,
- to present a general conception of creation and modification of local optimization tasks for different co-operation scheduling problems,
- to present an application of the method for NP-hard scheduling problem with state dependent resources.

It may be underlined, that the presented ideas of learning algorithm significantly differ from ideas for learning algorithms given in [7], [11].

2 Algebraic-Logical Model of a Multistage Decision Process

An admissible way of a project realization, and in particular assigning executors to tasks, can be determined with the help of simulation experiments. A single experiment establishes a sequence of decisions related to the assignment of executors (resources) to tasks and task realization ordering. It is impossible to provide a sensible sequence of decisions a priori. It needs to be established in the simulation course.

Simulation course of project realization consists in determining a sequence of process states and the related time instances. The new state and its time instant depend on the previous state and the decision that has been realized (taken) then. The decision determines the task to be performed, resources, transport unit, etc. Project realization processes belong to the larger class of discrete processes, namely discrete deterministic processes (DDP). A multistage decision process belongs to the class of DDP [1], [2].

Definition 1. A multistage decision process (MDP) is a process that is defined by the sextuple $MDP=(U, S, s_0, f, S_N, S_G)$ where U is a set of decisions, $S=X\times T$ is a set named a set of generalized states, X is a set of proper states, $T\subset \Re^+\cup\{0\}$ is a subset of non-negative real numbers representing the time instants, $f: U\times S\to S$ is a partial function called a transition function, (it does not have to be determined for all elements of the set $U\times S$), $s_0=(x_0,t_0)$, $S_N \subset S$, $S_G \subset S$ are respectively: an initial generalized state, a set of not admissible generalized states, and a set of goal generalized states, i.e. the states in which we want the process to take place at the end.

The transition function is defined by means of two functions, $f=(f_x,f_t)$ where $f_x: U\times X\times T\to X$ determines the next state, $f_t: U\times X\times T\to T$ determines the next time instant. It is assumed that the difference $\Delta t = f_t(u, x, t)-t$ has a value that is both finite and positive.

Thus, as a result of the decision u, that is taken or realized at the proper state x and the moment t, the state of the process changes for $x'=f_x(u, x, t)$ that is observed at the moment $t'=f_t(u, x, t)=t+\Delta t$.

Because of the fact that not all decisions defined formally make sense in certain situations, the transition function f is defined as a partial one. As a result, all limitations concerning the decisions in a given state s can be defined in a convenient way by means of so-called sets of possible decisions $U_p(s)$, and defined as: $U_p(s)=\{u\in U: (u, s)\in Dom f\}$.

At the same time, a MDP represents a set of its trajectories that starts from the initial state s_0. It is assumed that no state of a trajectory, apart from the last one, may belong to the set S_N or has an empty set of possible decisions. Only a trajectory that ends in the set of goal states is admissible. The control sequence determining an admissible trajectory is an admissible decision sequence. The task of optimization lies in the fact of finding such an admissible decision sequence $\tilde{u}$ that would minimize a certain criterion Q.

In the most general case, sets U and X may be presented as a Cartesian product $U=U^1\times U^2\times\ldots\times U^m$, $X=X^1\times X^2\times\ldots\times X^n$ i.e. $u=(u^1,u^2,\ldots,u^m)$, $x=(x^1,x^2,\ldots,x^n)$. Particular u^i represents separate decisions that must or may be taken at the same time and relate to particular executor (resources). There are no limitations imposed on the sets; in particular they do not have to be numerical. Thus values of particular co-ordinates of a state may be names of elements (symbols) as well as some objects (e.g. a finite set, sequence etc.). The sets S_N, S_G , and U_p are formally defined with the use of logical formulae. Therefore, the complete model constitutes a specialized form of algebraic-logic model of project. MDP can represent all potential possibilities of a project realization. According to its structure, the knowledge on a project is represented by coded information on U, S, s_0, f, S_N, S_G. The admissible trajectory corresponds to the admissible project realization.

3 Intelligent Co-operation Based on Learning

Let us consider co-operation scheduling problem when some previously performed tasks may become resources. To solve the problem one should determine assigning tasks to executors and order of task performing. An admissible solution corresponds

to admissible trajectory. The best admissible trajectory defines the best co-operation. The best co-operation is determined as a result of iterative procedure based on learning method. Thus, the presented learning method consists in generation of consecutive possibly better and better admissible solutions (admissible trajectory).

Trajectory generation is based on special local optimization tasks. The task lies in the choice of such a decision among the set of possibilities in the given state $U_p(s)$, for which the value of a specially constructed local criterion is the lowest. The form of the local criterion and its weight coefficients are modified in the process of solution search. The coefficients are modified as a result of learning process.

The local criterion consists of three parts and is created in the following way. The first part concerns the value of the global index of quality for the generated trajectory. It consists of the increase of the quality index resulting from the realization of the considered decision and the value related to the estimation of the quality index for the final trajectory section, which follows the possible realization of the considered decision. This part of the criterion is suitable for problems, which quality criterion is additively separable and monotonically ascending along the trajectory [1].

The second part consists of components related to additional limitations or requirements. The components estimate the distance in the state space between the state in which the considered decision has been taken and the states belonging to the set of non-admissible states S_N, as well as unfavorable states or distinguished favorable states. Since the results of the decision are known no further than for one step ahead, it is necessary to introduce the "measure of distance" in the set of states, which will aid to define this distance. For that purpose, any semimetrics can be applied. As it is known , semimetrics, represented here as ψ, differs from metrics in that it does not have to fulfill the condition $\psi(a,b)=0 \Leftrightarrow a=b$.

The third part includes components responsible for the preference of certain types of decisions resulting from problem analysis.

The basic form of the criterion $q(u,x,t)$ can be then represented as follows:

$$\begin{aligned} q(u,x,t)=\Delta Q(u,x,t)+\hat{Q}(u,x,t)+a1\cdot\varphi 1(u,x,t)+\ldots+ai\cdot\varphi i(u,x,t)+\ldots+an\cdot\varphi_n(u,x,t)+ \\ +b_1\cdot\rho_1(u,x,t)+\ldots+b_j\cdot\rho_j(u,x,t)+\ldots+ b_m\cdot\rho_m(u,x,t) \end{aligned} \quad (1)$$

where $\Delta Q(u,x,t)$ - increase of the quality index value as a result of decision u, undertaken in the state $s=(x,t)$, $\hat{Q}(u,x,t)$ - estimation of the quality index value for the final trajectory section after the decision u has been realized, $\varphi_i(u,x,t)$ - component reflecting additional limitations or additional requirements in the space of states, $i=1,2,...,n$, a_i - coefficient which defines the weight of i-th component $\varphi_i(u,x,t)$ in the criterion $q(u,x,t)$, $\rho_j(u,x,t)$ - component responsible for the preference of certain types of decisions that are responsible for co-operation, $j=1,2,...,m$, b_j - coefficient, which defines the weight of j-th component responsible for the preference of particular decision types.

The significance of particular local criterion components may vary. The more significant a given component is, the higher value of its coefficient is. It is difficult to define optimal weights a priori. They depend both on the considered optimization problem as well as the input date for the particular optimization task (instance). The knowledge gathered in the course of experiments may be used to verify these

coefficients. On the other hand, coefficient values established for the best trajectory represent aggregated knowledge obtained in the course of experiments.

The presented method consists in the consecutive construction of whole trajectories, started from the initial state $s_0=(x_0,t_0)$. Each generated trajectory is analyzed. If it is not admissible, the reasons of the failure are examined. For example, it is examined through which subsets of not advantageous states the trajectory has passed. A role of the criterion components connected with these subsets should be strengthened for the next trajectory i.e. the weights (priorities) of these components should increase. When the generated trajectory is admissible, the role of the components responsible for the trajectory quality can be strengthened, i.e. their weights can be increased. Based on the gained information, the local optimization task is being improved during simulation experiments. This process is treated as learning or intelligent searching algorithm.

The form of local criterion can by modified also in the course of trajectory generation.

4 Scheduling Problem with State Dependent Resources

The presented learning method is very useful for difficult scheduling problems especially for problems with state dependent resources. Managing projects, especially software projects, belongs to this class.

To illustrate an application of the method, let us consider a specific, very difficult scheduling problem that takes place during scheduling preparatory works in mines. The set of headings in the mine must be driven in order to render the exploitation field accessible. The headings form a net formally, represented by a non-oriented multigraph $G=(W,C,P)$ where the set of branches C and the set of nodes W represent the set of headings and the set of heading crossings respectively, and relation $P \subset (W \times C \times W)$ determines connections between the headings (a partial order between the headings).

There are two types of working groups. Each of them use driving machines that differ in efficiency, cost of driving, and necessity of transport. Machines of the first type (set $M1$) are more effective but their cost of driving is much higher than of the second type (set $M2$). Additionally, the first type of machines must be transported when driving starts at a different heading crossing than the one in which the machine is placed, while the second type of machines needs no transport. Driving a heading cannot be interrupted before its completion and can be done only by one machine at a time.

There are given due dates for some of the headings. They result from the formerly prepared plan of field exploitation. One must determine the order of heading driving and kind of working group which each heading should be driven by, so that the total cost of driving is minimal and each heading complete before its due date. For the sake of simplicity, group of workers will be identified by a type of machine it uses. There are given the following: lengths of the headings $dl(c)$, efficiency of both types of machines $V_{Dr(m)}$ (driving length per time unit), cost of a length unit driven for both kinds of machines, cost of the time unit waiting for both kinds of machines, speed of machine transport $V_{Tr(m)}$ and transport cost per a length unit.

Let us notice that the driven headings become the transport ways and may accelerate realization of other headings (tasks). Thus, it is problem with state dependent resources. It is *NP*-hard [8].

4.1 Formal Model of Problem

The process state at any instant t is defined as a vector $x=(x^0,x^1,x^2,...,x^{|M|})$, where $M=M1\cup M2$. A coordinate x^0 describes a set of headings (branches) that has been driven to the moment t. The other coordinates x^m describes state of the m-th machine, where $m=1,2,...,|M|$. A structure of the machine state is as follows:

$$x^m=(p, \omega, \lambda) \tag{2}$$

where $p\in C\cup\{0\}$ - represents the number of the heading assigned to the m-th machine to drive; $\omega\in W$ - the number of the crossing (node), where the machine is located or the number of the node, in which it finishes driving the assigned heading c; $\lambda\in[0,\infty)$ - the length of the route that remains to reach the node $\omega=w$ by the m-th machine (in particular $\lambda>$dl(c) meaning that the machine is being transported to the heading, the value λ is the sum of the length of heading c and the length of the route until the transportation is finished).

A state $s=(x,t)$ belongs to the set of non admissible states if there is a heading which driving is not complete yet and its due date is earlier than t. The definition S_N is as follows: $S_N=\{s=(x,t): (\exists c\in C,\ c\notin x^0) \wedge d(c) < t\}$, where $d(c)$ denotes the due date for the heading c.

A state $s=(x,t)$ is a goal if all the headings have been driven. The definition of the set of goal states S_G is as follows: $S_G=\{s=(x,t) : s\notin S_N \wedge (\forall c\in C,\ c\in x^0)\}$.

A decision determines the headings that should be started at the moment t, machines which drive, machines that should be transported, headings along which machines are to be transported and machines that should wait. Thus, the decision $u=(u^1,u^2,...,u^{|M|})$ where the co-ordinate u^m refers to the m-th machine and $u^m=C\cup\{0\}$. $u^m=0$ denotes continuation of the previous machine operations (continuation of driving with possible transport or further stopover). $u^m=c$ denotes the number of heading c that is assigned to be driven by machine m. As a result of this decision, the machine starts driving the heading c or is transported by the shortest way to the node of the heading c.

Obviously, not all decisions can be taken into the state (x,t). The decision $u(x,t)$ must belong to the set of possible (reasonable) decisions $Up(x,t)$. For example, a decision $u^m=c$ is possible only when the c-th heading is neither being driven nor completed and is available, i.e. there is a way to transport machine to the one of the heading crossing adjacent to the c-th heading or machine is standing in the one of the heading crossings adjacent to the c-th heading.

Moreover, in the given state $s=(x,t)$, to each machine waiting in the node w, (it has not assigned a heading to perform), we can assign an available heading or it can be decided that it should continue to wait. However, each machine which has been previously assigned a heading and is currently driving it or it is being transported to that heading, can be only assigned to continue the current activity. Based on the current state $s=(x,t)$ and the decision u taken in this state, the subsequent state

$(x',t')=f(u,x,t)$ is generated by means of the transition function f. The complete definition of the set of the possible decision $U_p(x,t)$ as well as the transition function f will be omitted here because it is not necessary to explain the idea of the learning method.

4.2 Learning Algorithm

The algorithm generates consecutive trajectories using the special local optimization task. The information gained as a result of the trajectory analysis is used to modify the local optimization task for the next trajectory. This approach is treated as a learning without a teacher.

The local criterion takes into account a component connected with cost of work, a component connected with necessity for trajectory to omit the states of set S_N and a component for preferring some co-operation decisions. Thus, the local criterion is of the form (a_1, b_1 - weights of particular components):

$$q(u,x,t)=\Delta Q(u,x,t)+\hat{Q}(u,x,t)+a_1 \cdot E(u,x,t)+b_1 \cdot F(u,x,t). \quad (3)$$

where $\Delta Q(u,x,t)$ denotes the increase of work cost as a result of realizing decision u and $\hat{Q}(u,x,t)$ the estimate of the cost of finishing the set of headings matching the final section of the trajectory. The third component $E(u,x,t)$, connected with the necessity for the trajectory to omit the states of set S_N, is defined by means of a semimetrics. The fourth component is aimed at reduction of machine idleness time and at performing transport ways. Since the model considers the possibility that the machines will stand idle in certain cases, it seems purposeful to prefer decisions which will engage all machines to for most of the time. It is therefore necessary to reduce the probability of selecting the decision about machine stopover when headings are available for driving and machines could be used for work. This may be realized by using an additional auxiliary criterion $F(u,x,t)$, which takes into consideration penalty for a decision about a stopover in the case when a machine could have started another work/task.

In the course of trajectory generation, the local optimization task may be changed. Problem analysis reveals that the moment all headings with due dates are already finished, it is advisable to use only cheaper machines. Formally, this corresponds to the limitation of the set of possible decisions $U_p(s)$. Moreover, it is no longer necessary to apply the component $E(u,x,t)$ in the local criterion. The modified criterion is as follows: $q(u,x,t)=\Delta Q(u,x,t)+\hat{Q}(u,x,t)+ b_1 \cdot F(u,x,t)$.

In order to select a decision in the given state s, it is necessary to generate and verify the entire set of possible decisions in the considered state $U_p(s)$. For each decision u_k, it is necessary to determine the state the system would reach after realizing it. Such a potentially consecutive state of the process will be represented as $s_{p\_k}=(x_{p\_k},t_{p\_k})$. Afterwards, the criterion components are calculated. The increase of cost $\Delta Q(u_k,x,t)$ is the sum of costs resulting from the activities of particular machines in the period of time t_{p_k}-t. The estimate of the cost of the final trajectory section $\hat{Q}(u_k,x,t)$ can be determined in a number of ways. One of these is to establish the summary cost of finishing previously undertaken decisions, which realization has not been completed yet, and the cost of a certain relaxed task, realized in the cheapest way. Taking into consideration that the estimate should take place with the lowest

number of calculations, there has been proposed relaxation which would include omitting temporal limitations and the assuming the least expensive procedure for finishing the remaining headings; this would involve using the least expensive machines.

The component $E(u_k,x,t)$ uses the value of the estimated "distance" between the state s_{p_k}, and the set of inadmissible states. The distance is estimated with the help of semimetrics $\psi(s,S_N)=min\{\psi(s,s'): s'\in S_N\}$.

Assuming that the speed of transporting the "fastest" machine is significantly higher than its speed of performance, and this one in turn significantly exceeds the speed of performance of the remaining machines, it is possible to omit the time of transporting of the fastest machine. For the sake of simplicity, let us assume that there is just one fastest machine.

One of the methods of determining $E(u_k,x,t)$ is to calculate the time reserve $rt_c(s_{p\_k})$ for each not realized and not assigned heading c with due date.

Taking into consideration these assumptions $rt_c(s_{p\_k})=d(c)-t_{p\_k}-\tau(c)-t_{end}$, where $d(c)$ denotes due date for the heading c, $\tau(c)$ denotes time necessary to drive heading c and all the headings situated along the shortest route from the heading c to the so-called *realized area* in the given state, by the fastest machine, t_{end} denotes time necessary to finalize the current activity of the fastest machine.

The parameters t_{end} and $\tau(c)$ results for following situations. First, "the fastest" machine may continue the previously assigned task to drive another heading. Second, heading c may be inaccessible and it might be necessary to drive the shortest route to this spot from the realized area, involving already excavated headings as well as those assigned for excavation together with relevant crossings. For the period of time when the fastest machine finishes the excavation of the previously assigned heading, a fragment of this distance may be excavated by the fastest of the remaining machines. When the heading is accessible the time equals the time of excavating the length of the heading with the fastest machine.

Finally, this component is as follow:

$$E(u_k,x,t)=\begin{cases}\infty & for\ \min rt_c(s_{p\_k})<0\\ \dfrac{1}{\min rt_c(s_{p\_k})} & for\ \min rt_c(s_{p\_k})\geq 0\end{cases} \tag{4}$$

As a result, the decision to be taken, from the set of considered decisions, is the one for which the subsequent state is most distant from the set of non admissible states.

Component $F(u_k,x,t)$ in certain cases should make it purposeful to prefer decisions which will engage all machines for most of the time. It is therefore necessary to reduce the probability of selecting the decision about machine stopover when headings are available for driving and machines could be used for work, especially for transport ways performance. For that purpose, a penalty may be imposed for the stopover of each machine that could potentially start driving an available heading. The proposed form is $F(u_k,x,t)=P\cdot i_{waiting}$ where P denotes the penalty for machine stopover, (calculated for the decision about stopover when there are still headings available for work), $i_{waiting}$ denotes the number of machines that are supposed to remain idle as a result of such a decision.

The values a_1 and b_1 are respectively coefficients defining the weight of particular components and reflect current knowledge about controls, whilst their values change in the course of calculations. The higher the weight of a given parameter, the higher its value.

If the generated trajectory is non-admissible, then for the subsequent trajectory, the value of weight a_1 should be increased; which means the increase of the weight of the component estimating the distance from the set of non-admissible states and/or the increase of weight b_1 value, which would result in lower probability of machine stopover. If the generated trajectory is admissible, then for the subsequent trajectory the values of coefficients may be decreased.

4.3 Experiments

Optimal executor co-operation results in the best possible value of global cost. In considered problem, the fact that some executors (machines) perform the job that enables work for another one is the essence of cooperation. In particular, in the case of little time reserve of some heading (small value of semimetric) some executors may prepare transport way for fast executor which should perform that heading to deadline. Component *F(u,x,t)* corresponds with it. The aim of experiments was to verify the effectiveness of proposed method for executor co-operation (especially form of local criterion and applying the components *E(u,x,t)* and *F(u,x,t)*).

The research was conducted for the set of 8 heading networks. Each network is represented by a planar graph, in which the vertex degrees equal from 1 to 4. The lengths of headings are numbers from the range [19, 120]. The number of headings with due dates is approximately 25% of all headings. Two machines are used to perform the task, one of the first type and one of the second type. Parameters of machines are given in the Table 1. The symbols GI and GII denote regular and irregular structure of the network respectively.

Table 1. Parameters of the machines

Parameter	Machine of M1 type	Machine of M2 type
Efficiency [m/h]	10.0	5.0
Transport speed [m/h]	100.0	unspecified
Driving cost [$/h]	200.0	50.0
Transport cost [$/h]	100.0	0.0
Waiting cost [$/h]	30.0	5.0

Firstly, the effectiveness of component *E(u,x,t)* was tested. For each network 40 trajectories was constructed with the changing value of coefficient a_1 and zero value of coefficient b_1. Based on the obtained results, it can be concluded that increasing the value of *E(u,x,t)* increases the probability of obtaining an admissible solution. When the component *E(u,x,t)* was omitted, an admissible solution was not found.

Secondly, component *F(u,x,t)* influence on the total cost was tested. For each network trajectories were constructed with the changing value of coefficient b_1. In most cases, the increase of weight of this component resulted in increased total costs, but at the same time the probability of finding an admissible solution was higher.

To evaluate the effectiveness of the proposed algorithm one has compared the obtained results with the optimal solution. A complete review algorithm for all considered heading network has been applied. Optimal solution was found only for network GI-2. In other cases, the calculations were interrupted after more than 2 days. Only for four networks an admissible solution has been found, while for the other networks an acceptable solution has not been found at this time. Table 2 presents the summary of performed experiments. The second column contains the optimal cost for network GI-2 and the best found total cost by a complete review algorithm for which calculation was stopped after 2 days. The symbol "*" means that any admissible solution has not been found during 2 days. The next columns respectively contains the best found total cost by a proposed algorithm, value of coefficient a_1 and value of coefficient b_1 for this cost. Moreover, percentage of the cost of using machine *M1* in the total cost and percentage of the length headings which was made by machine *M1* in total driving work is given.

Table 2. Results of applying the components *E(u,x,t)* and *F(u,x,t)*

Network	Best found cost by complete review alg.	Best found cost by proposed alg.	Value of coeff. a_1	Value of coeff. b_1	Cost of using machine *M1*[%]	Length made by machine *M1*[%]
GI-1	16480,00	16922,00	50000	2	70%	50%
GI-2	16524,40	17284,10	55555	2	75%	58%
GI-3	17448,90	16359,90	25000	0	61%	41%
GII-4	23132,40	22665,17	500000	0	61%	35%
GII-5a	*	30946,32	250000	0,5	65%	42%
GII-5b	*	30888,33	250000	0	62%	37%
GII-6	31142,54	30662,31	5000	0	60%	35%
GII-7	*	77288,44	250000	1	72%	52%

The result of experiments shows that the difference between sub-optimal cost and the best found by the learning algorithm is small and in the worst case is 4.59%. Percentage difference of cost is calculated as (LAcost-RAcost)/RAcost where LA, RA denote best cost calculated by learning algorithm and completed review algorithm respectively. It can be stated that the learning algorithm finds a better solution in most cases. It should be also pointed that its calculation time was very short (a several seconds).

Based on the obtained results, it can be concluded that the application of the proposed algorithm for the presented problem yields very positive results. Results of another experiment are given in [8], [9].

5 Conclusions

The paper presents a novel method for working out co-operation for project realizations. The method uses a sophisticated structure of local optimization task. The local criterion contains components responsible for co-operation of executors. The

structure as well as parameters of the task are modified during search process. It is done on a basis of gathering information during previous iterations. Thus, the method is a learning one. The method is based on a general formal model of multistage decision processes (MDP) that is given in the paper.

A large number of difficult combinatorial problems can be efficiently solved by means of the method. Especially, the method is very useful for difficult scheduling problems with state depended resources. Managing projects, especially software projects belong to this class.

To illustrate the conception, some NP-hard problem, namely a scheduling problem with state dependent resources is considered and the learning algorithm for it is presented. Results of computer experiments confirm the efficiency of the algorithm.

References

1. Dudek-Dyduch, E.: Formalization and Analysis of Problems of Discrete Manufacturing Processes, vol. 54. Scientific bulletin of AGH University, Automatics (1990) (in Polish)
2. Dudek-Dyduch, E.: Control of discrete event processes - branch and bound method. In: Proc. of IFAC/Ifors/Imacs Symposium Large Scale Systems: Theory and Applications, Chinese Association of Automation, vol. 2, pp. 573–578 (1992)
3. Dudek-Dyduch, E.: Learning based algorithm in scheduling. Journal of Intelligent Manufacturing 11(2), 135–143 (2000)
4. Dudek-Dyduch, E., Dyduch, T.: Learning algorithms for scheduling using knowledge based model. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1091–1100. Springer, Heidelberg (2006)
5. Dudek-Dyduch, E., Fuchs-Seliger, S.: Approximate algorithms for some tasks in management and economy. System, Modelling, Control 1(7) (1993)
6. Jędrzejowicz, P.: A-Teams and Their Applications. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 36–50. Springer, Heidelberg (2009)
7. Kolish, R., Drexel, A.: Adaptive Search for Solving Hard Project Scheduling Problems. Naval Research Logistics 42 (1995)
8. Kucharska, E.: Application of an algebraic-logical model for optimization of scheduling problems with retooling time depending on system state. PhD thesis (2006) (in polish)
9. Kucharska, E., Dutkiewicz, L.: Heuristic search of solution tree for scheduling problem with parallel machines, vol. 54. Scientific bulletin of AGH University, Automatics (2008) (in Polish)
10. Rajedran, C.: A no wait flow shop scheduling heuristic to minimize makespan. Journal of Operational Research Society 45, 472–478 (1994)
11. Sprecher, A., Kolish, R., Drexel, A.: Semiactive, active and not delay schedules for the resource constrained project scheduling problem. European Journal of Operational Research 80, 94–102 (1993)

Experimental Evaluation of the Agent-Based Population Learning Algorithm for the Cluster-Based Instance Selection

Ireneusz Czarnowski and Piotr Jędrzejowicz

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
{irek,pj}@am.gdynia.pl

Abstract. The aim of the paper is to propose and evaluate an agent-based population learning algorithm generating, by the prototype selection, a representative training dataset of the required size. It is assumed that prototypes are selected from clusters. The process of selection is executed by a team of agents, which execute various local search procedures and cooperate to find-out a solution to the considered problem. Rules for agent cooperation are defined within working strategies. In this paper influence of two different strategies and the population size on performance of the algorithm is investigated.

Keywords: instance selection, population learning algorithm, A-Team.

1 Introduction

The paper focuses on instance selection problem. Instance selection is considered especially useful as a mean to increasing effectiveness of the learning process when the available datasets are large [5, 11, 17, 20]. Finding a small set of representative instances, called patterns, prototypes or reference vectors, for large datasets can result in producing a classification model superior to one constructed from the whole massive dataset. Besides, such an approach may help to avoid working on the whole original dataset all the time [21]. It is also obvious that removing some instances from the training set reduces time and memory complexity of the learning process [12, 19].

Instance selection is also considered as an important step towards increasing effectiveness of the learning process when the available datasets are distributed and when the access to data is limited and costly from the commercial point of view. Selecting the relevant data from distributed locations and then moving only the local patterns can eliminate or reduce the restrictions on a communication bandwidth, reduce the cost of data shipping, and speed up the distributed learning process [12, 13]. The important distributed data mining problem is to establish a reasonable upper bound on the size of the dataset needed for an efficient analysis [18].

Since the instance selection is known to belong to the NP-hard problem class [9], none of the approaches proposed so far (see, for example, [19], [20]) can be considered superior, guaranteeing satisfactory results in terms of the learning error reduction or increased efficiency of the learning process. While there exists an

P. Jędrzejowicz et al.(Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 301–310, 2011.

abundance of methods for finding prototypes (see note in [21]), a method superior in certain domains is inferior in others [11]. Hence, searching for robust and efficient approaches to instance selection is still a lively field of research.

The contribution of the paper is proposing and evaluating through computational experiment an agent-based population learning algorithm generating a representative dataset of the required size. The agent-based population algorithm is based on the assumption that prototypes are selected from clusters. Only one single instance from each cluster is selected.

The number of clusters produced has direct influence on the size of the reduced dataset. Clusters are induced in two stages. At the first stage the initial clusters are produced using the procedure earlier proposed in [5]. The second stage involves merging of clusters obtained at the initial stage. The merging is carried out to obtain the required data compression rate.

From the practical point of view, the problem of cluster merging and instance selection should be solved using approximation algorithms, which is a standard approach for solving the NP-hard optimization problems. Thus, it was decided to design an agent-based population learning algorithm with the cluster merging and instance selection procedures executed by the set of agents cooperating and exchanging information within an asynchronous team of agents (A-Team).

The agent-based population learning algorithm, called ABIS (Agent-Based Instance Selection), was introduced in [6], where the initial results of the algorithm implementation have been discussed. In this paper the proposed algorithm is evaluated with respect to the working strategy, which define how to choose solutions which are forwarded to the optimizing agents for improvement and how to merge the improvement solutions returned by the optimizing agents with the whole population. The paper also reports how the size of the population influences the A-Team performance.

The paper is organized as follows. In Section 2 the instance selection problem is formulated and shortly discussed. The agent-based instance selection algorithm is presented in Section 3. Section 4 provides details on the computational experiment setup and discusses its results. Finally, the last section contains conclusions and suggestions for future research.

2 Problem Overview

The instance selection process aims at identifying and eliminating irrelevant and redundant information, and finding patterns or regularities within certain attributes, allowing to induce the so-called prototypes or reference vectors. A lot of research work confirms that instance selection can play a pivotal role in building robust learning models. Instance selection techniques aim at decreasing the quality of information required to learn a good quality classifiers. It has been also established in [15, 19, 21] that instance selection can result in:

- Reducing space complexity of the classification problem
- Decreasing the required computation time
- Diminishing the size of the formulas or decision and expression trees obtained by an induction algorithm on the reduced, through instance selection, datasets

- Speeding up the knowledge extraction process and the learning process in case data is stored in the separated and physically distributed repositories
- Eliminating or reducing cost of moving all of the data from distributed repositories to a central site when the distributed learning process is carried out.

In this paper the instance selection problem is defined as a problem of finding the optimal subset of prototypes, where the optimal set is defined with respect to the particular induction algorithm.

Based on the above it can be observed that a good set of prototypes has the following properties:

- Firstly, the cardinality of the reduced dataset, consisting of the prototypes, is smaller than the cardinality of the original, non-reduced dataset.
- Secondly, the reduced, through instance selection, dataset assures maximum or acceptable classification quality criterion or criteria with respect to the classifier induced using such reduced dataset.

Thus, the instance selection process can be seen as a mean to achieve one of the following goals:

- Reducing original dataset to maximize classification quality criterion or criteria.
- Finding the smallest set of prototypes that assures an acceptable value of the performance criterion used.

The overview of the instance selection, other techniques for data reduction as well as the overview of methods and algorithms for instance selection can be found in [4].

3 A Cluster-Based Instance Selection Algorithm

An A-Team architecture with the improvement procedures executed by the set of agents has been proposed. The paper deals with an A-Tem in which agents cooperate and exchange information with view to solve instance of the instance selection problem.

3.1 A-Team Concept

The A-Team architecture has been proposed as a problem-solving paradigm that can be easily used to design and implement the proposed population learning algorithm carrying-out the instance reduction tasks. The A-Team concept was originally introduced in [16].

In the discussed population-based multi-agent approach multiple agents search for the best solution using local search heuristics and population based methods. The best solution is selected from the population of potential solutions which are kept in the common memory. Specialized agents try to improve solutions from the common memory by changing values of the decision variables. All agents can work asynchronously and in parallel. During their work agents cooperate to construct, find

and improve solutions which are read from the shared common memory. Their interactions provide for the required adaptation capabilities and for the evolution of the population of potential solutions.

More information on the population learning algorithm with optimization procedures implemented as agents within an asynchronous team of agents (A-Team) can be found in [2]. In [2] also several its implementations are described.

3.2 Agent-Based Population Learning Algorithm for Instance Selection

An A-Team in which agents execute the improvement procedure and cooperate with a view to solve instances of the data reduction problem has been proposed in [6]. The basic assumptions behind the agent-based instance selection (ABIS) approach are following:

- Instances are selected from clusters of instances. Clusters are constructed separately from the set of training instances with the identical class label.
- Prototype instances are selected from clusters through the population-based search carried out by the optimizing agents.
- Clusters are induced in two stages. At the first stage the initial clusters are produced using the procedure based on the similarity coefficient introduced in [5]. The main feature of the clustering procedure run at the first stage is that the number of clusters is determined by the value of the similarity coefficient. The second stage involves merging of clusters obtained at the initial stage. Clusters are merged through the population-based search. The merging is carried-out in case when the number of clusters obtained at the first stage does not assure the required data compression rate. Such case means that the number of clusters obtained at the cluster initialization stage is greater than the upper bound of the number of clusters set by the user. Hence, the solution produced using the similarity coefficient only, is not feasible and must be further improved at the second stage.
- Initially, potential solutions are generated through randomly selecting exactly one single instance from each of the considered clusters (either merged or not merged).
- Each solution from the population is evaluated and the value of its fitness is calculated. The evaluation is carried out by estimating classification accuracy of the classifier, which is constructed using instances (prototypes) indicated by the solution as the training dataset.
- A feasible solution is represented by two data structures: a string and a binary square matrix of bits. A string contains numbers (numeric labels) of instances selected as prototypes. A total length of the string is equal to the number of clusters of potential reference instances. A matrix denotes whether or not clusters, induced at the cluster initialization stage, have been merged with a view to comply with the constraint on the number of reference vector allowed. The properties of the both solution structures have been discussed in details in [6].
- To solve the cluster-based instance selection problem, the following two groups of optimizing agents carrying out different improvement procedures have been implemented. The first group includes the improvement procedures

for instance selection as it was proposed in [5]. These procedures are: the local search with the tabu list for instance selection and the simple local search. The local search with the tabu list for instance selection, modifies a solution by replacing a randomly selected reference instance with some other randomly chosen reference instance thus far not included within the improved solution. The modification takes place providing the replacement move is not on the tabu list. After the modification, the move is placed on the tabu list and remains there for a given number of iterations. The simple local search modifies the current solution either by removing the randomly selected reference instance or by adding some other randomly selected reference instance thus far not included within the improved solution. The second group consists of the optimizing agents responsible for merging clusters. Agents execute a simple local search procedure. The procedure modifies the current solution changing the composition of clusters through transferring a single reference vector from randomly selected merged cluster into another randomly selected cluster in each iteration.

- In each of the above cases solutions that are forwarded to the optimizing agents for the improvement are selected from the population by the working strategy. The working strategy is also an implementation of a set of rules responsible for the process of merging the improvement solutions returned by the optimizing agents with the whole population (for example they may replace random or worst solution).
- If, during the search, an agent successfully improves the solution then it stops and the improved solution is stored in the common memory. Otherwise, agents stop searching for an improvement after having completed the prescribed number of iterations.

4 Computational Experiment

4.1 Computational Experiment Setting

The aim of the computation experiment was to evaluate how the choice of the working strategy can influence the quality of solutions obtained by the A-Team algorithm and how the population size can influence the performance of the cluster-based instance selection algorithm.

It has been decided to implement and evaluate the following working strategies:

- RM-RR – selection of individuals for improvement is a purely randomly move i.e. each optimizing agent receives a solution drawn at random from the population of solutions. A returning individual replaces randomly selected individual in the common memory.
- RM-RW - selection of individuals for improvement is a purely randomly move i.e. each optimizing agent receives a solution drawn at random from the population of solutions. A returning individual replaces first worst individual. Replacement does not take place if there is no worse individual in the common memory.

4.2 Dataset Choice and the Experiment Plan

To validate the proposed approach several benchmark classification problems have been solved. Datasets for each problem including Cleveland heart disease, credit approval, Wisconsin breast cancer, Ionosphere, Hepatitis, Diabetes and Sonar have been obtained from the UCI Machine Learning Repository [1]. Characteristics of these datasets are shown in Table 1.

Each benchmarking problem has been solved 30 times, and the experiment plan involved 3 repetitions of the 10-cross-validation scheme. The reported values of the quality measure have been averaged over all runs. The quality measure in all cases was the correct classification ratio. In the 10-cross-validation scheme, for each fold, the training dataset was reduced using the proposed approach.

Table 1. Datasets used in the reported experiment

Dataset	Number of instances	Number of attributes	Number of classes	The best reported classification accuracy
Heart	303	13	2	90.0% [8]
Cancer	699	9	2	97.5% [1]
Australian credit (ACredit)	690	15	2	86.9% [1]
Sonar	208	60	2	97.1% [1]
Hepatitis	155	19	2	87.13%[10]
German credit (GCredit)	1000	20	2	77.47%[10]
Ionosphere	351	34	2	94.9% [1]
Diabetes	768	8	2	77.34%[10]

The proposed algorithm has been run for the upper bound on the number of the selected prototypes set to t% of the number of instances in the original dataset, where during the experiment t was equal to 10, 15 and 20, respectively.

Population size for each investigated A-Team architecture was set to 20, 40, 60 and 80, respectively.

The process of searching for the best solution has been stopped either after 100 iterations or after there has been no improvement of the current best solution for one minute of computation. Values of these parameters have been set arbitrarily.

Generalization accuracy has been used as the performance criterion. The learning tool was the C4.5 algorithm [14].

The proposed A-Team has been implemented using the middleware environment called JABAT [2], based on JAVA code and built using JADE (Java Agent Development Framework) [3].

4.3 Experiment Results

Table 2 shows mean values of the classification accuracy of the classifiers obtained using the ABIS approach (i.e. using the set of prototypes found by selecting instances and using different implemented working strategies). The results are averaged over all

experiments irrespective of the population size. Table 2 also contains results obtained by machine classification without instance selection and results obtained using the set of prototypes produced through selection based on the k-means clustering (in this case at the first stage the *k*-means clustering has been implemented and next, from thus obtained clusters, the prototypes have been selected using the agent-based population learning algorithm as in [7]).

From Table 2 it can be observed that the best results have been obtained by the ABIS algorithm. The ABIS algorithm assures the required size of the reduced dataset and is competitive to the prototype selection based on the k-means clustering. It

Table 2. Accuracy of the classification results (%)

	Heart	Cancer	ACredit	Sonar	Hep.	GCredit	Ino.	Diab.
C 4.5	77,89	94,57	84,93	74,04	83,87	70,5	91,45	73,82
k-means (*k*=10%)	85,67	94,43	88,99	54,34	77,67	71	87,54	73,1
k-means (*k*=15%)	87,67	95,09	90,14	59,48	79,32	69,8	84,32	70,7
k-means (*k*=20%)	86	96,14	90,29	72,17	79,06	71,42	84,32	71,53
Strategy RM-RR								
ABIS (*k*=10%)	84	97,3	88,55	80,77	79,6	77,2	90,59	80,35
ABIS (*k*=15%)	86,33	**98,86***	**90,69***	**80,77**	**84,04**	78,1	91,45	80,61
ABIS (*k*=20%)	87,26	97,58	89,56	81,73	82,04	**78,7***	92,01	**80,62***
Strategy RM-RW								
ABIS (*k*=10%)	84,00	96,01	88,41	79,81	76,13	77,30	89,44	79,70
ABIS (*k*=15%)	86,00	96,87	88,55	77,88	76,67	78,10	92,29	79,44
ABIS (*k*=20%)	**89,00**	96,44	90,14	79,81	82,58	78,30	**92,59**	79,30

* - Result that outperform the literature-reported classification accuracy.

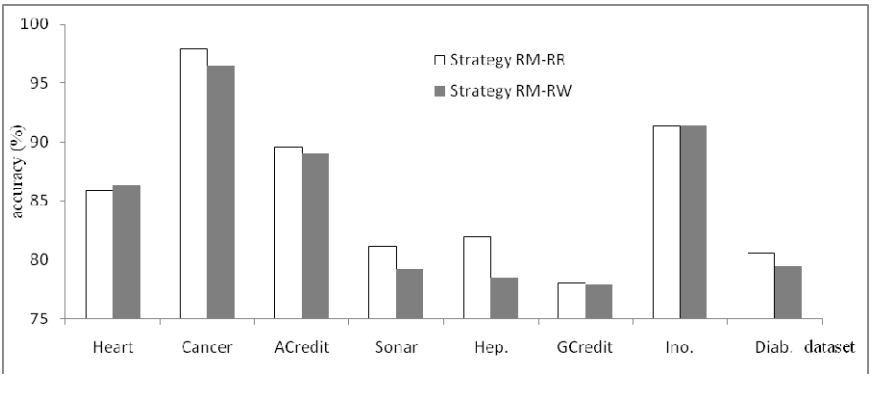


Fig. 1. Average accuracy (%) of the classification for different working strategies

should be also noted that the cluster-based approach produced very good results as compared with the case when the classifier is induced using original, non-reduced dataset.

When the results are compared with respect to the working strategy (see Fig. 1) it can be observed that the RM-RR strategy outperform the RM-RW strategy. Only in case of the heart problem the opposite conclusion is true.

Next question investigated in the experiment was whether population size influences the quality of solutions? To answer this question, for each problem and each strategy, a population size have been ranked: weights (points) have been assigned to different sizes, used in the experiment, with 4 point scale and 1 point given to the worst performing size and 4 to the best. Then the weight have been averaged over the problems. The ranking is shown in Fig.2. It may be noticed that the experiment does not allow to draw any conclusions with respect to the impact of the population size on results achieved. Besides, it can be in most cases RM-RR outperforms RM-RW.

To reinforce the above observation, the experiment results have been used to perform the two-way analysis of variance. The following null hypothesis were formulated:

I. The choice of the working strategy does not influence the performance of the A-Team algorithm.
II. The choice of the population size does not influence the performance of the cluster-based instance selection algorithm.
III. There are no interactions between both factors (i.e. the choice of the working strategy and the choice of the population size).

The two-way ANOVA has been carried out independently for each dataset and it was established, with the degree of confidence set at 95%, that all null hypothesis should be rejected. Thus the choice of the working strategy and the choice of the population size influence the performance of the cluster-based instance selection algorithm.

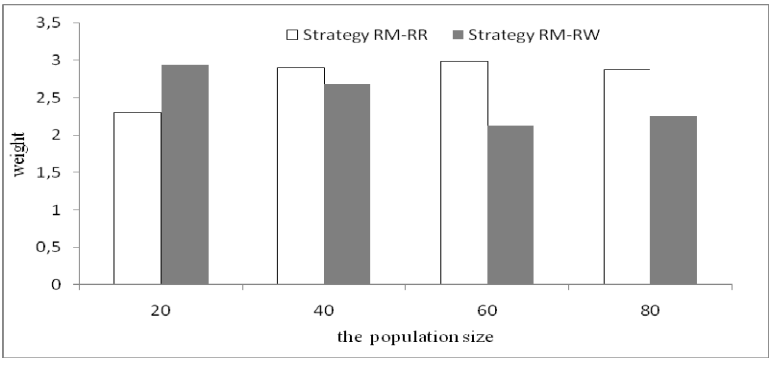


Fig. 2. The weights for different working strategies and for different population sizes within the A-Team averaged over classification problems

5 Conclusions

The paper proposes an agent-based population learning algorithm generating, by the prototype selection, a representative training dataset of the required size. The A-Team algorithm is experimentally evaluated. The presented research confirms the importance of choosing an effective working strategy when employing A-Team searching for solutions of the instance selection problem. The second confirmed observation is that choosing the population size influences the performance of cluster-based instance selection. The statistical analysis confirms the interaction between both factors. The reported experiment also shows that in a majority of cases working strategy based on selection with random moves and random replacement assures the best reduced datasets.

Finding more effective working strategies should be the focus of future research, as well as establishing general rules for finding the optimal configuration of the A-Team carrying out the instance selection problem.

Acknowledgments. This research has been supported by the Polish Ministry of Science and Higher Education with grant no. N N519 576438 for years 2010-2013.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: e-JABAT - An Implementation of the Web-Based A-Team. In: Nguyen, N.T., Jain, I.C. (eds.) Intelligent Agents in the Evolution of Web and Applications. SCI, vol. 167, pp. 57–86. Springer, Heidelberg (2009)
3. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper, Exp. 3(3), 6–20 (2003)
4. Czarnowski, I.: Distributed Learning with Data Reduction. In: Nguyen, N.T. (ed.) Transactions on CCI IV. LNCS, vol. 6660, pp. 1–2. Springer, Heidelberg (2011)
5. Czarnowski, I., Jedrzejowicz, P.: An Approach to Instance Reduction in Supervised Learning. In: Coenen, F., Preece, A., Macintosh, A. (eds.) Research and Development in Intelligent Systems XX, pp. 267–282. Springer, London (2004)
6. Czarnowski, I., Jedrzejowicz, P.: A New Cluster-based Instance Selection Algorithm. In: O'Shea, J., Nguyen, N.T., Crockett, K., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2011. LNCS (LNAI), vol. 6682, pp. 436–445. Springer, Heidelberg (2011)
7. Czarnowski, I., Jedrzejowicz, P.: Cluster Integration for the Cluster-based Instance Selection. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS (LNAI), vol. 6421, pp. 353–362. Springer, Heidelberg (2010)
8. Datasets used for classification: comparison of results. In. directory of data sets, http://www.is.umk.pl/projects/datasets.html (accessed September 1, 2009)
9. Hamo, Y., Markovitch, S.: The COMPSET Algorithm for Subset Selection. In: Proceedings of The Nineteenth International Joint Conference for Artificial Intelligence, Edinburgh, Scotland, pp. 728–733 (2005)

10. Jedrzejowicz, J., Jedrzejowicz, P.: Cellular GEP-Induced Classifiers. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS (LNAI), vol. 6421, pp. 343–352. Springer, Heidelberg (2010)
11. Kim, S.-W., Oommen, B.J.: A Brief Taxonomy and Ranking of Creative Prototype Reduction Schemes. Pattern Analysis Application 6, 232–244 (2003)
12. Klusch, M., Lodi, S., Moro, G.-L.: Agent-Based Distributed Data Mining: The KDEC Scheme. In: Klusch, M., Bergamaschi, S., Edwards, P., Petta, P. (eds.) Intelligent Information Agents. LNCS (LNAI), vol. 2586, pp. 104–122. Springer, Heidelberg (2003)
13. Krishnaswamy, S., Zaslavsky, A., Loke, S.W.: Techniques for Estimating the Computation and Communication Costs of Distributed Data Mining. In: Sloot, P.M.A., Tan, C.J.K., Dongarra, J., Hoekstra, A.G. (eds.) ICCS-ComputSci 2002. LNCS, vol. 2329, pp. 603–612. Springer, Heidelberg (2002)
14. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, SanMateo (1993)
15. Silva, J., Giannella, C., Bhargava, R., Kargupta, H., Klusch, M.: Distributed Data Mining and Agents. Engineering Applications of Artificial Intelligence Journal 18, 791–807 (2005)
16. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Co-operation Schemes for Autonomous, Computer-Based Agents. Technical Report EDRC 18-59-96, Carnegie Mellon University, Pittsburgh (1996)
17. Uno, T.: Multi-sorting Algorithm for Finding Pairs of Similar Short Substrings from Large-scale String Data. Knowledge and Information Systems (2009), doi:10.1007/s10115-009-0271-6
18. Vucetic, S., Obradovic, Z.: Performance Controlled Data Reduction for Knowledge Discovery in Distributed Databases. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 29–39 (2000)
19. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-based Learning Algorithm. Machine Learning 33(3), 257–286 (2000)
20. Yu, K., Xu, X., Ester, M., Kriegel, H.-P.: Feature Weighting and Instance Selection for Collaborative Filtering: An Information-Theoretic Approach. Knowledge and Information Systems 5(2), 201–224 (2004)
21. Zhu, X., Wu, X.: Scalable Representative Instance Selection and Ranking. In: IEEE Proceedings of the 18th International Conference on Pattern Recognition, vol. 3, pp. 352–355 (2006)

Double-Action Agents Solving the MRCPSP/Max Problem

Piotr Jędrzejowicz and Ewa Ratajczak-Ropel

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
{pj,ewra}@am.gdynia.pl

Abstract. In this paper an A-Team architecture for solving the multi-mode resource-constrained project scheduling problem with minimal and maximal time lags (MRCPSP/max) is proposed and experimentally validated. To solve this problem an asynchronous team of agents implemented using JABAT middleware has been proposed. Four kinds of optimization agent has been used. Each of them acts in two ways depending whether the received initial solution is feasible or not. The paper contains the MRCPSP/max problem formulation, description of the proposed architecture for solving the problem instances, description of optimization algorithms, description of the experiment and the discussion of the computational experiment results.

Keywords: project scheduling, multi-mode resource-constrained project scheduling, MRCPSP, optimization, agent, agent system, A-Team.

1 Introduction

The paper proposes an agent-based approach to solving instances of the MRCPSP/max, known also in the literature as the MRCPSP-GPR problem. MRCPSP stands for the Multi-mode Resource-Constrained Project Scheduling Problem, max or GPR is used to describe precedence relations as minimal and maximal time lags, also called Generalised Precedence Relations (GPR) or temporal constraints or time windows. MRCPSP/max has attracted a lot of attention and many exact and heuristic algorithms have been proposed for solving it (see for example [12], [4], [5], [10], [3]).

MRCPSP/max is a generalisation of RCPSP/max and thus it is NP-hard [2]. The approaches to solve this problem produce either approximate solutions or can be applied for solving instances of the limited size. Hence, searching for more effective algorithms and solutions to the MRCPSP/max problem is still a lively field of research. One of the promising directions of such research is to take advantage of the parallel and distributed computation solutions, which are the feature of the contemporary multiple-agent systems.

The multiple-agent systems are an important and intensively expanding area of research and development. There is a number of multiple-agent approaches proposed to solve different types of optimization problems. One of them is the

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 311–321, 2011.

concept of an asynchronous team (A-Team), originally introduced in [13]. The A-Team paradigm was used to develop the JADE-based environment for solving a variety of computationally hard optimization problems called E-JABAT [1]. E-JABAT is a middleware supporting the construction of the dedicated A-Team architectures based on the population-based approach.

In this paper the E-JABAT-based A-Team architecture for solving the MRCPSP/max problem instances is proposed and experimentally validated. A-Team includes optimization agents which represent heuristic algorithms. The proposed approach is an extension and improvement of the A-Team described in [8]. A new kind of optimization agent was added and all agents were redefined in order to act in two ways depending on feasibility or unfeasibility of the initial solution.

Section 2 of the paper contains the MRCPSP/max problem formulation. Section 3 provides details of the E-JABAT architecture implemented for solving the MRCPSP/max problem instances. In section 4 the computational experiment is described. In section 5 the computational experiment results are presented. Section 6 contains conclusions and suggestions for future research.

2 Problem Formulation

In the multi-mode resource-constrained project scheduling problem with minimal and maximal time lags (MRCPSP/max) a set of $n+2$ activities $V = \{0, 1, \ldots, n, n+1\}$ is considered. Each activity has to be processed without interruption to complete the project. The dummy activities 0 and $n+1$ represent the beginning and the end of the project. For each activity $i \in V$ a set $M_i = \{1, ..., |M_i|\}$ of (execution) modes is available. Activities 0 and $n+1$ can be performed in only one mode: $M_0 = M_{n+1} = \{1\}$. Each activity $i \in V$ has to be performed in exactly one mode $m_i \in M_i$. The duration (processing time) of an activity i, $i \in V$ executed in m_i mode is denoted by d_{im_i}, $d_{im_i} \in Z_{\geq 0}$. The processing times of activity 0 and $n+1$ equals 0, i.e. $d_{00} = d_{n+1\,0} = 0$.

S_i and C_i stand for the start time and the completion time (of the performance) of activity i, respectively. If we define $S_0 = 0$, S_{n+1} stands for the project duration. Provided that activity i starts in mode m_i at time S_i, it is being executed at each point in time $t \in [S_i, S_i + d_{im_i})$.

Between the start time S_i of activity i, which is performed in mode $m_i \in M_i$, and the start time S_j of activity j ($i \neq j$), which is performed in mode $m_j \in M_j$, a minimum time lag $d^{min}_{im_i,jm_j} \in Z_{\geq 0}$ or a maximum time lag $d^{max}_{im_i,jm_j} \in Z_{\geq 0}$ can be given. Note, that a time lag between activity i and activity j depends on mode m_i as well as on mode m_j.

Activities and time lags are represented by an activity-on-node (AoN) network $N = \langle V, E, \delta \rangle$ with node set V, arc set E, and arc weight function δ. Each element of node set V represents an activity. In the following, we do not distinguish between an activity and the corresponding node. An arc $\langle i, j \rangle \in E$ indicates that a time lag between S_i and S_j has to be observed. Arc weight function δ assigns to each arc $\langle i, j \rangle \in E$ a $|M_i| \times |M_j|$ matrix of arc weights as follow: for a minimum time lag $d^{min}_{im_i,jm_j}$ we set $\delta_{im_i,jm_j} = d^{min}_{im_i,jm_j}$, and for a maximum time lag $d^{max}_{im_i,jm_j}$ we set $\delta_{im_i,jm_j} = -d^{max}_{im_i,jm_j}$.

There are the set of renewable resources R^R and the set of nonrenewable resources R^N considered in this problem, $|R^R|, |R^N| \in Z_{>0}$. The availability of each renewable resource type $k \in R^R$ in each time period is R_k^R units. The availability of each nonrenewable resource type $k \in R^N$ is R_k^N units in total. Provided that activity i is performed in mode m_i, $r_{im_ik}^R$ units of renewable resource $k \in R^R$ are used at each point in time at which activity i is being executed. Moreover, $r_{im_ik}^N$ units of nonrenewable resource $k \in R^N$ are consumed in total. For activities 0 and $n+1$ we set $r_{01k} = r_{n+11k} = 0$ for $k \in R^R$ and $r_{01k}^N = r_{n+10k}^N = 0$ for $k \in R^N$.

The solution of the problem is a schedule (M, S) consisting of the mode vector M and a vector of activities starting times $S = [S_0, \ldots, S_{n+1}]$, where $S_0 = 0$ (project always begins at time zero). The mode vector assigns to each activity $i \in V$ exactly one mode $m_i \in M_i$ - execution mode of activity i. The start time vector S assigns to each activity $i \in V$ exactly one point in time as start time S_i where $S_0 = 0$ (project always begins at time zero). Precedence relations are described by the following formula: $S.S_j - S.S_i \geq \delta_{im_i,jm_j}$, where $\langle i, j \rangle \in E$.

The objective is to find a schedule (M, S) where precedence and resource constraints are satisfied, such that the schedule duration $T(S) = S_{n+1}$ is minimized. The detailed description of the problem can be found in [5] or [3]. The MRCPSP/max, as an extension of the RCPSP and RCPSP/max, belongs to the class of NP-hard optimization problems [2].

3 E-JABAT for Solving the MRCPSP/Max Problem

E-JABAT is a middleware allowing to design and implement A-Team architectures for solving various combinatorial optimization problems implemented using JADE framework. Detailed information about E-JABAT and its implementations can be found in [6] and [1]. E-JABAT environment was successfully used for solving the RCPSP, MRCPSP and RCPSP/max problems [7]. In the proposed approach the agents, classes describing the problem and ontologies have been implemented for solving the discussed problem. The above forms the package called JABAT.MRCPSPmax.

Classes describing the problem are responsible for reading and preprocessing the data and generating random instances of the problem. The discussed set includes the following classes:

- MRCPSPmaxTask inheriting from the Task class and representing the instance of the problem,
- MRCPSPmaxSolution inheriting from the Solution class and representing the solution of the problem instance,
- Activity, Mode, Resource, PredSuccA, PredSuccM and PredSuccT representing the activity of the problem and its parameters.

The next set includes classes allowing for definition of the vocabulary and semantics for the content of messages exchange between agents. In the proposed approach the messages include all data representing the task and solution:

– MRCPSPmaxTaskOntology inheriting from the TaskOntology class,
– MRCPSPmaxSolutionOntology inheriting from the SolutionOntology class,

The last set includes classes describing the optimization agents. Each of them includes the implementation of an optimization heuristic used to solve the problem. All of them are inheriting from OptiAgent class. The set includes:

– optiCA denoting Crossover Algorithm (DA_CA),
– optiLSA denoting Local Search Algorithm (DA_LSA),
– optiPRA denoting Path Relinking Algorithm (DA_PRA),
– optiTSA denoting Tabu Search Algorithm (DA_TSA),

In [8] an A-Team and CA, LSA and PRA algorithms to solving the MRCPSP/max problem have been proposed. In this paper the algorithms were modified to perform double-action and a new optimization agent TSA representing tabu search metaheuristic has been added.

The double-action optimization agents (DA_) use their algorithms to solve MRCPSP/max problem instances. Each optimization agent needs one (in the case of DA_LSA and DA_TSA) or two (DA_CA and DA_PRA) initial solutions. In many instances of MRCPSP/max problem it is difficult to generate a population of feasible solutions or even finding a single feasible solution. On the other hand the proposed CA, LSA, PRA and TSA algorithms are seldom effective when used with unfeasible solutions. Main reason of the above difficulty is a presence of cycles in the respective AoN network. The proposed double-action agents deal with the problem through applying the standard optimization procedure if, at least one feasible initial solution has been found. If not the second action is initiated additionally. An agent tries to find a schedule with the minimal cycle time where cycle time is calculated as the sum of elements over the diagonal of the longest path matrix. The matrix is calculated using Floyd-Warshall triple algorithm described in [9].

The objective of the proposed approach is to find best feasible schedule (M, S). The procedure of finding a new solution from the schedule (M, S) is based on the SGSU (Serial Generation Scheme with Unscheduling) described in [9] with several different priority rules.

Pseudo-codes of the algorithms are shown in Figures 1, 2, 3, 4, respectively. In pseudo-codes S denotes the schedule (M, S) with ordered list of activities. The solution is calculated using procedure based on SGSU. The objective functions used are as follow:

```
int objectiveFunctionS(S) { return S.S_{n+1} }
int objectiveFunctionF(S) { return cycles time for S }
```

The LSA (Figure 1) is a simple local search algorithm which finds local optimum by moving each activity with each possible mode to all possible places in the schedule. For each combination of activities the value of possible solution is calculated. The best schedule is returned.

The TSA (Figure 2) is an implementation of the tabu search metaheuristic. In a schedule the pairs of activities and simultaneously modes of these activities

```
DA_LSA(initialSchedule)
{  S = initialSchedule
   if(S is not feasible)
      bestS =LSA(S,startActPosF,itNumF,stepF,objectiveFunctionF)
   bestS =LSA(S,startActPos,itNum,step,objectiveFunctionS)
}

LSA(S,startActivityPosition,iterationNumber,step,objectiveFunction)
{  it = iterationNumber
   bestS = S
   while(it>0)
   {  bestSit = S
      for(all activities in S)
      {  pi = startActivityPosition
         for(all modes in activity from position pi in S)
         {  move activity from position pi to pi + step and change its
               active mode
            if(S is better then bestSit due to objectiveFunction)
               bestSit = S
            reverse last move
         }
      }
      if(bestSit is better then bestS due to objectiveFunction)
      {  bestS = bestSit
         it = iterationNumber
      }
      else it-
   }
   return bestS
}
```

Fig. 1. Pseudo-codes of the DA_LSA and LSA algorithms

are changed. Selected moves are remembered in a tabu list. For example, after changing two activities any following moves of each of them are blocked for 10% of iteration, moves chanaging the same two activities are blocked for 70% of iteration, etc. The best schedule is remembered and finally returned.

The CA (Figure 3) is an algorithm based on the idea of the one point crossover operator. For a pair of solutions one point crossover is applied. The *step* argument determines the frequency the operation is performed.

The PRA (Figure 4) is an implementation of the path-relinking algorithm. For a pair of solutions a path between them is constructed. The path consists of schedules obtained by carrying out a single move from the preceding schedule. The move is understood as moving one of the activities to a new position simultaneously changing its mode. For each schedule in the path the value of the respective solution is checked. The best schedule is remembered and finally returned.

```
DA_TSA(initialSchedule)
{  S = initialSchedule
   if(S is not feasible)
      bestS =TSA(S,startActPosF,itNumF,stepF,objectiveFunctionF)
   bestS =TSA(S,startActPos,itNum,step,objectiveFunctionS)
}

TSA(S,iterationNumber,step,objectiveFunction)
{  TabuList = ∅
   it = iterationNumber; bestS = S
   while(it>0)
   {  bestSit = S
      for(all activities in S)
         for(all modes in activity from position pi in S)
            for(all modes in activity from position pi + step in S)
            {  move = change activities from position pi and pi + step and
                  simultaneously change theirs active modes
               if(move is not in TabuList)
               {  make move
                  if(S is better then bestSit due to objectiveFunction)
                     bestSit = S
                  reverse last move
               }
            }
      update TabuList
      if(bestSit is not worse than bestS due to objectiveFunction)
      {  bestS = bestSit
         it = iterationNumber
         add selected moves to TabuList
      }
      else it-
   }
   return bestS
}
```

Fig. 2. Pseudo-codes of the DA_TSA and TSA algorithms

All optimization agents co-operate together using the E-JABAT common memory. The initial population in the common memory is generated randomly with the exception of a few individuals which are generated by heuristics based on the priority rules [9] and procedure based on SGSU. Because it is difficult to obtain feasible solution for some MRCPSP/max problem instances, the random drawing of an individual could be repeated several times. If this does not produce enough feasible solutions the infeasible ones are added to the population in the common memory. In some instances the initial population consist of the infeasible solutions only. Individuals in the common memory are represented as (M,S). The final solution is obtained from the schedule by the procedure based on SGSU.

```
DA_CA(initialSchedule1,initialSchedule2,step)
{  S1 = initialSchedule1;  S2 = initialSchedule2
   betterS = best from S1 and S2 due to objectiveFunctionS
   if(S1 is not feasible and S2 is not feasible)
      bestS =CA(S1,S2,stepF,objectiveFunctionF)
   bestS =CA(bestS,betterS,step,objectiveFunctionS)
}

CA(S1,S2,step,objectiveFunction)
{  S = best from S1 and S2 due to objectiveFunction
   for(all crossover points every step activities in S)
   {  S = result of one point crossover operation on S1 and S2
      if(objectiveFunction = objectiveFunctionF)
         for(all modes of activities in crossover point in S1 and S2
            S = best schedule due to objectiveFunctionF
      if(S is better than bestS due to objectiveFunction) bestS = S
   }
   return bestS
}
```

Fig. 3. Pseudo-codes of the DA_CA and CA algorithms

```
DA_PRA(initialSchedule1,initialSchedule2,step)
{  S1 = initialSchedule1;  S2 = initialSchedule2
   betterS = best from S1 and S2 due to objectiveFunctionS
   if(S1 is not feasible and S2 is not feasible)
      bestS =PRA(S1,S2,objectiveFunctionF)
   bestS =PRA(bestS,betterS,objectiveFunctionS)
}

PRA(S1,S2,objectiveFunction)
{  S = S1
   for(all activities in S)
   {  in S find position of activity from position pi in S2
      in S move found activity to position pi
      for(all modes of found activity)
         S = best schedule due to objectiveFunction
      if(S is better than bestS due to objectiveFunction) bestS = S
   }
   return bestS
}
```

Fig. 4. Pseudo-codes of the DA_PRA and PRA algorithms

The time and frequency an agent of each kind receives a solution or set of solutions from the common memory with a view to improve its quality is determined by the strategy. For solving the MRCPSP/max problem instances the strategy with blocking has been used where individuals forwarded to optimization agents for improvement are randomly chosen from the population stored in

the common memory. Such individuals are sent to optimization agents ready to start searching for a better solution. After computation the improved individual replaces the one which was send. Additionally, if some solutions (in this approach 5) are received but the best solution in the population has not been improved a new one is generated randomly. It replaces the worst one in the population.

4 Computational Experiment

To validate the proposed approach and to evaluate the effectiveness of the optimization agents the computational experiment has been carried out using benchmark instances of MRCPSP/max from PSPLIB [11] - test set mm100 with activities carried out in 3, 4 and 5 modes. The set includes 270 problem instances. The experiment involved computation with the fixed number of optimization agents representing DA_CA, DA_LSA, DA_PRA and DA_TSA algorithms, fixed population size, and the limited time period allowed for computation.

The discussed results have been obtained using 8 optimization agents - two of each kind. Population of solutions in the common memory consisted of 25 individuals. The computation has been stopped after 3 minutes if no better solution is found. The optimization algorithms have had the fixed and randomly chosen parameter values. For example, in the case of LSA the $startActivityPosition$ has been chosen randomly from the interval $[1, step]$. The $iterationNumber$ and $step$ parameters in the LSA algorithm has been fixed to n and 1 if the $objectiveFunctionF$ has been used and to 10 and 2 if the $objectiveFunctionS$ has been used. The same parameters in the TSA algorithm has been fixed to 100 and 1 if the $objectiveFunctionF$ has been used and to 10 and 1 if the $objectiveFunctionS$. In the case of CA the $step$ parameter has been set to 1.

Experiment has been carried out using nodes of the cluster Holk of the TASK Academic Computer Network. During the computation one node per two agents was used.

5 Computational Experiment Results

During the experiment the following characteristics have been calculated and recorded: mean relative error (Mean RE) calculated as the deviation from the lower bound, percent of feasible solutions (% FS) and mean computation time required to find the best solution (Mean CT). Each instance has been solved five times and the results have been averaged over these solutions. The results are presented in Table 1. These results are compared with the results reported in the literature obtained by tabu search algorithm TS_{DR} and heuristic based on multipass priority-rule method with backplanning Prio proposed in [4].

The influence of each agent performance on the results was evaluated as the average percent of individuals which were improved by it and the average percent of the current best solutions found by it. The results in terms of the percent of solutions improved by each agent, percent of solutions improved by any agent and percent of non-improved solutions are presented in Table 2.

Table 1. Performance characteristics of the proposed A-Team and literature reported results for benchmark test set mm100

#Modes	Mean RE	% FS	Mean CT [s]
Proposed A-Team with double-action agents			
3	57.2%	98.9%	543.2
4	85.4%	97.8%	393.1
5	102.1%	98.9%	257.6
Literature reported results – TS_{DR} [4]			
3	40%	53%	100
4	91%	61%	100
5	164%	67%	100
Literature reported results – Prio [4]			
3	63%	100%	100
4	113%	100%	100
5	170%	100%	100

Table 2. Percentage of the improved and best improved solutions obtained by the agents in the proposed approach

#Modes	CA	LSA	PRA	TSA	Any	No one
Percents of improved solutions						
3	96%	98%	99%	97%	99%	1%
4	92%	92%	98%	82%	98%	2%
5	82%	73%	94%	37%	98%	2%
Percents of best improved solutions						
3	35%	82%	82%	76%	94%	6%
4	27%	71%	76%	41%	89%	11%
5	20%	39%	63%	9%	74%	26%

Experiment results show that the proposed approach is effective. The obtained results are better than in the previous implementation [8]. The feasible solutions are found for 98.5% of instances (in the previous approach for 88.7%). Presented results are comparable with solutions known from the literature.

Evaluating particular agents, it can be noted that their contributions towards improving the current best solution have been quite similar. However, the effectiveness of TSA and also LSA lowers significantly in case of instances with more modes. The most effective agent, PRA can find best solutions in the population for 63% to 82% of instances. The less effective CA finds best solution for 20% to 35% of instances. The TSA is quite effective (76%) for instances with activities with 3 modes but very poor (only 9%) for instances with activities with 5 modes.

Considering effectiveness of agents, it should be noted that the most effective one is the PRA. However the experiment results show that other agents could find better solutions for different individuals then PRA, so co-operation of agents becomes effective.

6 Conclusions

Experiment results show that the proposed E-JABAT based A-Team implementation is an effective tool for solving instances of the MRCPSP/max problem. Presented results are comparable with solutions known from the literature and better than in the previous implementation.

Future research will concentrate on finding the best configuration of the heterogenous agents and parameters for the algorithms. It is interesting which agents should or should not be replicated to improve the results. Additionally, testing and adding to E-JABAT more different optimization agents and improving the existing ones could be considered. The other possibility is finding and testing different or additional objective functions for MRCPSP/max problem which could be used in the algorithms.

Acknowledgments. The research has been supported by the Ministry of Science and Higher Education grant no. N N519 576438. Calculations have been performed in the Academic Computer Centre TASK in Gdansk.

References

1. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: E-JABAT An Implementation of the Web-Based A-Team. In: Nguyen, N.T., Jain, L.C. (eds.) Intelligent Agents in the Evolution of Web and Applications, pp. 57–86. Springer, Heidelberg (2009)
2. Bartusch, M., Mohring, R.H., Radermacher, F.J.: Scheduling Project Networks with Resource Constraints and Time Windows. Annual Operational Research 16, 201–240 (1988)
3. Barrios, A., Ballestin, F., Valls, V.: A double genetic algorithm for the MRCPSP/max. Computers and Operations Research 38, 33–43 (2011)
4. Heilmann, R.: Resource-Constrained Project Scheduling: a Heuristic for the Multi-Mode Case. OR Spektrum 23, 335–357 (2001)
5. Heilmann, R.: A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. European Journal of Operational Research 144, 348–365 (2003)
6. Jędrzejowicz, P., Wierzbowska, I.: JADE-Based A-Team Environment. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
7. Jedrzejowicz, P., Ratajczak-Ropel, E.: Solving the rCPSP/max problem by the team of agents. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS (LNAI), vol. 5559, pp. 734–743. Springer, Heidelberg (2009)
8. Jędrzejowicz, P., Ratajczak-Ropel, E.: A-Team for Solving MRCPSP/max Problem. Accepted for publication in LNCS/LNAI series (2011)
9. Neumann, K., Schwindt, C., Zimmermann, J.: Project Scheduling with Time Windows and Scarce Resources, 2nd edn. Springer, Heidelberg (2003)
10. Nonobe, K., Ibaraki, T.: A Tabu Search Algorithm for a Generalized Resource Constrained Project Scheduling Problem. In: MIC 2003: The Fifth Metaheuristics International Conference, pp. 55-1–55-6 (2003)

11. PSPLIB, http://129.187.106.231/psplib
12. Reyck, B., Herroelen, W.: The Multi-Mode Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. European Journal of Operational Research 119, 538–556 (1999)
13. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Cooperation Schemes for Autonomous, Computer-Based Agents. Technical Report EDRC 18-59-96. Carnegie Mellon University, Pittsburgh (1996)

Parallel Cooperating A-Teams

Dariusz Barbucha, Ireneusz Czarnowski, Piotr Jędrzejowicz,
Ewa Ratajczak-Ropel, and Izabela Wierzbowska

Department of Information Systems, Gdynia Maritime University,
Morska 81-87, 81-225 Gdynia, Poland
{barbucha,irek,pj,ewra,iza}@am.gdynia.pl

Abstract. The paper proposes and evaluates an architecture for solving difficult combinatorial optimization problems in which several A-Teams work in parallel and cooperate. The proposed approach has been implemented to solve instances of several computationally hard combinatorial optimization problems. Computational experiment carried-out using the available benchmark datasets, has confirmed that the proposed architecture may be competitive in terms of the quality of solutions in comparison with using traditional, non-cooperating, teams of agents.

Keywords: Parallel A-Teams, agent cooperation, combinatorial optimization.

1 Introduction

As it has been observed in [1] the techniques used to solve difficult combinatorial optimization problems have evolved from constructive algorithms to local search techniques, and finally to population-based algorithms. In recent years, technological advances have enabled development of various parallel and distributed versions of the population based methods. At the same time, as a result of convergence of many technologies within computer science such as object-oriented programming, distributed computing and artificial life, the agent technology has emerged. An agent is understood here as any piece of software that is designed to use intelligence to automatically carry out an assigned task, mainly retrieving, processing and delivering information.

Paradigms of the population-based methods and multiple agent systems have been during mid nineties integrated within the concept of the asynchronous team of agents (A-Team). A-Team is a multi agent architecture, which has been proposed in several papers of S.N. Talukdar and co-authors [10], [11], [12], [13].

The middleware platforms supporting implementation of A-Teams are represented by the JADE-Based A-Team environment (JABAT). Its subsequent versions and extensions were proposed in [2], [5] and [6]. The JABAT middleware was built with the use of JADE (Java Agent Development Framework), a software framework proposed by TILAB [3]. JABAT complies with the requirements of the next generation A-Teams which are portable, scalable and in conformity with the FIPA standards. To solve a single task (i.e. a single problem instance) JABAT uses a population of solutions that are improved by optimizing agents which represent

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 322–331, 2011.

different optimization algorithms. In traditional A-Teams agents work in parallel and independently and cooperate only indirectly using a common memory containing population of solutions.

In [7] JABAT environment has been extended through integrating the team of asynchronous agent paradigm with the island-based genetic algorithm concept first introduced in [4]. A communication, that is information exchange, between cooperating A-Teams has been introduced. It has been shown that in experiments with the Euclidean planar travelling salesman problem (EPTSP) a noticeable improvement in the quality of the computation results has been achieved. In this paper more problems are examined.

The paper is constructed as follows: Section 2 describes *TA-Teams* which implement specialized Team of A-Teams designed to solve instances of different problems. Section 3 gives details of the experiment settings and Section 4 contains a discussion of the experiment results. Finally some conclusions and suggestions for future research are drawn.

2 Team of A-Teams Implementation and Settings

JABAT middleware environment can be used to implement A-Teams producing solutions to optimization problems using a set of optimizing agents, each representing an improvement algorithm. Such an algorithm receives a solution and attempts to improve it. Afterwards, successful or not, the result is send back to where it came from. The process of solving a single task (that is an instance of the problem at hand) consists of several steps. At first the initial population of solutions is generated and stored in the common memory. Individuals forming the initial population are, at the following computation stages, improved by independently acting agents (called optimization agents), each executing an improvement algorithm, usually problem dependent. Different improvement algorithms executed by different agents supposedly increase chances for reaching the global optimum. After a number of reading, improving and storing back cycles, when the stopping criterion is met, the best solution in the population is taken as the final result.

A typical JABAT implementation allows for running a number of A-Teams in parallel providing the required computational resources are available, however the teams never communicate and produce results independently. The implementation of Teams of A-Teams allows for a number of A-Teams to solve the same task by exploring different regions of the search space, with the added process of communication, that makes it possible to exchange some solutions between common memories maintained by each of the A-Teams with a view to prevent premature convergence and assure diversity of individuals. Similar idea of carrying out the evolutionary process within subpopulations before migrating some individuals to other islands and then continuing the process in cycles involving evolutionary processes and migrations was previously used in, for example, [14] or [17].

In *TA-Teams* JABAT implementation the process of communication between common memories is supervised by a specialized agent called *MigrationManager* and defined by a number of parameters including:

- *Migration size* – number of individuals sent between common memories of A-Teams in a single cycle
- *Migration frequency* - length of time between migrations
- *Migration topology* – an architecture in which an A-Team receives communication from another A-Team and sends communication to some other A-Team
- *Migration policy* – a rule determining how the received solution is incorporated into a common memory of the receiving A-Team.

2.1 Working Strategy

The process of solving a single task in JABAT by an A-Team is controlled by the, so called, working strategy understood as a set of rules applicable to managing and maintaining the common memory. It contains a population of solutions called individuals. All individuals are feasible solutions of the instance to be solved.

A-Teams in *TA-Teams* follow the working strategy known as RB-RE which before was identified as generating solutions of a good quality [1]. In this strategy:

- All individuals in the initial population of solutions are generated randomly.
- Selection of individuals for improvement is a random move, however once selected individual (or individuals) can not be selected again until all other individuals have been tried.
- Returning individual replaces the first found worse individual. If a worse individual can not be found within a certain number of reviews (review is understood as a search for the worse individual after an improved solution is returned) then the worst individual in the common memory is replaced by the randomly generated one, representing a feasible solution. The number of reviews after which a random solution is generated equals 5.
- The computation time of a single A-Team is defined by the *no improvement time gap* = 2 minutes. If in this time gap no improvement of the current best solution has been achieved, the A-Team stops computations. Then all other A-Teams solving the same task stop as well, regardless of recent improvements in their best solutions.

The overall best result from common memories of all A-Teams in *TA-Teams* is taken as the final solution found for the task.

3 Computational Experiment Design

3.1 Problems

The impact of implementing the distributed population-based approach has been assessed by comparing results obtained from solving some well-known combinatorial optimization problems by respective *TA-Teams*. The following combinatorial optimization problems have been selected to be a part of the experiment:

- Euclidean planar traveling salesman problem (EPTSP).
- Vehicle routing problem (VRP).

- Clustering problem (CP).
- Resource-constrained project scheduling problem (RCPSP).

The euclidean planar traveling salesman problem is a particular case of the general TSP. Given n cities (points) in the plane and their Euclidean distances, the problem is to find the shortest TSP-tour, i.e. a closed path visiting each of the n cities exactly once.

The vehicle routing problem can be stated as the problem of determining optimal routes through a given set of locations (customers) and defined on a directed graph $G=(V, E)$, where $V = \{0, 1, \ldots, N\}$ is the set of nodes and E is the set of edges. Node 0 is a depot with NV identical vehicles of capacity W. Each other node $i \in V\backslash\{0\}$ denotes a customer with a non-negative demand d_i. Each link $(i,j) \in E$ denotes the shortest path from customer i to j and is described by the cost c_{ij} of travel from i to j ($i,j = 1, \ldots, N$). It is assumed that $c_{ij} = c_{ji}$. The goal is to find vehicle routes which minimize total cost of travel (or travel distance) such that each route starts and ends at the depot, each customer is serviced exactly once by a single vehicle and the total load on any vehicle associated with a given route does not exceed vehicle capacity.

The clustering problem can be defined as follows. Given a set of N data objects, partition the data set into K clusters, such that similar objects are grouped together and objects with different features belong to different groups. Clustering arbitrary data into clusters of similar items presents the difficulty of deciding what similarity criterion should be used to obtain a good clustering. It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Euclidean distance and squared Euclidean distance are probably the most commonly chosen measures of similarity. Partition is understood as providing for each data object an index or label of the cluster to which it is assigned. The goal is to find such a partition that minimizes the objective function, which, in our case, is the sum of squared distances of the data objects to their cluster representatives.

The resource-constrained project scheduling problem consists of a set of n activities, where each activity has to be processed without interruption to complete the project. The dummy activities 1 and n represent the beginning and the end of the project. The duration of the activity j, $j = 1, \ldots, n$ is denoted by d_j where $d_1 = d_n = 0$. There are r renewable resource types. The availability of each resource type k in each time period is r_k units, $k = 1, \ldots, r$. Each activity j requires r_{jk} units of resource k during each period of its duration where $r_{1k} = r_{nk} = 0$, $k = 1, \ldots, r$. All parameters are non-negative integers. There are precedence relations of the finish-start type (FS) with a zero parameter value (i.e. $FS = 0$) defined between the activities. In other words, activity i precedes activity j if j cannot start until i has been completed. The structure of the project can be represented by an activity-on-node network $G = (SV, SA)$, where SV is the set of activities and SA is the set of precedence relationships. SS_j (SP_j) is the set of successors (predecessors) of activity j, $j = 1, \ldots, n$. It is further assumed that $1 \in SP_j$, $j = 2, \ldots, n$, and $n \in SS_j$, $j = 1, \ldots, n-1$. The objective is to find a schedule $S$of activities starting times $[s_1, \ldots, s_n]$, where $s1 = 0$ and resource constraints are satisfied, such that the schedule duration $T(S) = s_n$ is minimized. The above formulated problem is a generalization of the classical job shop scheduling problem.

3.2 Optimizing Agents

To solve instances of each of the above described problems four specialized *TA-Teams* have been designed and implemented:

Instances of the EPTSP are solved using the following optimization algorithms:

- *Simple exchange* (Ex2) – deletes two random edges from the input solution thus breaking the tour under improvement into two disconnected paths and reconnects them in the other possible way, reversing one path.
- *Recombination1* (R1) – there are two input solutions. A subpath from one of them is randomly selected. In the next step it is supplemented with edges from the second solution. If this happens to be impossible the procedure constructs an edge connecting endpoint of the sub path with the closest point in the second input solution not yet in the resulting tour.
- *Mutation* (M) – two randomly selected points from the input solution are directly connected. This subpath is supplemented with edges as in R1. If this happens to be impossible the procedure constructs an edge connecting endpoint of the subpath with the closest remaining point from the initial solution.

Instances of the VRP are solved using the following optimization algorithms:

- *Opti3Opt* – an agent which is an implementation of the 3-opt local search algorithm, in which for all routes first three edges are removed and next remaining edges are reconnected in all possible ways.
- *Opti2Lambda* – an implementation of the local search algorithm based on λ-interchange local optimization method [10]. It operates on two selected routes and is based on the interchange/move of customers between routes. For each pair of routes from an individual a parts of routes of length less than or equal to λ are chosen and next these parts are shifted or exchanged, according to the selected operator. Possible operators are defined as pairs: (v, u), where u, $v = 1,\ldots, \lambda$ and denote the lengths of the part of routes which are moved or exchanged. For example, operator (2,0) indicates shifting two customers from the first route to the second route, operator (2,2) indicates an exchange of two customers between routes. Typically, $\lambda = 2$ and such value was used in the reported implementation.
- *OptiLSA* – an implementation of local search algorithm which operate on two selected routes. First, a node (customer) situated relatively far from the centroid of the first route is removed from it and next it is inserted to the second route.

Instances of the CP are solved using the following optimization procedures:

- *OptiLSA* - a simple local search algorithm which finds the local optimum by moving each activity to all possible places within the current solution.
- *OptiCA* - a heuristic based on using the one point crossover operator. Two initial solutions are repeatedly crossed until a better solution will be found or all crossing points will be checked.

– *OptiPRA* – a procedure based on the path-relinking algorithm. For a pair of solutions from the population a path between them is constructed. Next, the best of the feasible solutions is selected.

In the computational experiment each of the four *TA-Teams* implementations has been used to solve several instances of the respective combinatorial optimization problem. All these instances have been taken from several well-known benchmark datasets libraries as shown in Table 1.

Table 1. Instances used in the reported experiment

Problem	Instance names and dimensions	Source
EPTSP	pr76 (76 cities), pr144, (144) pr299 (299), pr439 (439), pr1002 (1002)	TSPLIB [15]
VRP	vrpnc1 (50 customers), vrpnc2 (75), vrpnc3 (100), vrpnc4 (150), vrpnc5 (199)	OR-Library [8]
CP	Ruspini (75 objects, 2 attributes), Iris (150, 4), Cleveland heart disease (303, 13), Credit approval (690,15) [12]	UCI Machine Learning Repository [16]
RCPSP	j3013_01 (30 activities), j3029_09 (30), j6009_02 (60), j9021_03 (90), j12016_01 (120)	PSPLIB [9]

3.3 Architectures

Each instance chosen for the experiment has been solved with the use of three different architectures:

- A1 – single A-Team,
- A2 – 5 A-Teams without communication nor migration feature,
- A3 – fully functional *TA-Teams* (with migration) consisting of 5 A-Teams.

For the A3 the following parameter settings have been used:

– *Migration size* = 1 (in one cycle one individual is sent from the common memory of an A-Team to the common memory of another A-Team)
– *Migration frequency* = 0.3 minute
– *Migration topology* – we consider a ring architecture, in which each A-Team receives communication from one adjacent A-Team and sends communication to another adjacent A-Team.
– *Migration policy – best-worst* policy, in which the best solution taken from the source population replaces the worst solution in the target population.

The other settings are shown in Table 1 and guarantee that for each problem the complexity in all three architectures remains similar: the total number of individual solutions and agents is the same.

Experiment has been carried out on the cluster Holk of the Tricity Academic Computer Network built of 256 Intel Itanium 2 Dual Core with 12 MB L3 cache processors with Mellanox InfiniBand interconnections with 10Gb/s bandwidth.

TA-Teams have been implemented using JABAT middleware derived from JADE. As a consequence it has been possible to create agent containers on different

Table 2. Experiment settings

	A1	A2 and A3
Common memory size (number of individuals)	250 - RCPSP 50 - VRP, CP 35 - TSP	50 - RCPSP 10 – VRP, CP 7 - TSP
Number of optimizing agents of each kind cooperating with each common memory	5	1

machines and connecting them to the main platform. Then agents may migrate from the main platform to these containers. In each of the reported experiments 5 nodes of the cluster have been used - one for the main platform and four for the optimising agents to migrate to.

For each problem and each architecture there were no less then 30 runs. Computation errors have been calculated in relation to the best results known for the problems. The results - in terms of relative computation error - have been averaged.

4 Computational Experiment Results

Table 3 shows mean values and standard percentage deviations of the respective fitness functions calculated for all computational experiment runs as specified in the experiment design described in the previous section.

The main question investigated in the reported experiment is to decide whether choosing a particular architecture from among the considered ones influences the quality of solutions? It has been decided to use the non-parametric Friedman test to obtain the answer to the question weather particular architectures are equally effective independently of the kind of problem being solved.

The above test has been based on weights (points) assigned to architectures used in the experiment: 1, 2 or 3 points for the worse, second worse and best architecture. The test aimed at deciding among the following hypotheses:

- H0 – zero hypothesis: considered architectures are statistically equally effective regardless the kind of problem,
- H1 – alternative hypothesis: not all architectures are equally effective.

The analysis has been carried out at the significance level of 0.05. The respective value of the $\chi 2$ statistics with 3 architectures and 23 instances of the considered problems is 18 and the value of $\chi 2$ distribution is equal to 5.99. Thus it can be observed that not all architectures are equally effective regardless of the kind of problem of which instances are being solved.

In Fig. 1 and 2 sums of weights obtained by the architectures for each of the considered problems and overall total of weights for each architecture are, respectively, compared. The highest score in total obtained architecture A3. Within the problems A3 also scored best except of RCPSP problem, where A1 and A2 both scored the same total.

Table 3. Mean values and standard percentage deviations of the respective fitness functions calculated for all computational experiment runs

Problem	A1		A2		A3	
pr76	108904,8	0,8%	108628,1	0,5%	108382,1	0,4%
pr144	58714,8	0,4%	58636,4	0,2%	58606,1	0,2%
pr299	49046,7	0,9%	49481,0	0,8%	48862,7	0,7%
pr439	111860,8	1,5%	111209,5	1,2%	110381,6	1,5%
pr1002	274172,0	1,5%	279592,8	0,6%	273698,2	1,1%
vrpnc1	524,6	0,0%	524,6	0,0%	524,6	0,0%
vrpnc2	851,8	2,0%	853,9	2,2%	851,0	1,9%
vrpnc3	837,1	1,3%	837,8	1,4%	837,8	1,4%
vrpnc4	1064,8	3,5%	1067,5	3,8%	1062,8	3,3%
vrpnc5	1376,0	6,5%	1379,4	6,8%	1367,5	5,9%
CreditTrening	1923,7	8,7%	1718,4	3,7%	1655,0	3,4%
HeartTrening	951,8	14,5%	888,5	4,0%	862,0	3,3%
Iris2	295,4	25,0%	270,3	13,6%	227,3	10,5%
Iris3	157,4	19,5%	141,7	13,0%	121,4	16,5%
Iris4	167,5	9,3%	133,5	17,9%	134,2	17,8%
Ruspini2	3549,8	5,8%	3406,8	2,3%	3369,3	2,4%
Ruspini3	2342,5	7,2%	2317,7	11,4%	2083,0	15,1%
Ruspini4	1752,2	5,9%	1841,6	12,9%	2017,6	18,8%
j30_29_9	98,4	1,0%	98,1	0,9%	98,3	0,9%
j30_29_9	85,2	0,5%	84,7	0,4%	84,9	0,4%
j60_9_2	71,0	0,0%	71,0	0,0%	71,0	0,0%
j90_21_3	129,4	0,6%	128,7	0,6%	128,6	0,6%
j120_16_1	213,0	0,6%	212,3	0,6%	212,3	0,6%

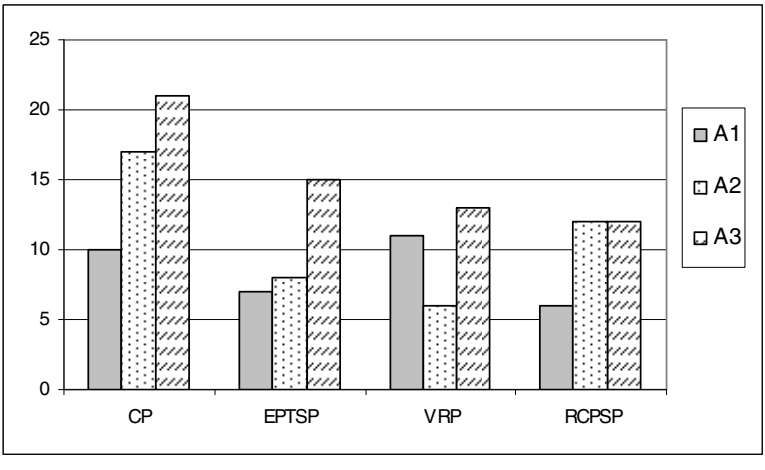


Fig. 1. The Friedman test weights for each problem and each architecture

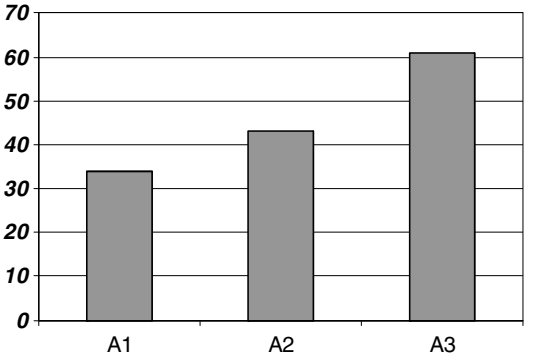


Fig. 2. The total of Friedman weights for the investigated architectures

5 Conclusions

The presented research has confirmed that in most cases integrating the distributed evolutionary concept, and especially the island based evolutionary algorithm with the A-Team paradigm might result in achieving a noticeable improvement in the quality of the computation results. Thus, the paper confirms the importance of choosing an effective architecture.

The range of parameters used in the experiments is insufficient to draw further conclusions as to strategy of choosing features of the *TA-Teams* constructed from heterogeneous optimizing agents with a view to achieving best results. Future research will focus on evaluating effects of a wider set of parameters and solving more test data, which might be helpful in identifying more general observations.

Acknowledgments. Calculations have been performed in the Academic Computer Centre TASK in Gdańsk. The research has been supported by the Ministry of Science and Higher Education no. N N519 576438 for years 2010-2013.

References

1. Barbucha, D., Czarnowski, I., Jêdrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: Influence of the Working Strategy on A-Team Performance. Smart Information and Knowledge Management, 83–102 (2010)
2. Barbucha, D., Czarnowski, I., Jêdrzejowicz, P., Ratajczak, E., Wierzbowska, I.: JADE-Based A-Team as a Tool for Implementing Population-Based Algorithms. In: Chen, Y., Abraham, A. (eds.) Intelligent Systems Design and Applications, IDSA, Jinan Shandong China, pp. 144–149. IEEE, Los Alamitos (2006)
3. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper, Exp. 3(3), 6–20 (2003)
4. Cohoon, J.P., Hegde, S.U., Martin, W.N., Richards, D.: Punctuated Equilibria: a Parallel Genetic Algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 148–154. Lawrence Erlbaum Associates, HiUsdale (1987)

5. Czarnowski, I., Jêdrzejowicz, P., Wierzbowska, I.: A-Team Middleware on a Cluster. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS (LNAI), vol. 5559, pp. 764–772. Springer, Heidelberg (2009)
6. Jêdrzejowicz, P., Wierzbowska, I.: JADE-Based A-Team Environment. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
7. Jędrzejowicz, P., Wierzbowska, I.: Parallel Cooperating A-Teams Solving Instances of the Euclidean Planar Travelling Salesman Problem. In: O'Shea, J., et al. (eds.) KES AMSTA 2011. Springer, Heidelberg (to appear, 2011)
8. OR-Library, http://people.brunel.ac.uk/~mastjjb/jeb/orlib/vrpinfo.html
9. PSPLIB, http://129.187.106.231/psplib
10. Talukdar, S.N.: Collaboration Rules for Autonomous Software Agents. Decision Support Systems 24, 269–278 (1999)
11. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Cooperation Schemes for Autonomous Agents. Journal of Heuristics 4(4), 295–321 (1998)
12. Talukdar, S.N., Pyo, S.S., Giras, T.: Asynchronous Procedures for Parallel Processing. IEEE Trans. on PAS PAS-102(11), 3652–3659 (1983)
13. Talukdar, S.N., de Souza, P., Murthy, S.: Organizations for Computer-Based Agents. Engineering Intelligent Systems 1(2), 56–69 (1993)
14. Tanese, R.: Distributed genetic algorithms. In: Schaffer, J. (ed.) Proceedings of the Third International Conference on Genetic Algorithms, pp. 434–439. Morgan Kaufmann, San Mateo (1989)
15. TSP library, http://comopt.ifi.uniheidelberg.de/software/TSPLIB95
16. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/
17. Whitley, D., Rana, S., Heckendorn, R.B.: The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. Journal of Computing and Information Technology 7, 33–47 (1998)

Solving the Capacitated Vehicle Routing Problem by a Team of Parallel Heterogeneous Cooperating Agents

Dariusz Barbucha

Department of Information Systems
Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
barbucha@am.gdynia.pl

Abstract. The paper aims at evaluating experimentally the influence a population size and method of creating the initial population of solutions on computational results produced by the population-based multi-agent systems solving instances of the capacitated vehicle routing problem. The reported experiment involved several methods of creating of initial population of various sizes and several cooperating agents representing improvement heuristics working in parallel.

Keywords: collective problem solving, cooperation, agent-based optimization, capacitated vehicle routing problem.

1 Introduction

The effectivenes of solving combinatorial optimization problems strongly depends on both - problem particular features and the method used for solve it. Traditional exact methods (like branch-and-bound), which guarantee obtaing the optimal solution, are often unpractical because of their exponential computational complexity. On the other hand, the wide range of approximate methods, which in fact, not guarantee obtainig the optimal solution, require less computation time to obtain solutions.

Over the last years, interest in combining of various algorithms, including population-based methods, has risen considerably among researchers in the field of combinatorial optimization. Different forms of hybridization of various search algorithms, like sequential or parallel execution of combined algorithms or incorporating one algorithm in another [11], gives an opportunity to diversify the problem-solving strategies. As a consequence, such implementations have provided a very powerful search algorithms and produce solutions better than methods based on a single optimization procedure or provide (sub)optimal solutions in shorter time. Moreover, for many NP-hard problems, the best found results are obtained by the hybrid algorithms.

On the other hand, during the last decade, the agent technology [8], where the multi-agent paradigm is used to enhance traditional approaches to solving difficult optimization problems has emerged [1], [10].

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 332–341, 2011.

Hybridization of different methods into one agent-based approach, in which a set of heterogeneous software agents, each representing improvement algorithm, cooperate to solve a problem by dynamically evolving a population of solutions has been proposed by Barbucha et. al. [3]. As a population-based approach it shares several features common to all population-based methods. Among them most characteristic include diversification by means of generating a population of individuals which are solutions or parts of solutions and introducing some random noises at various stages of searching for the solution.

As it has been shown in previous works of the author, an effectiveness of the agent-based approaches for solving instances of the optimization problems may depends on various parameters, like number and form of optimization agents [2], mode of cooperation [4], or mode of communication between agents [5]. Additionally, in the population-based methods, a proper definition of the population features (its size, methods of creating and operating on them during the search) may be crucial in terms of quality of results obtained.

The paper aims at evaluating to what extent a population size and a method of creating the initial population may influence computational results produced by a multi-agent system while solving instances of one of the classic combinatorial optimization problems - the capacitated vehicle routing problem. To gain the required insight the computational experiment has been carried out using JABAT - the middleware cooperative search environment [3], based on the multi-agent paradigm. The reported experiment involved several methods of creating the initial population of various sizes and several heterogeneous cooperating agents representing improvement local search heuristics working in parallel.

The paper is organized as follows. Section 2 describes main features of the cooperative multi-agent environment used in the experiment. In section 3 an implementation of the system for solving the capacitated vehicle routing problem is described. Section 4 shows the experiment plan and discusses its results. Finally, conclusions included in Section 5 end the paper.

2 A Multi-agent Cooperative Search Environment

The middleware cooperative search environment JABAT [3] is based on the concept of an *asynchronous team* (A-Team), originally introduced by Talukdar [12]. A-Team is a collection of software agents which collectively work and cooperate to solve a problem by dynamically evolving the population of solutions stored in the common memory. Each agent encapsulates a particular problem-solving method, which usually is inspired by some natural phenomena including, for example, evolutionary processes or particle swarm optimization, as well as local search techniques like, for example, tabu search.

The ground principle of asynchronous teams rests on combining algorithms, which alone could be inept for the task, into effective problem-solving organizations, possibly creating a synergetic effect, in which the combined effect of cooperation between agents is greater than the sum of their separate effects.

Main functionality of the proposed environment is organizing and conducting the process of search for the best solutionm mainly carried out by two types of

agents: solution manager and a set of optimization agents. The search process is organized as a sequence of steps, including initialization and improvement phases. At first the initial population of solutions is generated. Individuals forming the initial population are, at the following computation stages, improved by independently acting autonomous optimization agents, thus increasing chances for reaching the global optimum. Finally, when the stopping criterion is met, the best solution from the population is taken as the result. Details on implementation of the environment can be found in [3].

3 An Agent-Based Approach for Solving the Capacitated Vehicle Routing Problem

The suggested investigation has been carried out on instances of the Capacitated Vehicle Routing Problem (CVRP). It can be modelled as an undirected graph $G = (V, E)$, where $V = \{0, 1, \ldots, N\}$ is the set of nodes and E is a set of edges. Node 0 is a central depot with NV identical vehicles of capacity W and each other node $i \in V - \{0\}$ denotes customer (with its request) with a non-negative demand d_i. Each link (i, j) between two customers denotes the shortest path from customer i to j and is described by the cost c_{ij} of travel from i to j by shortest path $(i, j = 1 \ldots, N)$. It is assumed that $c_{ij} = c_{ji}$.

The goal is to find vehicle routes which minimize the total cost of travel (or travel distance) and such that each route starts and ends at the depot, each customer is serviced exactly once by a single vehicle, and the total load on any vehicle associated with a given route does not exceed the vehicle capacity.

In addition to the vehicle capacity constraint, a further limitation can be imposed on the total route duration. In such case t_{ij} is defined to represent the travel time for each edge $(i, j) \in E$ $(t_{ij} = t_{ji})$, and t_i represents the service time at any vertex i $(i \in V \setminus \{0\})$. It is required that the total duration of any route should not exceed a preset bound T.

In order to used the beforehand characterized environment for solving an optimization problem, several elements need to be defined. These include representation of individuals, population size, method of creating the initial population, the form of fitness function, improvement algorithms, strategy for managing a population of individuals, and stopping criterion.

3.1 Representation of an Individual

Representation of an individual (solutions) should reflect typical features of the problem, assuring an effective processing of individual by operators or functions. One of the most common and used in suggested approach is a permutation of N numbers (customers), where the order of numbers reflects order in which customers are visited. Originally such representation comes from the Traveling Salesman Problem (TSP) and is known as path representation. Adapting it for CVRP requires additional procedure of splitting the individual on segments (routes).

3.2 Creating the Initial Population

Typically the initial population of individuals (solutions) is formed randomly or using a dedicated constructive heuristic. Two methods have been suggested for creating the initial population. As a starting point, they require a permutation of N numbers (customers), which reflects one route, but uses different approaches to transfer such permutation to a list of routes. Hence, in the proposed approach the process of creating an initial population of individuals is divided into two phases:

1. Creation of the permutation of N numbers,
2. Division of these permutation into segments (assignment of customers to routes of the available vehicles).

Creation of the Permutation. The permutation is created randomly or using polar representation of each vertex.

Random creation of the permutation. The first method simply generates randomly permutation of N numbers which determines an individual.

Polar creation of the permutation. The second method is based on polar representation of each vertex (customer) and uses an idea originated from *split* phase of the *sweep* algorithm of Gillett and Miller [7]. First, each vertex $i \in V$ is transformed from cartesian coordinates to polar coordinates (θ_i, ρ_i), where θ_i is the angle and ρ_i is the ray length. Generation of each individual in the population starts from randomly choosing an arbitrary vertex i^* and assigning a value $\theta_i^* = 0$ to it. Next, the remaining angles centered at 0 from the initial ray $(0, i^*)$ are computed and the vertices are ranked in increasing order of their θ_i value. Resulting ranking determines an individual.

Division of Permutation into Segments (Routes). The process of assignment vertices to clusters (routes) starts from the first element of permutation (first unrouted vertex), assigns it to the route using one of the insertion method in such way that total capacity of vehicle assigned to the current route and the maximal route length are not exceeded. The whole process is repeated until end of the permutation is reached.

The form of each route is determined by the insertion method used in process of transfromation of permutation into the routes. Two methods are suggested in the proposed implementation: *simple insertion* and *cheapest insertion.*

Simple insertion. The first method creates routes one by one through succesively inserting consecutive elements of permutation into the current route. In case of violation of capacity or maximal route constraints the next route is constructed and the next customer is inserted into this route. The order of customers in each route is determined by their order within the permutation.

Cheapest insertion. The cheapest insertion method tries to insert a customer currently taken from the permutation and insert it into the already existing set of routes. The main problem is to determine the route and the position in this

route on which the considered customer should be inserted. Let $v \in V$ is the currently considered customer taken from the permutation and $R = \{R_1, ...R_k\}$ - current set of routes, where k is a number of routes. For each edge $(i, j) \in R_p$, where $i, j \in V$, $p = 1..k$ calculate $c(R_p, (i, j), v) = c(i, v) + c(v, j) - c(i, j)$. The minimum of $c(R_p, (i, j), v)$ determines the route and position of insertion.

3.3 Population Size

Typically, population size is set experimentally as a compromise between effectiveness of the suggested approach and computation time needed by the system during the search process in order to obtain satisfactory results. In the suggested approach the population size has been set at the same level during the whole process of search but the question how many individuals such population should include is open, for which we will try to find the answer in the computational experiment.

3.4 Fitness

Each individual from the population is evaluated and value of its fitness is calculated as a sum of the costs related to each vehicles' route.

3.5 Improvement Heuristics

In the proposed system it has been decided to implement several local improvement procedures. Each algorithm starts from a feasible solution $s \in S$, where S - set of all feasible solutions. Next, according to the defined neighborhood $N(s)$ of s, the best move, which transforms s into $s' \in N(s)$, is selected. If s' improves the best obtained so far solution, then s' is accepted as a new solution and the process is repeated.

The suggested improvement moves are divided into two groups operating on one (*intra-route*) or two (*inter-routes*) routes:

- *Intra-route moves*:
  - *relocate(1)_1R* - relocates one customer from one position to another within the same route
  - *relocate(2)_1R* - relocates two customers within the same route
  - *swap(1)_1R* - swaps two customers within the same route
  - *2opt_1R* - removes two edges from one route forming two disconnected segments and next these reconnects these segments in all possible ways.
- *Inter-routes moves*:
  - *relocate(1)_2R* - relocates one customer from one route to another
  - *relocate(2)_2R* - relocates two customers from one route to another
  - *swap(1)_2R* - swaps one customer from one route with one customer from second route
  - *swap(2)_2R* - swaps two customers from one route with two customers from second route

Moreover, the above defined moves may be used in two ways:

- *First Accepted move* (FA) - the procedure of finding a local optimum is restricted to the first iteration of local search procedure. For example in *relocate(1)_1R* for each route from the current solution all possible moves of each customer are considered and move which gives better solution than the current one and brings the biggest increment in fitness function is accepted as a resulting solution. Otherwise, the original solution, without improvement, is returned as a solution.
- *Best Accepted move* (BA) - full procedure of finding a local optimum is implemented. For example in *relocate(1)_1R* for each route from the current solution all possiblem moves of each customer are considered and move which gives better solution than the current one and brings the biggest increment in fitness function is accepted as a temporary resulting solution. Next, these solution is treated as the current solution, and process of finding the best move from these solution is repeated. If no improvement is observed, the process stops.

In all improvement algorithms and all kinds of moves only moves which improve the current solution are accepted and added to the population of individuals.

Management of the Population of Individuals. Management of population of individuals includs selection, acceptance, and updating criteria. Selection criterion controls how to choose solutions from the common memory, acceptance criterion describes whether new individual returned by the optimization agent after improvement phase is accepted (or not) to be added to the common memory, and updating criterion describes how to merge the improved solutions returned by the optimizing agents with the whole population.

In the proposed implementation a randomly chosen individual is forwarded to the optimizing agents for improvement. Additionally, in order to prevent such solution from being sent to the other optimization agent, it is blocked for a period of time. An updating mechanism gives the solution manager an opportunity to replace the worst solution from the current population by a solution currently received from an optimization agent. Additionally, if the last consecutive five solutions received from optimization agents have not improved the current best solution in the population, the worst solution is removed from the population and a newly generated one is added to the pool of individuals. Only solutions locally or globally improved received from optimization agents are accepted and added to the common memory.

Stopping Criterion. Stopping criterion determines when to stop the process of search. Typically used criteria are the number of iterations or time after which the process of search will stop. In the proposed implementation the system stops after a given period of time without improvement of the current best solution.

4 Computational Experiment

Computational experiment has been carried out to validate effectiveness of the proposed approach and to evaluate how the size of the population and the method of creating an initial population may influence computation results, measured as mean relative error (MRE) from the optimal (or the best known) solution.

Five different population sizes have been used in the experiment: *5, 10, 20, 30, 50* and four methods of creating the initial population have been suggested: *Random/Simple, Random/Cheapest, Polar/Simple, Polar/Cheapest*. The first part of each name shows how the initial order of customer in permutation is determined (randomly or using polar representation of each vertex), the second part - how the final order of customers is defined in resulting routes (reflecting simple order stemming from permutation or using cheapest insertion method). Moreover, two groups of optimization agents have been used in the experiment with *First Accepted* and *Best Accepted* moves, implemented, respectively, within their local search algorithms.

Computational experiment has been carried out on 5 test instances of the Christofides et al. [6] containing 50-199 customers and only the capacity constraint.

Each of the 5 instances of the problem was solved using combinations of each possible *population size*, each *method of creating the initial population*, and each *move* defined in local search heuristic, in total giving 200 (5x5x4x2) test problems. Moreover, each test problem was repeatedly solved 10 times and mean results from these runs were recorded.

All computations have been carried out on the cluster Holk of the Tricity Academic Computer Network built of 256 Intel Itanium 2 Dual Core with 12 MB L3 cache processors with Mellanox InfiniBand interconnections with 10Gb/s bandwidth.

The experiment results are shown in Tab. 1 and Tab. 2. For all considered population sizes and all methods of creating the initial population both tables present mean values of relative error - MRE (in %) from the best known solution [9] obtained by the system while solving instances of CVRP.

Analysis of the results presented in Table 1 allows one to conclude that performance of the presented multi-agent approach for solving CVRP strongly depends on all three considered factors: population size, method of creating an initial population and kind of moves used in local search procedures of optimization agents.

Among the methods for creating the initial population, a group of methods with polar representation of customers used for determining the order in which customers are assigned to the routes (*Polar/**) outperforms a group with order randomly generated (*Random/**) for almost all instances and considered population sizes. Also, the better performance of the system can be observed when the cheapest insertion method (**/Cheapest*) has been used for determining the

Table 1. MRE calculated for all tested population sizes and all considered methods of creating an initial population using *Best Accepted* move in the local search algorithms

Population size	*Creating of initial population*	INSTANCE *vrpnc1* (50)	*vrpnc2* (75)	*vrpnc3* (100)	*vrpnc4* (150)	*vrpnc5* (199)
5	*Random/Simple*	1.99%	3.26%	5.04%	10.21%	11.37%
	Random/Cheapest	1.99%	3.46%	5.08%	10.24%	12.91%
	Polar/Simple	0.00%	2.77%	1.91%	5.36%	8.71%
	Polar/Cheapest	0.00%	2.11%	1.89%	2.97%	5.13%
10	*Random/Simple*	3.61%	3.03%	3.96%	13.21%	17.87%
	Random/Cheapest	2.72%	4.00%	4.42%	9.87%	14.91%
	Polar/Simple	0.00%	2.73%	2.06%	4.44%	8.18%
	Polar/Cheapest	0.00%	1.64%	2.14%	3.37%	6.58%
20	*Random/Simple*	1.29%	4.21%	6.83%	12.84%	16.14%
	Random/Cheapest	0.46%	3.58%	3.28%	7.57%	13.52%
	Polar/Simple	0.59%	2.74%	2.17%	6.56%	8.76%
	Polar/Cheapest	0.00%	2.24%	1.62%	1.96%	4.60%
30	*Random/Simple*	3.01%	6.03%	6.96%	10.84%	13.81%
	Random/Cheapest	0.23%	3.31%	3.14%	8.12%	14.92%
	Polar/Simple	0.69%	2.84%	1.79%	4.33%	7.02%
	Polar/Cheapest	0.00%	2.45%	1.28%	4.39%	5.75%
50	*Random/Simple*	5.27%	8.08%	11.84%	12.13%	16.94%
	Random/Cheapest	3.76%	2.49%	2.88%	13.52%	18.71%
	Polar/Simple	0.71%	2.36%	1.39%	3.47%	8.27%
	Polar/Cheapest	0.00%	1.40%	1.26%	2.62%	4.71%

final order of customers in all vehicles' routes for majority of cases, although the difference is not as significant as in former case.

Another observation is that the results produced by the system strongly depend on the instance of the problem. MRE increases with the size of the problem, especially for instances where number of customers exceeds 100.

By focusing the observation on dependence of results on population size one can conclude that influence of population size on performance of the system is generally not significant. Hovewer the answer to the question about how many individuals the population should include may vary for different instances, for most of them the reasonable size seems to be set to between 10 and 30.

And finally, comparison of results from both tables does not provide a simple answer what kind of move implemented in local search procedures of optimization agents is better in terms of produced results. Taking into account the overall average results, the behaviour of agents for both moves is similar, although they may differ for particular instances.

It has been also observed that agents with *First Accepted* move implemented in them require much more time than agents with *Best Acceptance* move in order to obtain similar results. The convergence to the best results is definitely slower in case of agents with FA move than in case of BA move implemented in them.

Table 2. MRE calculated for all tested population sizes and all considered methods of creating an initial population using *First Accepted* move in the local search algorithms

Population size	*Creating of initial population*	INSTANCE *vrpnc1* (50)	*vrpnc2* (75)	*vrpnc3* (100)	*vrpnc4* (150)	*vrpnc5* (199)
5	*Random/Simple*	1.12%	2.76%	5.17%	11.69%	12.60%
	Random/Cheapest	1.63%	5.11%	4.60%	15.74%	12.88%
	Polar/Simple	0.02%	3.89%	3.33%	3.57%	4.52%
	Polar/Cheapest	0.00%	2.52%	2.04%	4.60%	5.96%
10	*Random/Simple*	3.28%	4.20%	4.42%	12.59%	18.52%
	Random/Cheapest	3.46%	5.95%	5.73%	10.11%	8.94%
	Polar/Simple	0.02%	4.52%	2.96%	3.15%	5.84%
	Polar/Cheapest	0.00%	2.91%	2.75%	2.94%	5.96%
20	*Random/Simple*	1.73%	5.58%	8.76%	11.91%	18.68%
	Random/Cheapest	3.35%	4.44%	8.00%	5.18%	9.32%
	Polar/Simple	0.61%	2.80%	1.97%	3.59%	5.46%
	Polar/Cheapest	0.00%	2.83%	1.71%	2.68%	5.25%
30	*Random/Simple*	4.26%	8.27%	8.63%	13.14%	17.55%
	Random/Cheapest	3.17%	3.52%	4.38%	8.27%	8.67%
	Polar/Simple	1.43%	2.96%	1.32%	3.66%	4.20%
	Polar/Cheapest	0.00%	1.92%	2.46%	2.87%	5.33%
50	*Random/Simple*	5.51%	8.33%	10.16%	13.60%	15.91%
	Random/Cheapest	0.61%	1.98%	1.79%	7.43%	9.61%
	Polar/Simple	0.00%	2.67%	1.68%	2.28%	3.82%
	Polar/Cheapest	0.00%	1.99%	1.50%	2.36%	5.53%

5 Conclusions

The main goal of the paper was to evaluate to what extent some population's features of multi-agent system (population size and method of creating of initial population) as well as a kind of move defined in local search procedures used for improvement a solutions during the search process influence computational results produced by such system while solving instances of the capacitated vehicle routing problem. The experiment has shown the strong dependence of obtained results on method of creating an initial population of individuals. Also, dependence of mean relative error on other two factors has been noted but not for all instances and with weaker strength.

Acknowledgments. The research has been supported by the Ministry of Science and Higher Education grant no. N N519 576438. Calculations have been performed in the Academic Computer Centre TASK in Gdansk.

References

1. Aydin, M.E., Fogarty, T.C.: Teams of autonomous agents for job-shop scheduling problems: An Experimental Study. Journal of Intelligent Manufacturing 15(4), 455–462 (2004)

2. Barbucha, D., Jędrzejowicz, P.: An experimental investigation of the synergetic effect of multiple agents working together in the A-team. Systems Science 34(2), 55–63 (2008)
3. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: JABAT Middleware as a Tool for Solving Optimization Problems. In: Nguyen, N.T., Kowalczyk, R. (eds.) Transactions on CCI II. LNCS, vol. 6450, pp. 181–195. Springer, Heidelberg (2010)
4. Barbucha, D.: Cooperative Solution to the Vehicle Routing Problem. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS, vol. 6071, pp. 180–189. Springer, Heidelberg (2010)
5. Barbucha, D.: Synchronous vs. Asynchronous Cooperative Approach to Solving the Vehicle Routing Problem. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS (LNAI), vol. 6421, pp. 403–412. Springer, Heidelberg (2010)
6. Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.): Combinatorial optimization. John Wiley, Chichester (1979)
7. Gillett, B.E., Miller, L.R.: A heuristic algorithm for the vehicle dispatch problem. Operations Research 22, 240–349 (1974)
8. Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1, 7–38 (1998)
9. Laporte, G., Gendreau, M., Potvin, J., Semet, F.: Classical and modern heuristics for the vehicle routing problem. International Transactions in Operational Research 7, 285–300 (2000)
10. Parunak, H.V.D.: Agents in overalls: Experiences and issues in the development and deployment of industrial agent-based systems. International Journal of Cooperative Information Systems 9(3), 209–228 (2000)
11. Puchinger, J., Raidl, G.R.: Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 41–53. Springer, Heidelberg (2005)
12. Talukdar, S., Baeretzen, L., Gove, A., de Souza, P.: Asynchronous teams: Cooperation schemes for autonomous agents. Journal of Heuristics 4, 295–321 (1998)

Validated Decision Trees versus Collective Decisions

Krzysztof Grąbczewski

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
kg@is.umk.pl
http://www.is.umk.pl/~kg

Abstract. In the most common decision tree (DT) induction approaches, cross-validation based processes validate the final DT model. This article answers many questions about advantages of using different types of committees constructed from the DTs generated within the validation process, over single validated DTs. Some new techniques of providing committee members and their collective decisions are introduced and evaluated among other methods. The conclusions presented here, are useful both for human experts and automated meta-learning approaches.

Keywords: decision trees, committees, pruning, validation, meta-learning.

1 Introduction

Decision tree (DT) classification models are very popular because of their comprehensibility and fast learning processes. The most popular algorithms include CART [1], C4.5 [10] and QUEST [7]. The SSV approach [4,5,3] has also proven to be successful. All these algorithms build oversized trees in a top-down manner and then prune them for better generalization. The most often used pruning techniques are based on cross-validation (CV), e.g. the cross-complexity pruning of CART and degree-based pruning of SSV. As an alternative to postpruning methods, including the CV-based strategies and Reduced Error Pruning (REP) [9], some prepruning algorithms have also been proposed, but despite their small computational requirements they are not as common as postpruning, because their applications are less successful.

In this paper, using CV-based committees of decision trees instead of single validated trees is proposed. Some new techniques are introduced into DT validation and decision making modules. The new components and some commonly known ones are examined in an extensive test. The results analysis introduces a number of interesting conclusions about how to build successful DT committees, which should be very useful for human experts building such models manually and for automated meta-learning approaches.

The rest of the article is organized as follows: first the algorithm for DT induction, used in the analysis, is introduced (section 2), then the techniques of CV committees construction are presented (section 3) and next, the experiments are described and their results including thorough analysis are presented (section 4).

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 342–351, 2011.

2 Decision Tree Algorithms and Validation Methods

In general, DT induction algorithms can be referred to as recursive, top-down splitting procedures. Most of them just apply a greedy search to subsequently divide the feature space into clear regions (i.e. containing data belonging to a single class). The main differences between the algorithms consist in the way, they split nodes into subnodes. Some methods use only binary splits, others accept splits into more than two subnodes. Some use univariate splits and the others perform multivariate analysis.

For the sake of this work, trees with binary splits were produced with a greedy, almost exhaustive search. Only univariate binary splits were examined in the context of four different split criteria: the Gini index, information gain (IG), information gain ratio (IGR) and the separability of split value (SSV) criterion. To split an ordered feature, all the points between adjacent training data objects representing different classes were analyzed. The splits of unordered (categorical) features were selected from the set of possible binary splits with a restriction to avoid the bias toward unordered features: the number of analyzed splits was approximately equal to the number of data vectors within the sample.

Generalization capabilities of DT learning machines come from the use of techniques belonging to one of the following two categories: prepruning and postpruning. Both allow to avoid overfitting the training data, but in different ways: the former ones prevent splitting the nodes during the search and the latter build oversized and overfitted trees and then prune them. Prepruning is known from being less accurate [8], so in vast majority of cases postpruning is used. The postpruning techniques are also diverse. Some use statistical tests to estimate split significance, some other use a single validation dataset to adjust the model built for the training data (this is the case of REP, which prunes the tree so as to maximize the accuracy on the validation dataset) and yet another use multiple training and validation to estimate the most accurate complexity of the tree. Cross-validation (CV) is the most common technique to perform such multiple validation. It constitutes the basis of both cost-complexity (CC) pruning [1] and degree-based pruning of SSV. In CC pruning a parameter controlling the trade-off between the accuracy and size of the trees is adjusted within the CV and in degree-based pruning, an optimal degree of pruning is determined. The idea of the degree of pruning is based on counting the difference between errors (within training data) of a node and its descendants. Pruning with given degree means pruning the splits, for which the count is not greater than the degree.

3 CV Committees

During the processes of DT validation based on CV, a tree is generated and validated in each fold of the CV. It is the scenario of both cross-complexity and degree-based validation. CV committees arose from the (natural) suspicion that these nicely validated DTs should provide a successful collective model. The idea of using the validated trees as a committee, results in a number of possibilities in the preparation of the committee members and then combining their decisions.

First of all, the committees may include different numbers of members. The most reasonable way of members selection seems the one obtained by ordering the candidate trees by their validation results and adding them one by one, creating a series of committees.

Training data error factors. When searching for optimal pruning parameters, a series of values is examined. Estimation of given value efficiency is usually done by counting the classification error on the validation data. In the case of DTs, especially, when the validation dataset is small, the pruned tree may overfit the validation data—some important branches of the tree may be removed, because there are no corresponding data within the validation set. REP is especially prone to such tree corruption.

To diminish this effect, the accuracy of pruned trees can be estimated not just on the basis of the validation data, but also on the training data. Because the training data is usually larger, it would dominate the result if we took the union of the sets and counted the accurate decisions. Therefore we calculate accuracies separately for the training and validation data and combine them as $Acc_{tst} + factor \cdot Acc_{trn}$. The idea has been examined for the factor values of 0, 0.5 and 1. The results are presented and discussed in section 4.

Committee decision making. Classifiers committees usually make the final decision by counting votes of the members and determining the winner class of the voting. When the member classifiers are capable of providing probabilities of belonging to different classes, instead of simple voting, the average probabilities supplied by the members may decide about the committee decision. As shown below, such decisions are much more reliable then voting.

Another new idea is to use weights provided by the committee members in place of probabilities. In the case of DTs, the weights can be the numbers of vectors of different classes falling into the same tree leaf as the examined data item. When summed for all the trees (committee members), they can be used to determine the winner class. The suspicion that it could perform better than the probabilities-based collective decisions turned out quite missed (see the discussion in section 4).

Validation modes. Pruning decision trees in the validation process, can also be done in a number of ways. In the preceding section, reduced error pruning (REP), cost-complexity (CC) optimization and degree-based pruning are mentioned. The three methods have been applied in the experiments described below, but the last two were run in two variants: common optimum parameter for all the validated trees was searched or independent optimization for each CV fold was performed. In the tables, the two methods are distinguished by the suffixes -common (-com.) and -seperate (-sep.) respectively.

4 Experiments and Their Analysis

To check the value of the committees being the main subject of the article, different configurations of the committees have been tested and compared against traditional models with single validated trees as the final classifiers. The comparison was done on the basis of 10 repetitions of 10-fold CV (100 accuracy results) and paired t-test

with $\alpha = 0.01$. For the same 100 splits, standard CV validated (CC and degree) trees have been trained and different committees created. Thanks to the solutions of Intemi [6], the comparison could be easily conducted in completely fair manner: internal CVs (within the training data) used exactly the same data splits for all the committees and as a result the same set of induced trees. Apart from the fairness, the experiments have an advantage of fast calculations, because the trees did not have to be induced repeatedly—Intemi provides mechanisms for machine unification which saves time and memory without much effort from the designer of the test projects.

Different CV committees were created by different settings of the following parameters (all their values are shortly introduced in the preceding section):

- committee size (i.e. members count): 1,...,10,
- training data error factor; 0, 0.5, 1,
- decision making strategy: voting, probabilities, weights,
- CV validation method: REP, CC-common, CC-separate, degree-common, degree-separate,
- split criterion: Gini index, IG, IGR, SSV.

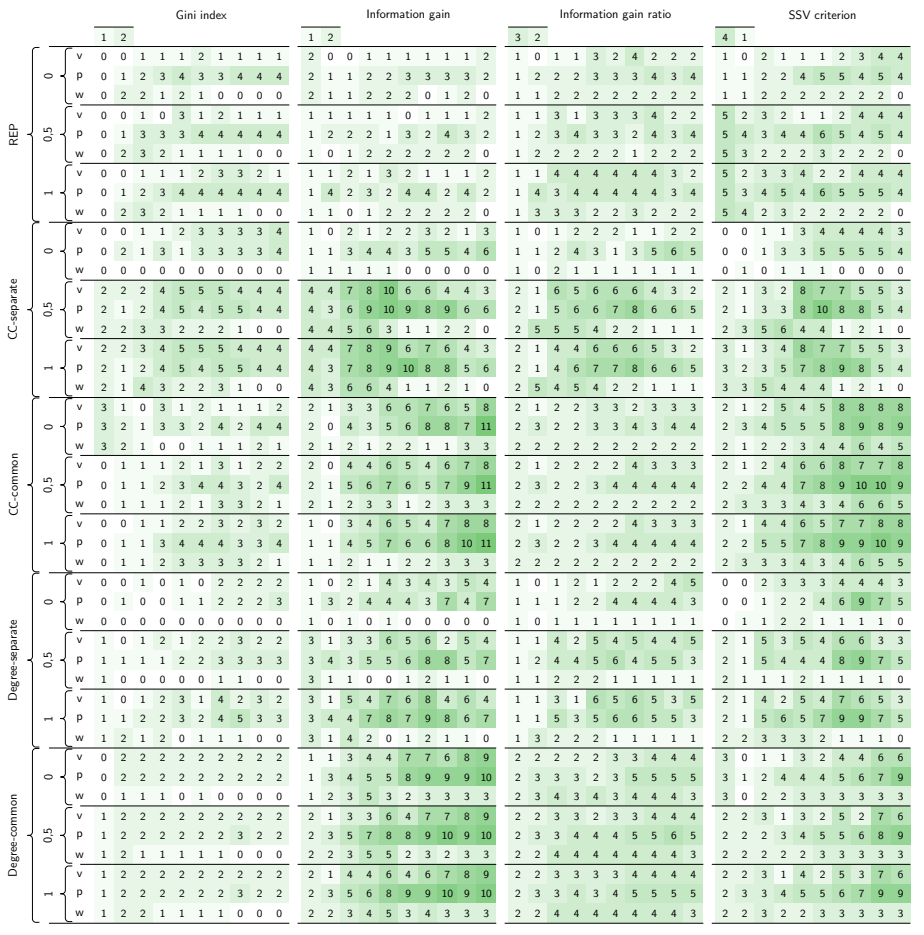
Such parameterization gives the total of 10*3*3*5*4 = 1800 different committee configurations. Standard (single tree) validated models were built for each of the 4 split criteria and for the two CV pruning methods: CC and degree-based.

The tests were performed for 21 different datasets from the UCI repository [2], summarized in table 1. The datasets are fully satisfactory for the sake of the tests, and they are probably the most popular, so they were chosen as the most adequate. Some datasets (of the repository) were not selected because they would need some preprocessing (for example to delete classes with very few examples) and that would spoil the clarity of

Table 1. Datasets used for the tests

Symbol	Dataset	classes	instances	features	ordered f.
APP	appendicitis	2	106	7	7
AUS	Australian credit	2	690	14	6
BRE	breast cancer (Wisconsin)	2	699	9	9
FLA	flag	8	194	28	10
GLA	glass	6	214	9	9
HEA	heart	2	303	13	13
IMA	image	7	2310	19	19
ION	ionosphere (trn+tst)	2	351	34	34
IRI	iris	3	150	4	4
KVK	kr-vs-kp	2	3196	36	0
LBR	Ljubjlana breast cancer	2	286	9	1
LET	letter recognition	26	20000	16	16
PIM	Pima indians diabetes	2	768	8	8
SON	sonar	2	208	60	60
SOY	soybean large	19	307	35	0
SPL	splice	3	3190	60	0
THY	thyroid (trn+tst)	3	7200	21	6
VOT	vote	2	435	16	0
VOW	vowel	6	871	3	3
WAV	waveform	3	5000	21	21
WIN	wine	3	178	13	13

Table 2. Numbers of insignificantly different results from the best ones (among the 21 datasets)



the tests. The mushroom data was rejected, because it is too easy for DT learning—all the algorithms would be 100% accurate with zero variance, so it would add no value to the comparison.

The first extensive test consisted in collecting test results of 10 repetitions of 10-fold cross-validation for each tested configuration. The means were calculated and the configuration of the best mean determined. Then, each of the 1808 configurations was compared to the best one with paired t-test with $\alpha = 0.01$. Table 2 presents a summary of the results: for each configuration, we can see the number of datasets (among the 21) for which the result was not significantly different from the best one. Apart from the numbers, the table shows the results by means of darkened background: the better result, the darker, so that it is easy to see the areas of high and low results. The table has four "large" columns presenting the results for different split criteria. Each one has two results in the first row, which correspond to single DTs resulting from validation with

cost-complexity and degree-based pruning respectively. Below the two, we can see the results of different CV committees in rows of 10 results corresponding to subsequent committee sizes (from 1 to 10). On the left, the rows are succinctly described: the letters "v", "p" and "w" denote the types of decision making within the committees: based on voting, probabilities and weights (see section 3 for explanations of the CV committees configuration). The groups of three rows represent the results for a fixed parameter of training data error factor (one of 0, 0.5 and 1). Finally we can see 5 groups of 9 rows each. They correspond to the validation methods used: REP, common and separate cross-complexity validation and common and separate degree-based pruning.

Provided table 2, we can draw interesting conclusions about the performance of different committee parameters with naked eye.

Committee sizes. To compare the performance of different committee sizes (members counts), the whole set of 1800 committee results got split into ten groups (corresponding to the sizes) and the groups compared with paired t-test. It must be pointed out (and it concerns all the following grouped results analyses), that the 600 results within each group are not quite statistically independent (some committees consist of the same trees, are tested on the same data, and differ only in a single parameter which does not always cause changes in the decisions), so the assumptions of the t-test are not necessarily satisfied. Anyway, the tested populations are large enough to show the differences we want to observe. The interpretation of the confidence level is probably violated, but it is not so important here.

The counts of insignificant differences (again $\alpha = 0.01$) from the committee with the best mean are 3, 3, 7, 9, 14, 14, 14, 12, 14, 8 (respectively for the committee sizes of 1,...,10). This lets as claim, that in general, the most adequate CV committees are built from 5 to 9 members.

Committee decision making. One of the easily visible conclusions is that the new type of committee decision making failed. The weighs derived from DT leaves, being the counts of the training vectors falling into the nodes, turned out to spoil the results rather than to improve them. A detailed analysis confirmed the weakness of such strategy: the data items lying close to representatives of other classes often fall into small leaves, when they are correctly classified and into large leaves, when they are dominated by the other classes and are misclassified by the tree. Therefore, after summing the weights, the negative answers dominate the positive ones and the overall committee classification deteriorates.

To compare the three ways of decision making (voting, probabilities, weights) the whole set of results was split into three corresponding groups, and the groups were compared with paired t-test (again $\alpha = 0.01$). The subsequent techniques reached insignificant differences with the best results for 4, 20 and 5 datasets respectively. Therefore, we can reliably claim that the probabilities perform the best. Although, the result seems to show that weighting performs better than voting, it is not so: the direct comparison of the results of the two groups reveal 16 significant wins of voting and just 3 significant wins of weighting. It is not an incompatibility of the results—in the first case the results were compared to those of the best mean, which is something different.

Training data error factors. In contrary to the idea of weighting, the idea of including training data error in the validation process is very advantageous. Since the tests have been done for the factors of 0, 0.5 and 1, we can compare the adequate three groups of corresponding results. Paired t-test with $\alpha = 0.01$ showed no significance difference from the best result in 7, 15 and 19 cases, respectively. This proves that the information about the training error is valuable in validation, and the factor of 1 is better than the factor of 0.5 (4 wins and no loss in direct competition).

Validation methods. Another interesting observations concern tree pruning. Splitting the results into adequate 5 groups and comparing them gives the numbers of 4, 6, 10, 4 and 11 insignificant differences between the subsequent validation methods and the one with the best mean. The first number tells that the REP is not very successful in the whole test. The next two pairs seem to provide evidence that the approaches of common pruning are better than their relatives with separate pruning. When we have a closer look at table 2, we can see that the common pruning approaches improve mainly the poor results, for example those obtained with weights. The improvement strongly affects the overall result, but it is not what we are interested in, because we do not need a validation method that works nice on average, but it is fully satisfactory if it works perfectly with some selected values of the remaining parameters. Therefore, a restricted comparison of the most attractive configurations is presented int the following subsection.

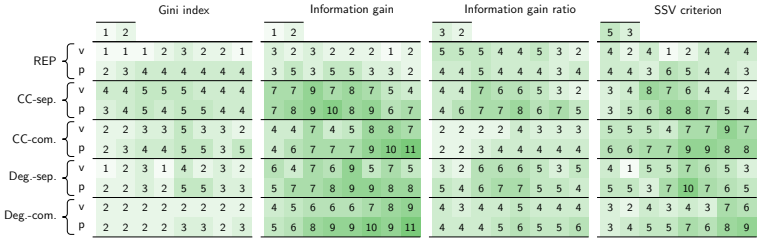
Some regularities about the validation methods can also be observed in table 2 with naked eye. Visual comparison of the result blocks corresponding to separate and common pruning lets us observe that with separate pruning, the best results are obtained with 3-8 committee members, while with common pruning, the best scores come when all 10 trees are included in the committee. It is very reasonable: common pruning optimizes collective decisions of all the trees, so it improves the poor trees, but may also reduce the accuracies of the best ones, while separate pruning accepts diverse tree accuracies, so it is better to take only the better trees to the committee.

Split criteria. Comparison of the split criteria is a bit aside the scope of this article, but a short analysis would not be harmful. The groups of results corresponding to Gini index, information gain index, information gain ratio and SSV separability obtained the scores of 3, 8, 5 and 8 insignificant differences with the methods obtaining maximum mean accuracy. The superiority of IG and SSV can also be seen with naked eye in the table. The weakness of IGR may be a surprise, but it must be pointed out, that the tests engaged a DT induction algorithms producing binary trees, so the arguments for IGR in relation to IG do not apply here.

4.1 Analysis of a Restricted Set of the Results

To prevent the poor results from affecting the conclusions drawn from the experiments, a restricted analysis has been done. The set of results was restricted by removing the results corresponding to the weights-based committees, to the committees of 1 or 2 members and the once using training error factor different than 1. The comparison of the remaining results to the ones of the best mean (among the kept configurations) are presented in table 3. In comparison to the table 2, some scores increased and some

Table 3. The results after removing the weakest ones



others decreased—both changes are possible, because in the case of a change of the best mean result to compare with, the statistical significance may change in both ways. Anyway, the changes are rare and small, which confirms, that the most interesting results were kept in the restricted set.

Comparing the groups of results corresponding to particular parameters is more reliable now (less statistical dependence), but in general, it confirms the conclusions presented above. The scores obtained by committee sizes of 3-10 are 5, 3, 7, 7, 11, 10, 8 and 8, suggesting 7 members as the best choice (on average). Voting compared to probabilities reaches just 1 significant wins and 18 significant losses, confirming the superiority of probabilities. The scores from comparison of the validation methods are 6, 13, 6, 6 and 5 and no longer point common pruning as better than separate, but still common pruning needs more committee members than separate. The split criteria scores are almost the same as before: 3, 9, 5 and 8.

4.2 Collective Decisions vs Single Validated Trees

The final analysis directly addresses the main subject of the paper, i.e. compares particular committees with single validated tree classifiers (not with the best results, as above). The presentation of the results is reduced to 320 of 1800 committee configurations. As in the previous analysis the less successful parameters were discarded. This time only the committees of 3-10 members are analyzed, training data errors are taken with the factor of 1 in the validation process and the decision module combines probabilities provided by the members. Table 4 illustrates the comparative results of each of the selected committee configuration with the classifiers based on single validated DTs. The left half depicts comparisons with the trees validated with the cross-complexity pruning and the right half—with degree-based pruning. The numbers in each cell are the numbers of significant wins, draws (neither competitor won) and significant losses respectively, estimated by the paired t-test with $\alpha = 0.01$. Apart from the numbers, there are bars in the background presenting the same information as the numbers but graphically. It is easy to see that most of the committees perform much better than single trees. Only the committees pruned with REP happen to be worse than single trees, but these are just several cases. In most cases the advantage of committees is overwhelming, up to the score of 17 wins, 4 draws and 0 losses.

Table 4. Comparison of CV committees with single validated tree classifiers

		vs single CC					vs single Deg.				
		REP	CC-s.	CC-c.	Deg.-s.	Deg.-c.	REP	CC-s.	CC-c.	Deg.-s.	Deg.-c.
Gini index	3	7-9-5	13-7-1	11-9-1	11-7-3	8-10-3	9-8-4	12-8-1	13-8-0	12-9-0	12-9-0
	4	10-6-5	15-5-1	10-11-0	14-4-3	8-11-2	10-7-4	14-7-0	13-8-0	15-6-0	13-8-0
	5	10-6-5	15-6-0	14-6-1	13-6-2	8-11-2	11-6-4	15-6-0	14-7-0	15-6-0	14-7-0
	6	10-7-4	15-6-0	12-9-0	13-6-2	9-10-2	11-7-3	15-6-0	13-8-0	17-4-0	14-6-1
	7	11-5-5	15-6-0	14-6-1	13-6-2	9-10-2	11-7-3	14-7-0	13-8-0	15-6-0	15-6-0
	8	11-5-5	15-6-0	13-8-0	13-5-3	10-9-2	12-6-3	13-8-0	13-8-0	14-7-0	16-5-0
	9	10-6-5	13-7-1	14-6-1	11-8-2	10-9-2	12-6-3	13-8-0	14-7-0	14-6-1	16-5-0
	10	9-8-4	13-7-1	14-6-1	11-7-3	11-8-2	11-7-3	12-9-0	13-8-0	14-6-1	15-6-0
Information gain	3	6-11-4	11-10-0	7-14-0	9-9-3	7-11-3	7-10-4	13-8-0	8-12-1	10-10-1	10-11-0
	4	8-9-4	13-8-0	11-10-0	12-8-1	9-10-2	6-11-4	13-8-0	10-10-1	12-9-0	13-8-0
	5	7-10-4	14-7-0	12-9-0	12-8-1	8-11-2	8-9-4	12-9-0	9-11-1	13-8-0	11-10-0
	6	8-9-4	14-6-1	11-10-0	12-7-2	8-11-2	9-8-4	13-8-0	10-11-0	13-8-0	14-7-0
	7	9-8-4	15-5-1	12-8-1	13-6-2	10-9-2	8-9-4	15-6-0	10-11-0	14-7-0	16-5-0
	8	8-9-4	15-5-1	13-7-1	12-7-2	8-11-2	8-9-4	13-8-0	11-10-0	14-7-0	14-7-0
	9	9-8-4	14-6-1	12-8-1	10-9-2	10-9-2	8-8-5	12-9-0	12-9-0	12-8-1	12-8-1
	10	8-8-5	12-7-2	13-7-1	9-10-2	10-9-2	8-8-5	12-8-1	13-7-1	10-10-1	13-8-0
Information gain ratio	3	7-8-6	14-7-0	10-10-1	11-9-1	10-9-2	9-8-4	13-7-1	10-9-2	15-3-3	11-10-0
	4	9-6-6	14-7-0	12-8-1	11-9-1	9-12-0	9-8-4	15-6-0	10-10-1	15-5-1	14-7-0
	5	9-6-6	15-6-0	11-9-1	15-5-1	11-10-0	9-8-4	15-6-0	12-8-1	17-3-1	13-8-0
	6	10-6-5	17-4-0	15-6-0	15-5-1	13-8-0	10-7-4	16-5-0	12-8-1	17-4-0	15-6-0
	7	10-6-5	16-4-1	16-5-0	15-5-1	14-7-0	10-7-4	14-7-0	12-8-1	17-3-1	15-6-0
	8	11-5-5	16-4-1	15-6-0	15-5-1	14-7-0	9-8-4	15-6-0	13-7-1	15-6-0	15-6-0
	9	11-5-5	14-6-1	15-5-1	13-7-1	14-7-0	9-8-4	14-6-1	13-7-1	15-5-1	15-6-0
	10	9-7-5	14-6-1	15-6-0	12-8-1	14-7-0	8-9-4	12-8-1	13-6-2	14-6-1	16-5-0
SSV	3	6-8-7	9-11-1	7-14-0	7-11-3	6-13-2	9-5-7	12-8-1	10-11-0	12-7-2	8-13-0
	4	6-7-8	10-10-1	8-12-1	7-11-3	8-11-2	9-4-8	11-8-2	12-9-0	12-7-2	11-10-0
	5	7-6-8	13-6-2	13-8-0	10-9-2	10-9-2	9-5-7	15-4-2	12-9-0	14-5-2	12-9-0
	6	9-4-8	14-5-2	12-8-1	12-8-1	10-9-2	8-6-7	16-3-2	13-7-1	15-4-2	10-11-0
	7	8-5-8	14-5-2	12-8-1	12-8-1	11-8-2	12-2-7	16-3-2	14-5-2	15-4-2	13-8-0
	8	8-5-8	13-6-2	11-9-1	12-8-1	11-8-2	10-4-7	15-4-2	13-6-2	14-5-2	13-8-0
	9	5-8-8	12-8-1	13-8-0	12-7-2	13-6-2	10-4-7	12-7-2	14-6-1	12-7-2	14-6-1
	10	6-7-8	11-8-2	12-9-0	10-9-2	12-7-2	8-6-7	12-7-2	13-8-0	13-6-2	15-5-1

5 Summary

Decision trees are very attractive models, especially when their comprehensiveness is accompanied by high accuracies. It is very common to validate decision trees with CV-based techniques like cross-complexity pruning. In this paper, many different configurations of CV-based committees of DTs have been examined. Old and new techniques of combining DT decisions into collective ones have been presented and evaluated. Many interesting conclusions have emerged from the results analysis and visualization.

A general conclusion may be formulated, that DT committees may be highly more accurate than single validated trees with no additional effort at learning time. Conversely: building committees is computationally cheaper because no final DT needs to be induced. Classification of new data is not very expensive with DTs, so it is not an effort to classify data with 5 or 7 trees and combine the decisions. Also the comprehensibility of such complex models is not restricted. Sometimes it is even desirable to be supplied with several alternative sets of classification rules, because they may point to different aspects of the classification problem and, this way, they provide higher probability of finding the most important premises of particular classification decisions.

Very advanced tests have been quite easily performed thanks to the versatility and rich test tools of the Intemi system. Thanks to the machine unification mechanism, testing 1808 complex learning machines with 10 repetitions of 10-fold cross-validation was conducted in quite short time, because the decision trees underlying the committees were not built repeatedly for each committee, but were unified and reused.

The results, presented here, include very precious meta-knowledge. According to them, advanced search processes may omit testing some parameter settings without significant loss in their final achievements. Our meta-learning approaches will certainly comply with the meta-knowledge to save time of further explorations. More conclusions of this kind will bring more and more accurate learning machines, resembling human experts in not loosing time for doubtful approaches, but more thorough and systematic.

Acknowledgments. The research is supported by the Polish Ministry of Science with a grant for years 2010–2012.

References

1. Breiman, L., Friedman, J.H., Olshen, A., Stone, C.J.: Classification and regression trees. Wadsworth, Belmont (1984)
2. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
3. Grąbczewski, K.: Separability of split value criterion with weighted separation gains. LNCS. Springer, Heidelberg (submitted, 2011)
4. Grąbczewski, K., Duch, W.: A general purpose separability criterion for classification systems. In: Proceedings of the 4th Conference on Neural Networks and Their Applications, Zakopane, Poland, pp. 203–208 (June 1999)
5. Grąbczewski, K., Duch, W.: The Separability of Split Value criterion. In: Proceedings of the 5th Conference on Neural Networks and Their Applications, Zakopane, Poland, pp. 201–208 (June 2000)
6. Grąbczewski, K., Jankowski, N.: Saving time and memory in computational intelligence system with machine unification and task spooling. Knowledge-Based Systems, Corrected Proof (in Press 2011), http://www.sciencedirect.com/science/article/B6V0P-51YBTF9-1/2/93db4bd34b2d02f8be135ff200e8f61d
7. Loh, W.Y., Shih, Y.S.: Split selection methods for classification trees. Statistica Sinica 7, 815–840 (1997)
8. Mingers, J.: An empirical comparison of pruning methods for decision tree induction. Machine Learning 4(2), 227–243 (1989)
9. Quinlan, J.R.: Simplifying decision trees. Int. J. Man-Mach. Stud. 27(3), 221–234 (1987)
10. Quinlan, J.R.: Programs for machine learning (1993)

Time and Personality Dependent Behaviors for Agent Negotiation with Incomplete Information

Amine Chohra, Arash Bahrammirzaee, and Kurosh Madani

Images, Signals, and Intelligent Systems Laboratory (LISSI / EA 3956), Paris-East University (UPEC), Senart Institute of Technology, Avenue Pierre Point, 77127 Lieusaint, France
{chohra,bahrammirzaee,madani}@u-pec.fr

Abstract. Finding adequate negotiation strategy with incomplete information, even in one to one negotiation, is a complex problem. Inspired from research works aiming to analyze human behavior and those on social negotiation psychology, integration of psychological aspects, with essential time parameter, is becoming necessary. For this purpose, first, one to one bargaining process, in which a buyer agent and a seller agent negotiate over single issue (price), is developed, where social and cognitive behaviors based on time (Faratin et al. 1998) and personality aspects are suggested. Second, experimental environments and measures, allowing a set of experiments, carried out for different negotiation deadlines, are detailed. Third, experimental results are analyzed with regard to time dependent behaviors. Results demonstrate that more increasing conciliatory aspects lead to increased agreement point (price) and decreased agreement time, and more increasing aggressive aspects lead to decreased agreement point and increased agreement time.

Keywords: Negotiation model, incomplete information, social and cognitive systems, time and personality dependent behaviors, decision-making.

1 Introduction

Interesting surveys on negotiation models in the AI field are given in [1], [2], [3]. Elsewhere, Lomuscio *et al.* [4] identified the main parameters on which any automated negotiation depends and provided a classification scheme for negotiation models. The environment that a negotiator is situated in greatly impacts the course of negotiation actions. Instead of focusing on analyzing the strategy equilibrium and historical information as in game theory, AI researchers are interested in designing adaptive negotiation agents, with incomplete information, to environment changes. Agents have incomplete and uncertain information about each other, and each agent's information (e.g., deadline, utility function, strategy, …) is its private knowledge.

An important research work has been developed by Faratin *et al.* [5] which devised a negotiation model that defines a range of strategies and behaviors for generating proposals based on time, resource, and behaviors of negotiators. By another way, in the research works developed aiming to analyze and describe human behavior in [6], twelve categories representing three major behavior parts have been defined: positive

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 352–362, 2011.

socio-emotional part, a neutral task part, and negative socio-emotional part. In another side, in research works on the social negotiation psychology of Rubin and Brown developed in [7], the interpersonal orientation of a person has an influence on his negotiating behavior. It is predominantly concerned with the degree of a person's responsiveness. Responsive people are more co-operative and therefore expect positive results. Personality type should therefore be determined first to obtain the best results in negotiation. Thus, negotiation behaviors, in which characters such as conciliatory, neutral, or aggressive define a 'psychological' personality aspect of a negotiator, play an important role.

Negotiations have received wide attention from distributed Artificial Intelligence (AI) community [8] and In general, any negotiation settings will have four different components [9]: 1) a negotiation set, space of possible proposals that agents can make ; 2) a protocol, legal proposals that agents can make ; 3) a collection of strategies, one for each agent, which determine what proposals agents will make ; and 4) an agreement rule that determines reach agreement stopping negotiation.

Negotiation usually proceeds in a series of rounds, with every agent making a proposal at every round. A particular source of complexity is the agent number involved, and way in which these agents interact [9]: one to one, many to one (can be treated as concurrent one to one negotiations), and many to many (hard to handle).

This paper deals with social and cognitive negotiation behaviors for autonomous agents with incomplete information in order to find the adequate (win-win solutions for both parties) negotiation strategy in one to one negotiation which is a complex problem. Inspired from research works aiming to analyze human behavior and those on social negotiation psychology, the integration of psychological aspects of the agent personality, with the essential time parameter, is becoming necessary. In fact, behaviors, in which characters such as conciliatory or aggressive define a 'psychological' personality aspect, play an important role in real negotiation.

The aim of this paper is then to analyze the psychological personality impacts (effects) on the negotiation particularly with regard to agreement point and agreement time. Thus, an important aspect of the analysis of the suggested negotiation model is to assess the variation consequences of different psychological agent characters on the decisions agents make. For this purpose, first, a one to one bargaining process, in which a buyer agent and a seller agent negotiate over a single issue (price), is developed, where social and cognitive behaviors based on time (Faratin *et al.* [5]) and personality psychological aspects are suggested. Second, experimental environments and measures, allowing a set of experiments, carried out for different negotiation deadlines of buyer and seller agents, are detailed. Third, experimental results of time-psychology dependent behaviors are analyzed with regard to time dependent behaviors for different time deadlines.

2 One to One Negotiation

In this Section, one to one bargaining process shown in Fig. 1, in which buyer and seller agents negotiate, over a single issue (price), is developed.

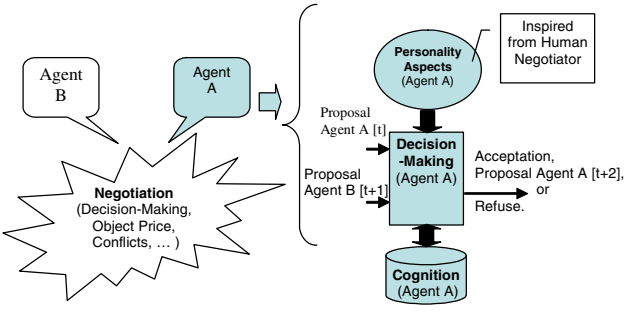


Fig. 1. Bilateral negotiation implying an autonomous agent

2.1 Negotiation Set

A negotiation set is the space of possible proposals that agents can make. The negotiation set (objects): the range of issues over which an agreement must be reached. Let i represents the negotiating agents, in bargaining bilateral negotiation $i \in \{\text{buyer(b), seller(s)}\}$, and j the issues under negotiation, in single issue negotiation j = *price*. The value for issue *price* acceptable by each agent i is $x^i \in [\min^i, \max^i]$.

2.2 Negotiation Protocol

A protocol is the legal proposals that agents can make. The process of negotiation can be represented by rounds, where each round consists of an offer from agent b (buyer) at time t_1 and a counter-offer from an agent s (seller) at time t_2. Then, a negotiation consists in a sequence of rounds: round1 (t_1, t_2), round2 (t_3, t_4), … Thus, for a negotiation between agents b and s, and if agent b starts first, then it should offer in times (t_1, t_3, t_5, …, t^b_{max}), and agent s provides counter-offers in (t_2, t_4, t_6, …, t^s_{max}), where t^b_{max} and t^s_{max} denote negotiation deadline for agents b and s, respectively. Note that the three different deadline cases are allowed: 1) $t^b_{max} > t^s_{max}$, where considered deadline is $T_{max} = t^s_{max}$; 2) $t^b_{max} = t^s_{max}$, where considered deadline is $T_{max} = t^b_{max} = t^s_{max}$; 3) $t^b_{max} < t^s_{max}$, where considered deadline is $T_{max} = t^b_{max}$.

For agent b, the proposal to offer or accept is within interval $[\min^b, \max^b]$, where $\max^b$ is the buyer reservation price in negotiation thread, and $\min^b$ is the lower bound of a valid offer. Similarly, for agent s, the proposal to offer or accept is within interval $[\min^s, \max^s]$, where $\min^s$ is the seller reservation price and $\max^s$ is the upper bound of a valid offer. Initially a negotiator offers most favorable value for himself: agent b starts with $\min^b$ and agent s starts with $\max^s$. If proposal is not accepted, a negotiator concedes with time proceeding and moves toward other end of the interval.

2.3 Negotiation Behaviors

The paces of concession depend on the negotiation behaviors of agent b and agent s which are characterized by negotiation decision functions. For negotiation strategies, time t is one of predominant factors used to decide which value to offer next.

Time Dependent Behaviors: Time dependent functions are used as negotiation decision functions varying the acceptance value (price) for the offer depending on the remaining negotiation time (an important requirement in negotiation), i.e., depending on t and t^b_{max} for agent b and depending on t and t^s_{max} for agent s. Thus, proposal $x^b[t]$ to be offered by agent b and the one $x^s[t]$ to be offered by agent s at time t, with $0 <= t <= t^i_{max}$ belonging to [0, T - 1], are as follows. The proposal $x^s[t]$ to be offered by agent s at time t, with $0 <= t <= t^s_{max}$ belonging to [0, T - 1], is defined by Eq. (1).

$$x^s[t] = min^s + (1 - \alpha^s(t))\,(max^s - min^s),$$

where $\alpha^s(t)$ are time-dependent functions ensuring that: $0 <= \alpha^s(t) <= 1$, (1)

$$\alpha^s(0) = K^s \text{ (positive constant) and } \alpha^s(t^s_{max}) = 1.$$

Such $\alpha^s(t)$ functions can be defined in a wide range according to the way in which $\alpha^s(t)$ is computed (the way they model the concession), e.g., polynomial in Eq. (2).

$$\alpha^s(t) = K^s + (1 - K^s)\left(\frac{min(t, t^s_{max})}{t^s_{max}}\right)^{\frac{1}{\beta}}. \quad (2)$$

Indeed, the constant $\beta > 0$ determines the concession pace along time, or convexity degree of the offer curve as a function of the time. By varying β a wide range of negotiation behaviors can be characterized: Boulware (B) with $\beta < 1$ and Conceder (C) with $\beta > 1$ [3], and the particular case of Linear (L) with $\beta = 1$.

Social and Cognitive Behaviors: The proposal $x^b[t]$ to be offered by agent b at time t, with $0 <= t <= t^b_{max}$ belonging to [0, T - 1], is defined using behaviors based on time and personality psychological aspects of a negotiator agent detailed in Sect. 2. 4.

2.4 Negotiation Strategies

A collection of strategies, one for each agent, which the role is to determine what proposals agents will make (which behavior should be used at any one instant).

Time-Dependent: During a negotiation *thread* (the sequence of rounds with offers and counter-offers in a two-party negotiation), a negotiation strategy based on time dependent behaviors defined in [5] consists to define the way in which such behaviors are used. In this paper, each strategy uses individually the behaviors Boulware (B), Linear (L), or Conceder (C) during a negotiation thread.

Parameter β ranges [10], [11] are defined as: $\beta 1 \in [20.00, 40.00]$ for Conceder (C) or *Conciliatory (Con)*; $\beta 2 = 1.00$ for Linear (L) or *Neutral (Neu)*, $\beta 3 \in [0.01, 0.20]$ for Boulware (B) or *Aggressive (Agg)*. Then, constants K^i are chosen as small positive $K^i = 0.1$, for s, in order to not constrain the behavior of each time dependent function.

Time-Psychology Dependent: These strategy behaviors integrate time and personality aspects and it is expected from such strategy following hypothesis:

Hypothesis. The suggested strategy is expected to integrate time and personality aspects such that more increasing Conciliatory character leads to increasing agreement point and decreasing agreement time; and more increasing Aggressive character leads to decreasing agreement point and increasing agreement time.

Thus, such strategy is detailed from buyer point of view, where seller offers first.

Step 1 (Computing First Offers). The agent proposal is obtained Eq. (3) from conciliatory part Eq. (4), neutral part Eq. (5), and aggressive part Eq. (6).

$$x_j^b[t] = (W_{Con}^b[t] * xCon_j^b[t]) + (W_{Neu}^b[t] * xNeu_j^b[t]) + (W_{Agg}^b[t] * xAgg_j^b[t]) \tag{3}$$

where the sum of weights is $W_{Con}^b[t] + W_{Neu}^b[t] + W_{Agg}^b[t] = 1$,

$$xCon_j^b[t] = \min_j^b + \alpha Con_j^b[t](\max_j^b - \min_j^b),$$

$$\text{where } \alpha Con_j^b[t] = k_j^b + (1 - k_j^b)\left(\frac{\min(t, t_{\max}^b)}{t_{\max}^b}\right)^{\frac{1}{\beta 1}}, \tag{4}$$

$$xNeu_j^b[t] = \min_j^b + \alpha Neu_j^b[t](\max_j^b - \min_j^b),$$

$$\text{where } \alpha Neu_j^b[t] = k_j^b + (1 - k_j^b)\left(\frac{\min(t, t_{\max}^b)}{t_{\max}^b}\right)^{\frac{1}{\beta 2}}, \tag{5}$$

$$xAgg_j^b[t] = \min_j^b + \alpha Agg_j^b[t](\max_j^b - \min_j^b)$$

$$\text{where } \alpha Agg_j^b[t] = k_j^b + (1 - k_j^b)\left(\frac{\min(t, t_{\max}^b)}{t_{\max}^b}\right)^{\frac{1}{\beta 3}}. \tag{6}$$

Step 2 (Predicting β of the Seller). b predicts βs in Eq. (7) of s by Eq. (1) and (2).

Step 3 (Character To Change from Predicted β). According to the result of the predicted βs, the buyer changes the corresponding character such as in Eq. (8).

$$\beta s = \frac{Ln\dfrac{t-1}{t_{\max}^b}}{Ln\dfrac{\alpha_j^s[t] - k_j^b}{1 - k_j^b}} \quad \text{where } \alpha_j^s(t_i) = \frac{x_j^s[t-1] - \min_j^b}{x_j^s[0] - \min_j^b}. \tag{7}$$

$$\text{If } \beta s > 1 \text{ Then Character Con Changes:}$$
$$NewxCon_j^b[t] = \min_j^b + New\alpha Con_j^b[t](\max_j^b - \min_j^b),$$
$$New\alpha Con_j^b[t] = k_j^b + (1 - k_j^b)(\frac{\min(t, t_{\max}^b)}{t_{\max}^b})^{\frac{1}{\beta s}}.$$
$$\text{If } \beta s = 1 \text{ Then Character Neu Changes:}$$
$$NewxNeu_j^b[t] = \min_j^b + New\alpha Neu_j^b[t](\max_j^b - \min_j^b),$$
$$New\alpha Neu_j^b[t] = k_j^b + (1 - k_j^b)(\frac{\min(t, t_{\max}^b)}{t_{\max}^b})^{\frac{1}{\beta s}}. \quad (8)$$
$$\text{If } \beta s > 1 \text{ Then Character Agg Changes:}$$
$$NewxAgg_j^b[t] = \min_j^b + New\alpha Agg_j^b[t](\max_j^b - \min_j^b),$$
$$New\alpha Agg_j^b[t] = k_j^b + (1 - k_j^b)(\frac{\min(t, t_{\max}^b)}{t_{\max}^b})^{\frac{1}{\beta s}}.$$

Step 4 (Computing DeltaCharacter). According to the result of the character to change, the buyer computes the DeltaCharacter DC^b *as follows*:

$$\text{If Character To Change is Con: } DC^b = \frac{NewxCon_j^b[t] - xCon_j^b[t]}{xCon_j^b[t]}.$$
$$\text{If Character To Change is Neu: } DC^b = \frac{NewxNeu_j^b[t] - xNeu_j^b[t]}{xNeu_j^b[t]}. \quad (9)$$
$$\text{If Character To Change is Agg: } DC^b = \frac{NewxAgg_j^b[t] - xAgg_j^b[t]}{xAgg_j^b[t]}.$$

Step 5 (Weight Updating). According character to change, buyer updates:

$$\text{If Character To Change is Con: } W_{Con}^b[t] = W_{Con}^b[t-1] + DC^b[t],$$
$$W_{Neu}^b[t] = W_{Neu}^b[t-1] - 0.3 * DC^b[t],\ W_{Agg}^b[t] = W_{Agg}^b[t-1] - 0.7 * DC^b[t].$$
$$\text{If Character To Change is Neu: } W_{Con}^b[t] = W_{Con}^b[t-1] - 0.5 * DC^b[t],$$
$$W_{Neu}^b[t] = W_{Neu}^b[t-1] + DC^b[t],\ W_{Agg}^b[t] = W_{Agg}^b[t-1] - 0.5 * DC^b[t]. \quad (10)$$
$$\text{If Character To Change is Agg: } W_{Con}^b[t] = W_{Con}^b[t-1] - 0.7 * DC^b[t],$$
$$W_{Neu}^b[t] = W_{Neu}^b[t-1] - 0.3 * DC^b[t], W_{Agg}^b[t] = W_{Agg}^b[t-1] + DC^b[t].$$

Step 6 (Computing the Proposal). According the character to change, b updates:

$$x_j^b[t] = (W_{Con}^b[t] * xCon_j^b[t]) + (W_{Neu}^b[t] * xNeu_j^b[t]) + (W_{Agg}^b[t] * xAgg_j^b[t]) . \quad (11)$$

2.5 Agreement Rule

An agreement rule determines the reach agreements stopping negotiation. Agent b accepts an offer $x^s[t]$ from agent s at time t if it is not worse than the offer he would submit in next step, i.e., only if the relation given in Eq. (11) is satisfied. Similarly, s accepts an offer $x_b[t]$ from b at time t only if the relation given in Eq. (12) is satisfied.

$$\begin{cases} x^b(t+1) >= x^s(t) \\ t <= T_{max} \end{cases}, \begin{cases} x^s(t+1) <= x^b(t) \\ t <= T_{max} \end{cases} . \quad (12)$$

3 Experiments: Environments and Measures

In this Section, experimental environments and measures are presented and a set of experiments, carried out for different deadlines of agents b and s, are detailed.

3.1 Experimental Environments

Environments are defined in bargaining bilateral negotiation between buyer(b) and seller(s), in single issue negotiation j = *price*. The experimental environment is defined by the following variables [t^b_{max} , t^s_{max} , T_{max}, K^b, K^s, min^b, max^b, min^s, max^s].

The negotiation interval (difference between minimum and maximum values of agents) for price is defined using: θ^i (length of the reservation interval for an agent i) and Φ (degree of intersection between the reservation intervals of the agents, ranging between 0 for full overlap and 0.99 for virtually no overlap). In the experimental environment: θ^i are randomly selected between the ranges [10, 30] for both agents, and $\Phi = 0$. The negotiation intervals are then computed, setting $min^b = 10$, by:

$$min^b = 10 , \; max^b = min^b + \theta^b , \; min^s = \theta^b \Phi + min^b , \text{ and } max^s = min^s + \theta^s . \quad (13)$$

The analysis and evaluation of negotiation behaviors and strategies developed in [12], indicated that negotiation deadlines significantly influence the negotiation performance. From this, the experimental environment is defined from random selection of the round number within [10, 50] which corresponds to a random selection of T_{max} within [20, 100]. Initiator of an offer is randomly chosen because the agent which opens the negotiation fairs better, irrespective of whether agent is b or s.

3.2 Experimental Measures

To produce statistically meaningful results the precise set of environments is sampled from parameters specified in Sect. 3.1 and environment number used is N = 200, in

each experiment. This ensures that the probability of the sampled mean deviating by more than 0.01 from true mean is less than 0.05.

Average Round Number, rounds to reach an agreement (deal), lengthy negotiation incurs penalties for resource consumption, thus shrinking utilities obtained by negotiators indirectly [13]. Average round number AR is given in Eq. (14):

$$AR = \frac{\sum_{n=1}^{N} R_D[n]}{N_D}, \text{ where } R_D \text{ is the number of rounds, for each environment} \quad (14)$$

with deal, and N_D is the number of environments with deals.

4 Experimental Results of Social and Cognitive Behaviors

In this Section, experimental results of the social and cognitive negotiation behaviors (time-psychology dependent) represented on figures by X (varying curves) are presented, analyzed, and compared for different deadlines with regard to time dependent behaviors where both agents b and s use Linear (time dependent behaviors): Linear-Linear represented on figures by L-L (constant curves).

The results presented in Fig. 2 and Fig. 3 concern the variation effects of the Conciliatory character of the buyer on negotiation behaviors. For both deadlines (short and long), results demonstrates that more Conciliatory character is increasing implies more agreement point is increasing while more agreement time is decreasing.

The results presented in Fig. 4 and Fig. 5 concern the variation effects of the Neutral character of the buyer (i.e., of the personality psychological aspects) on negotiation behaviors. For both deadlines (short and long), results demonstrates that more Neutral character is increasing implies more agreement point is decreasing while more agreement time is increasing (for long term deadline).

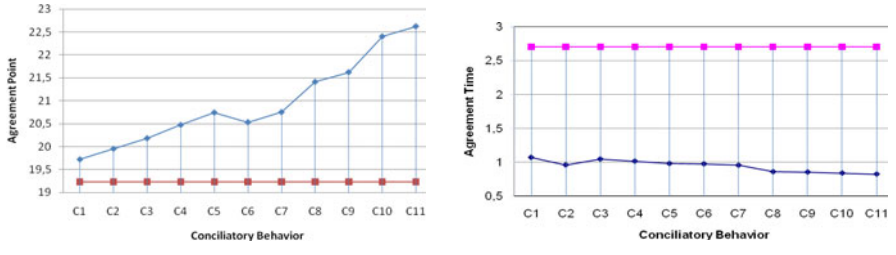


Fig. 2. Conciliatory behaviors (short term deadlines): agreement point and agreement time

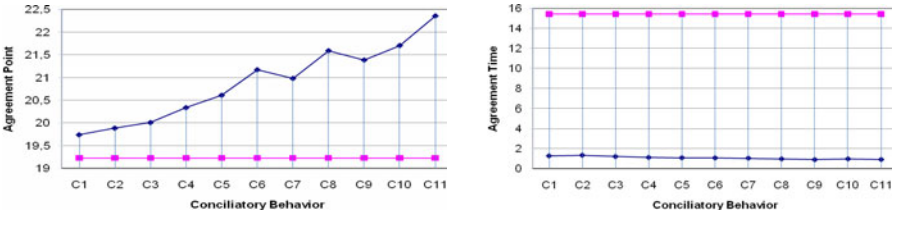


Fig. 3. Conciliatory behaviors (long term deadlines): agreement point and agreement time

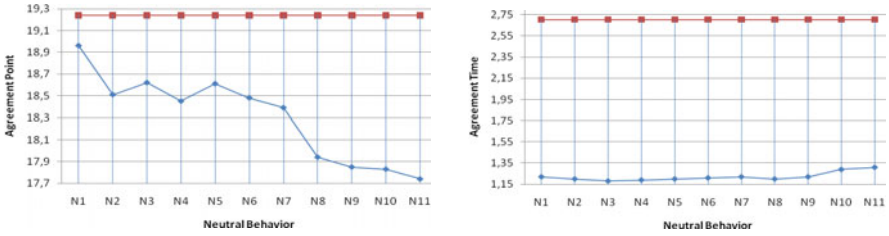


Fig. 4. Neutral behaviors (short term deadlines): agreement point and agreement time

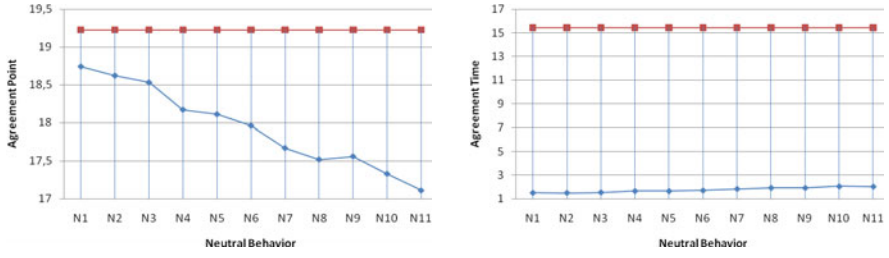


Fig. 5. Neutral behaviors (long term deadlines): agreement point and agreement time

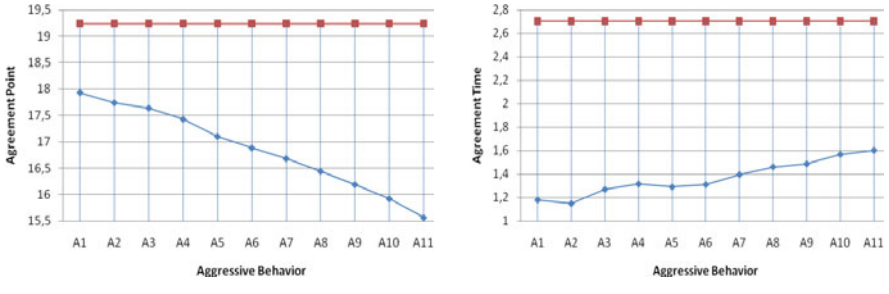


Fig. 6. Aggressive behaviors (short term deadlines): agreement point and agreement time

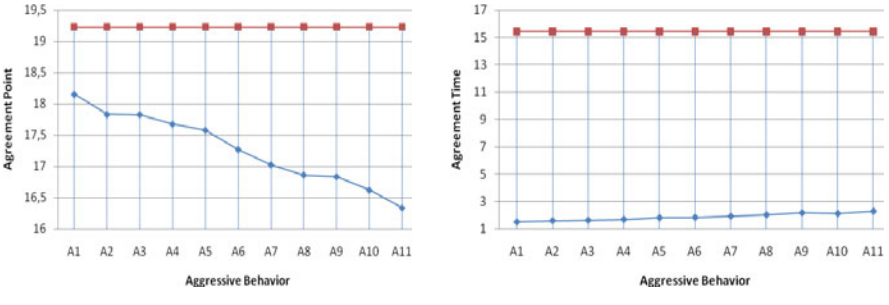


Fig. 7. Aggressive behaviors (long term deadlines): agreement point and agreement time

The results presented in Fig. 6 and Fig. 7 concern the variation effects of the Aggressive character of the buyer (i.e., of the personality psychological aspects) on negotiation behaviors. For both deadlines (short and long terms), these results demonstrates that more Aggressive character is increasing implies more agreement point is decreasing while more agreement time is increasing.

5 Conclusion

In this paper, *social* and *cognitive* negotiation behaviors (time-psychology dependent) have been suggested for autonomous agents with *incomplete* information in one to one single issue (price) intending to find adequate (*win-win* solutions) strategy. Results demonstrate, more increasing conciliatory aspects lead to increased agreement point (price) and decreased agreement time. On the other hand, more increasing aggressive aspects lead to decreased agreement point and increased agreement time.

Now, a central question that emerges is how to control personality aspects to the benefit of the negotiator agent in order to improve negotiation process in terms of agent utilities, round number to reach an agreement, and agreement percentage ?

Another important and necessary step is to integrate fuzzy reasoning to the suggested time-psychology dependent behaviors [14]. Afterwards, learning from interaction, which is fundamental from embodied cognitive science and understanding natural intelligence perspectives [15], [16] for understanding human behaviors and developing new solution concepts [17], will be necessary in negotiation.

References

1. Jennings, N.R., et al.: Automated Negotiation: Prospects, Methods, and Challenges. Int. J. of Group Decision and Negotiation 10(2), 199–215 (2001)
2. Gerding, E.H., van Bragt, D., Poutré, J.L.: Scientific Approaches and Techniques for Negotiation: A Game Theoretic and Artificial Intelligence Perspective. Tech. Report (2000)
3. Li, C., Giampapa, J., Sycara, K.: Bilateral Negotiation Decisions with Uncertain Dynamic Outside Options. IEEE Trans. on Systems, Man, and Cybernetics, Part C 36(1), 1–13 (2006)
4. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A classification Scheme for Negotiation in Electronic Commerce. Int. J. of Gro. and Neg. 12, 31–56 (2003)
5. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation Decision Functions for Autonomous Agents. Int. J. of Robotics and Autonomous Systems 24(3-4), 159–182 (1998)
6. Bales, R.F.: Interaction Process Analysis: A Method for the Study of Small Groups. Addisson-Wesley, Cambridge (1950)
7. Rubin, J.Z., Brown, B.R.: The Social Psychology of Bargaining and Negotiation. Academic Press, New York (1975)
8. Rosenschein, J., Zlotkin, G.: Rules of Encounter. MIT Press, Cambridge (1994)
9. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, Chichester (2002)
10. Pruitt, D.: Negotiation Behavior. Academic Press, London (1981)

11. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
12. Wang, K.-J., Chou, C.-H.: Evaluating NDF-Based Negotiation Mechanism Within an Agent-Based Environment. Robotics and Aut. Systems 43, 01–27 (2003)
13. Lee, C.-F., Chang, P.-L.: Evaluations of Tactics for Automated Negotiations. Group Decision and Negotiation 17(6), 515–539 (2008)
14. Richter, J., Kowalczyk, R., Klusch, M.: Multistage Fuzzy Decision Making in Bilateral Negotiation with Finite Termination Times. In: Nicholson, A., Li, X. (eds.) AI 2009. LNCS (LNAI), vol. 5866, pp. 21–30. Springer, Heidelberg (2009)
15. Pfeifer, R., Scheier, C.: Understanding Intelligence. MIT Press, Cambridge (1999)
16. Chohra, A.: Embodied Cognitive Science, Intelligent Behavior Control, Machine Learning, Soft Computing, and FPGA Integration: Towards Fast, Cooperative and Adversarial Robot Team (RoboCup). Technical GMD Report (136), Germany (June 2001) ISSN 1435-2702
17. Zeng, D., Sycara, K.: Benefits of Learning in Negotiation. In: Proc. of the 14th National Conference on Artificial Intelligence (AAAI 1997), Providence, RI, pp. 36–41 (1997)

Dynamic Selection of Negotiation Protocol in Multi-agent Systems for Disaster Management

Amelia Bădică, Costin Bădică, Sorin Ilie, Alex Muscar, and Mihnea Scafeş

University of Craiova, Bvd.Decebal 107, 200440, Craiova, Romania

Abstract. Collaboration between human experts during distributed problem solving for disaster management requires advanced self-configuration mechanisms for selection of potentially good service providers. Self-configuration is based on a framework for development of cooperative multi-issue one-to-many service negotiations. The framework allows the definition of: negotiation protocols, negotiation subjects composed of multiple negotiation issues, properties of negotiation issues, and utility functions of participant stakeholders. We show how this framework can be applied for service contracting in a disaster management information system. We provide initial experimental results concerning the effect of dynamic selection of negotiation protocol onto the quality of the negotiation outcome and the communication complexity of negotiation interactions.

Keywords: multi-agent system, service negotiation, contract net.

1 Introduction

The increasing complexity of real-world problems demands special support for distributed collaborative problem solving. Multi-agent systems (MAS) are a special class of distributed systems that combine interaction and coordination with distribution of computation to improve performance of problem solving processes. MAS were successfully applied for solving problems that require distributed reasoning, decentralization and collaboration. An example is a collaboration system for helping human experts and population to deal with disasters (see the FP7 DIADEM project[1] that targets crisis management in the context of chemical incidents in industrial and urban areas).

The Dynamic Process Integration Framework (DPIF) [6] is the underlying service-oriented MAS on which the DIADEM system is based. In the DPIF, communication links between local processes in agents are facilitated using service discovery: whenever an agent supplying some service (we will call this agent the *manager*) in a workflow requires data provided by some other service (we will call this agent the *contractor*), a communication link must be established between them.

[1] DIADEM Distributed information acquisition and decision making for environmental management: http://www.ist-diadem.eu/

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 363–373, 2011.

The self-configuration provided by service discovery and matching can be improved by enhancing DPIF with a finer level of control based on one-to-many service negotiation ([3], [5], [7], [10]). Our negotiation model follows the conceptual framework for one-to-many service negotiation introduced in [1].

2 Background

Negotiation is a process that describes the interaction between one or more participants that must agree on a subject by exchanging proposals about this subject [4]. Negotiation about a service that one or more participants agree to provide to other participants is called *service negotiation*. We have developed a conceptual framework for service negotiation that addresses protocols, subjects and decision components. Our framework supports generic one-to-many negotiations and it defines two roles: manager and contractor [9,5]. The manager is the agent that requests a service and thus initiates the negotiation. The contractor is the agent that is able to provide the service requested by the manager. More details can be found in [1].

Currently we have configured our framework with three protocols that we have found useful in disaster and environment management problems. These protocols are:

1. *Direct Task Assignment*. Using this protocol there is actually no negotiation happening between the manager and the contractors. The manager simply picks up one or more contractors and assigns them the task[2]. Although simple, *Direct Task Assignment – DTA* negotiation protocol is actually very useful to model subordination relationships, that are common for task assignments to team members that operate in a disaster environment. For example in a disaster management information system, an *Incident Commander* can dispatch tasks to team members operating in the area affected by the disaster.
2. *Contract Net*. Proposed by [9], *Contract Net – CNET*[3] is probably one of the most influential negotiation protocols utilized in distributed collaborative problem solving, with many extensions currently available (see for example [10]). *CNET* is essentially a one step protocol: the manager announces a task to the contractors in the announcement stage, each contractor proposes or refuses to submit a bid in the bidding stage, and finally the manager decides to award the task to at least one of the contractors in the awarding phase. It can be easily noticed that the *DTA* protocol is in fact a simplified *CNET* without the announcement and the bidding stages.

[2] *DTA* has similarities with *FIPA Request* interaction protocol, see http://www.fipa.org/specs/fipa00026/, excepting that the contractor cannot refuse the task.

[3] *CNET* is standardized by Foundation for Intelligent Physical Agents, see http://www.fipa.org/specs/fipa00029/

3. *Iterated Contract Net. CNET* can be extended to multiple steps thus obtaining *Iterated Contract Net – ICNET*[4]. This protocol is very useful in situations when a single step is not sufficient for the manager to select a contractor for awarding the task. This may happen for example if the requirements set for the task announced by the manager are too restrictive and thus neither of the contractors is able to meet them in a satisfactory way. Another situation is when contractors are operating in a constrained environment that does not allows them to bid. For example, considering a disaster management scenario, one of the stakeholders (for example a chemical expert) might be caught in an important meeting that forbids him or her to answer to incoming phone calls. However, repeated calls, probably originating from repeated task announcements in a negotiation caused by the occurrence of a very important event, can enable him to pickup his phone and answer, i.e. to bid.

Negotiation subject comprises the service description and a subset of the service parameters that are important decision factors during negotiation (i.e. their current values are taken into consideration when selecting the appropriate service providers). During negotiation, these parameters are considered *negotiation issues*.

Negotiation issues are described by name, data type, and monotonicity. The name uniquely identifies the issue in a negotiation subject. The data type describes the type of the value the issue is allowed to take (e.g. number, string, geographical location, date/time). The monotonicity specifies whether the manager prefers higher values to lower values of this issue: (i) INCREASING if the agent prefers high utility values of the issue and (ii) DECREASING if the agent prefers low utility values of this issue.

Service negotiations in disaster management are *cooperative*. Cooperativity stems from the fact that the overall goal of participants is the optimization of the response for situation assessment. Negotiations for a certain service provision are carried out only with agents that can provide the required service (i.e. that possess the domain knowledge or physical capabilities that are needed to provide the service). Provider agents will usually accept to offer their services if they are currently able to do so (i.e. if they posses all the necessary resources). During a negotiation: (i) the manager is the leading decision factor seeking to optimize the assignment of the negotiation task to the contractor(s); (ii) the contractor(s) make their best proposals for the manager, taking into account their current duties and availability.

Service parameters can be classified into 4 classes: (i) DYNAMIC that specifies that the issue value is not fixed by the manager, i.e. the contractor can propose different values for the issue; (ii) FIXED that specifies that the issue value is fixed by the manager, i.e. if the contractor proposes a different value for the issue then the corresponding local utility of the issue is zero; (iii) CONDITION that specifies that the issue value is fixed by the manager, but if the contractor

[4] *ICNET* is standardized by Foundation for Intelligent Physical Agents, see http://www.fipa.org/specs/fipa00030/

proposes a different value for the issue then the total utility of the proposal is zero; normally a contractor that cannot meet the issue value requested by the manager should decide to not bid because the utility of her bid will be zero; (iv) TRIVIAL that means that the issue is not taken into account in the computation of the bid utility, although it can be set in the request of the manager and consequently, it can be taken into account in the negotiation by the contractor and help her to make a more informed decision if and what to bid.

Negotiation participants in either manager or contractor role use utility functions to measure their preferences over proposals. In our framework the manager uses a weighted additive utility function to evaluate proposals and to select the service provider (we omit the discussion of contractor utility functions, as it is not relevant for this paper; for details on contractors' utilities please consult [1]). Each negotiation issue i has a weight $w_i \in [0,1]$ and a partial utility function f_i. Note that weights are normalized i.e. $\sum_i w_i = 1$. Intuitively, for the manager the weight of an issue represents the relative importance of that issue in the set of all issues associated to a negotiation subject.

The partial utility of an issue i maps the issue domain D_i to a value in the interval $[0,1]$, i.e. $f_i : D_i \rightarrow [0,1]$. The function f_i depends on the domain of the issue. For example, a possibility to define the partial utility function of a real valued issue with $D_i = [x_{min}, x_{max}]$ is $f_i(x) = \frac{|x-x^*|}{|x_{max}-x_{min}|}$. Here, x^* is the reference value assigned by the manager to the issue i (that represents the preferred value from the manager point of view) and $|x-y|$ is the distance between x and y (note that the distance depends on the data type of the negotiation issue). Let I be the set of negotiation issues partitioned into sets $I^{\uparrow}$ and $I^{\downarrow}$ of issues with INCREASING and DECREASING monotonicity. The utility function of a proposal $x = (x_i)_{i \in I}$ is $u_m(x) = \sum_{i \in \mathcal{I}^{\uparrow}} w_i * f_i(x_i) + \sum_{i \in \mathcal{I}^{\downarrow}} w_i * (1 - f_i(x_i))$.

3 Sample Scenario

We illustrate our approach by using an example derived from a real world use case investigated in the FP7 DIADEM project. For the sake of clarity but without the loss of generality we assume a significantly simplified scenario. In a chemical incident at a refinery a leaking chemical starts burning which results in harmful fumes. The impact of the resulting fumes is assessed through a service composition involving collaboration of human experts. During this incident health complaints are reported. Consequently, the *Incident Commander* dispatches a chemical expert that holds expertise in estimating the gas concentration in the affected area. This expert is denoted as *Chemical Adviser* and, among other things, she guides fire fighter *Measurement Teams* which can measure gas concentrations at specific locations in order to provide feedback for a more accurate estimation of the critical area. This interaction between *Chemical Adviser* and *Measurement Teams* involves negotiation to determine potentially good providers of appropriate measurements (for more details see [2]).

We can easily observe that in this scenario *Chemical Adviser* agent plays the negotiation role of manager looking for a provider for the service *Concentration*

Table 1. Negotiable issues and manager request (left) and contractors' bids (right)

Issue	Location	Quality	Deadline
Reference value	*loc*	100	11:47 AM
Weight	1	2	3
Data type	REGION	INTEGER	TIME
Boundary	n/a	100	100
Negotiable	FIXED	DYNAMIC	DYNAMIC

Issue	Location	Quality	Deadline
Bid Value (1)	*loc*	70	11:58 AM
Bid Value (2)	*loc*	100	12:12 PM

Measurements. The selection of the service provider takes into account: the location where the measurement must be performed, the quality of the measurement, and the duration for performing the measurement. Additionally we assume that the measurement quality is given as a percentage and that the maximum time frame for performing the measurement is 100 minutes. The description of the negotiation issue, together with the manager proposal are given in Table 1. Weights of negotiation issues were normalized as follows: $w_{Location} = 1/6$, $w_{Quality} = 2/6$, and $w_{Deadline} = 3/6$.

Let us assume that there are two *Measurement Teams* in the system and each of them decides to bid with an offer for providing the service *Concentration Measurements*. Their bids are shown in Table 1.

The utility of the bid of the first *Measurement Team* is computed as follows:

$$\begin{aligned} u_{Location_1} &= 1/6 \times (1 - 0/1) &&= 0.166 \\ u_{Quality_1} &= 2/6 \times (1 - 30/100) &&= 0.233 \\ u_{Deadline_1} &= 3/6 \times (1 - 11/100) &&= 0.445 \\ u_{MT_1} &= u_{Location_1} + u_{Quality_1} + u_{Deadline_1} &&= 0.844 \end{aligned}$$

The utility of the bid of the second *Measurement Team* is computed as follows:

$$\begin{aligned} u_{Location_2} &= 1/6 \times (1 - 0/1) &&= 0.166 \\ u_{Quality_2} &= 2/6 \times (1 - 0/100) &&= 0.333 \\ u_{Deadline_2} &= 3/6 \times (1 - 25/100) &&= 0.375 \\ u_{MT_2} &= u_{Location_2} + u_{Quality_2} + u_{Deadline_2} &&= 0.874 \end{aligned}$$

Chemical Adviser agent uses these equations to compute the utilities of each bid received from *Measurement Team* agents. Then *Chemical Adviser* applies a strategy that allows it to either immediately select the winning bid or to decide if to continue the negotiation using a new iteration. Let us assume that *Chemical Adviser* applies a strategy that considers acceptable only those bids that pass a given threshold. If none is above the threshold then *Chemical Adviser* can perform a second iteration either by relaxing the conditions of the call for proposals (for example by decreasing the required quality of the measurements or by extending the deadline for performing the measurements) or by decrementing the threshold, thus giving a chance to the *Measurement Team* agents to update their bids. If at least one bid is considered acceptable then *Chemical Adviser* can decide to accept one or more *Measurement Team* agents to contract the *Concentration Measurements* service. Assuming a threshold of 0.85, according to this algorithm *Chemical Adviser* will select the second *Measurement Team* after the first iteration.

4 Experimental Results

4.1 Modeling Assumptions

When faced with a service negotiation problem, an agent playing the manager role will have to decide what negotiation protocol to use. In our case, the manager may decide to use one of the *DTA*, *CNET* or *ICNET* negotiation protocols that were discussed in Section 2. This choice will affect important factors like quality of the negotiation outcome as well as computational complexity incurred during the negotiation interaction [8] that overall impact the efficacy of the collaboration process for resolving the incident.

In what follows we propose a simple experimental setting aimed at analyzing the impact of the strategy employed by the manager agent to choose a service negotiation protocol on the quality of the negotiation outcome as well as on the computational complexity of negotiation interactions. The experiment is focused on evaluating a single negotiation rather than a complex collaboration workflow comprising more negotiations (we are aware of this limitation and we plan to address it in the near future).

Let us assume that a manager agent M is negotiating for contracting a service from a contractor agent that is member of a set of n contractors $C_1, \ldots, C_n$. Simplifying things we assume that each contractor C_i can offer a utility value u_i to the manager such that $u_i \in [u_i^{min}, u_i^{max}] \subseteq [0, 1]$. The interval $[u_i^{min}, u_i^{max}]$ is part of the profile of the contractor. For example, if the utility intervals of contractors C_i and C_j have an empty intersection and $u_i^{min} < u_j^{min}$ it means that C_j will always appear as more profitable than C_i for the manager. However, if the intersection of the intervals is nonempty it means that sometimes C_i can also be more profitable than C_j for the manager.

Moreover, each negotiation will always involve requirements that are set by the manager and must be met by the contractor in order to be allowed to bid for a contract. These requirements depend on the negotiation issues and on the constraints on the negotiation issues that are set by the manager in the request for service. For example, considering the *Concentration Measurements* service, *Chemical Adviser* may require *Measurement Teams* to bid if and only if they are equipped with certain specialized measurement devices like for example *drager tubes*[5]. We assume that for each contractor C_i there is a probability of $c_i \in [0, 1]$ that she will be able to satisfy the requirements set by the manager's request. The probability p_i is also part of contractor's C_i profile.

Contractors are usually critical resources that do not always have the possibility to bid for a service contract. Let us take for example the *Chemical Adviser* that can be caught in an important meeting while health complaints start being reported. In such a situation the *Incident Commander* must assign the task *Map of Critical Zones* to an expert in the chemistry of gases, i.e. the *Chemical Adviser*. Then the *Incident Commander* must adopt an iterative negotiation protocol to

[5] They are tubes that contain chemicals that react to a specific compound to form a stain or color when passed through the gas.

allow the *Chemical Adviser* to bid. Let us assume that in our experimental setting each contractor C_i is busy with a probability $b_i \in [0.1]$. Nevertheless, we assume that even when she is busy, if C_i is able to meet the requirements set by the manager then she will find time to bid in the second iteration.

Let us also assume that the manager M will choose to utilize one of the DTA, $CNET$ or $ICNET$ (two iterations version) negotiation protocols with the probabilities $p, q, r \in [0, 1]$ such that $p+q+r = 1$. The triple (p, q, r) is part of the manager's profile. Depending on the utilized protocol the negotiation will incur a certain communication cost estimated as the number of messages exchanged between the manager and the contractor during the negotiation. Moreover the quality of the negotiation outcome will be estimated as the utility perceived by the manager for the contracted service.

If the manager is utilizing the DTA negotiation protocol then she will randomly assign the task to one of the contractors. However, a contractor that does not meet the requirements will not be able to provide the service, so she will have to report failure. In this case the manager will randomly select another contractor and so on, until a suitable contractor is found (we assume this is always the case). Note however that this trial-and-error process performed by the manager affects the outcome of the negotiation by decrementing her perceived utility. More precisely, if the successful contractor C_i that could perform the task was selected in the k-th trial then the utility perceived by the manager will be $u_i \times (1-(k-1)/n)$ rather than u_i. Moreover, the communication cost associated to this negotiation interaction consists of $2 * k$ message exchanges.

If the manager is utilizing the $CNET$ negotiation protocol then she will select the contractor C_i that provides her the highest utility u_i from those contractors that met the requirements of the call for proposals and were not busy (i.e. they were able to bid). The communication cost consumed for a busy contractor consists of 2 message exchanges, while for a not busy contractor (it doesn't matter if she could met or not the requirements, according to the $CNET$ negotiation protocol [9] she either proposed or refused to bid) the communication cost consists of 3 message exchanges.

If the manager is utilizing the $ICNET$ protocol we assume that she will always perform two negotiation iterations. We also assume that a busy contractor that meets the requirements of the call for proposals will always find time to bid during the second negotiation iteration. The manager decides the contractor to whom to award the task after the second iteration. The communication cost for a busy contractor consists in 4 message exchanges, while for a non-busy contractor it consists of 5 message exchanges.

4.2 Experimental Results and Discussions

We created a simulation experiment assuming the model introduced in Section 4.1. In the simulation we considered one manager and n contractors. During a simulation, the given manager and contractors' profiles are set and a large number of negotiation instances are run. The manager is characterized by her profile (p, q, r) that defines her strategy for dynamic selection of the negotiation

protocol for each negotiation instance. Each contractor C_i is characterized by her profile defined as a triple $([u_i^{min}, u_i^{max}], c_i, b_i)$. The contractors' profiles are given as input to the simulation algorithm. The values of the utilities u_i are randomly selected for each negotiation instance assuming uniform distributions. Moreover, the probabilities c_i and b_i are utilized to determine the status of a contractor (as satisfying or not satisfying the requirements of the manager proposal and busy or not busy) for each negotiation instance.

The goal of the simulation was to observe and analyze the quality of the negotiation outcome as well as the incurred communication cost for different profiles of the manager agent. Therefore we ran the simulation for various manager profiles by sampling the probability space (p, q, r) accordingly. Assuming a sampling rate of $1/m$ (where m is a given natural number) we developed a simulation program that evaluates the average utility perceived by the manager, as well as the average communication cost incurred during negotiations for the following large set of manager profiles $\{(i/m, j/m, 1-(i+j)/m) | 0 \leq i, j \leq m,\ 0 \leq i+j \leq m\}$.

In our initial experiment we considered a simple negotiation case with 1 manager and 3 contractors. We ran 2000 negotiations and we recorded the negotiation outcome as well as the number of exchanged messages. We took $m = 20$ in the sampling rate of the probabilities that define the manager profile. The contractor profiles were set as follows: $C_1 = ([0.2, 0.3], 0.8, 0.2)$, $C_2 = ([0.5, 0.6], 0.5, 0.5)$, and $C_3 = ([0.8, 0.9], 0.2, 0.8)$. These values show that the manager will always prefer C_2 and C_3 to C_1.

If the manager is using the DTA protocol, C_1 will perform the task whenever she is able to meet the manager's requirements and (i) she is either selected the first or (ii) she is selected after C_2 and C_3 but both C_2 and C_3 could not meet the manager's requirements and reported failure (note that as C_1 meets the requirements requested by the manager with a higher probability than C_2 and C_3, she will perform quite often the task according to the DTA protocol). The conditions for C_2 or C_3 to perform the task are analogous with the conditions for C_1. Note that the status of being "busy" is not taken into account by the manager when she is playing the DTA negotiation protocol, i.e. if she selects a certain contractor, that task will be assigned to her in any case. Nevertheless, if the assigned contractor cannot finalize the task successfully she will report failure and consequently the manager will assign the task to another contractor.

If the manager is using the $CNET$ protocol the contractors that are set to "busy" are not taken into account by the manager, as they cannot bid. although they receive the call for proposals from the manager. For example, even if C_3 can offer a high utility to the manager, she has a high probability of being busy, so she will not be able to bid in many $CNET$ negotiations. So in this case C_2 can C_1 can win the negotiation more often. Moreover, C_2 is also busy quite often, giving a chance to award the task to C_1.

Finally, when the manager is using the $ICNET$ protocol, even if a contractor is "busy" and cannot bid in the first negotiation iteration she will still be able to bid in the second iteration. This interaction pattern allows for example to contractor C_3 to bid and consequently to increase the utility perceived by the manager.

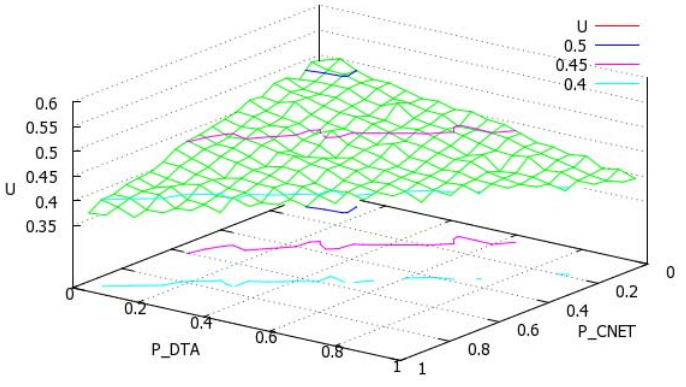


Fig. 1. Manager utility vs. probabilities for selecting DTA and $CNET$ protocols

The experimental results are shown graphically in Figures 1 and 2. These figures present the plots of the average utility perceived by the manager as well as of the average number of exchanged messages per negotiation as functions of the probabilities p and q (shown as P_DTA and P_CNET on the figures) of the manager to choose between the DTA and $CNET$ negotiation protocols. Note that the dependencies on the probability r for choosing the $ICNET$ protocol trivially follow as $p + q + r = 1$.

The first observation is that the highest utility (slightly above 0.5) is perceived by the manager when she always uses $ICNET$ negotiation protocol. However, this strategy also brings her the highest overhead in terms of communication complexity, i.e. slightly above 14 messages on average per negotiation. This clearly shows the tradeoff that exists between the utility of the solution and the communication complexity that occurs in negotiation interactions when the negotiation protocol can be dynamically selected by the initiator of the negotiation.

Secondly, we can observe that the difference in terms of manager's perceived utility between the DTA and $CNET$ protocols is small (at least in this experiment). For example, if $ICNET$ is not utilized, i.e. $r = 0$ or equivalently $p + q = 1$, we can easily observe that the average utility perceived by the manager is slightly variable around 0.4. This can be explained by the fact that what is gained by the bidding stage that is present in $CNET$ (and absent in DTA) is actually lost by the fact that a "busy" contractor cannot bid, while DTA can use him or her for awarding the task. In such a situation a good strategy of the manager will depend on his knowledge of the probabilities of availability to bid of the contractors. i.e. the less are they busy the higher will be the manager's perceived utility. Note however that communication complexity is clearly higher for $CNET$ (about 10 messages per negotiation if $p = 0$ and $q = 1$) than for DTA (about 4 messages per negotiation if $p = 1$ and $q = 0$).

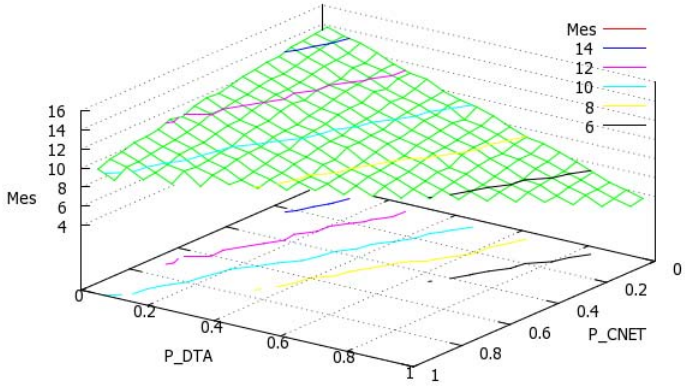


Fig. 2. Number of exchanged messages vs. probabilities for selecting DTA and $CNET$ protocols

Finally, note that the manager utility u_m for a manager profile (p, q, r) can be decomposed as "expected utility" based on the utilities that would have been obtained if the manager had been used either DTA, $CNET$ or $ICNET$ protocols only, i.e. $u_m(p.q.r) = p \times u_m^{DTA} + q \times u_m^{CNET} + r \times u_m^{ICNET} = p \times u_m(1,0,0) + q \times u_m(0,1,0) + r \times u_m(0,0,1)$. This decomposition of u_m explains the planar shape of the surface representing function u_m in Figure 1. This observation extends also to the planar shape of the surface representing Mes function in Figure 2.

5 Conclusions

In this paper we presented a framework that allows definition of one-to-many service negotiation protocols in agent-based collaborative processes. This framework supports flexible configuration of multi-issue negotiation subjects, properties of negotiation issues, and utility functions of participant agents. An example describing a sample negotiation scenario was discussed in detail, emphasizing how service negotiation can improve the selection of potentially good service providers. We also presented a simple experimental setting and initial experimental results aimed at analyzing the impact of the strategy employed by the manager agent for selecting a service negotiation protocol on the quality of the negotiation outcome as well as on the computational complexity of negotiation interactions. As future work we plan to expand the experiments in at least two directions: (i) to analyze the impact of the negotiation protocol on the quality of the negotiation outcome as well as on the computational complexity of negotiation interactions depending on several profiles of contractor agents; the results can help the manager to tune his strategy for the selection of the negotiation protocol; (ii) to consider more complex workflows involving at least two interdependent negotiations such that the contracted service might also involve contracting of other required services.

Acknowledgement. The work reported in this paper was carried out as part of the Diadem project: http://www.ist-diadem.eu. The Diadem project is funded by the E.U. under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D, ref. no: 224318.

References

1. Bădică, C., Scafeş, M.: Conceptual Framework for Design of Service Negotiation in Disaster Management Applications. In: Bai, Q., Fukuta, N. (eds.) Advances in Practical Multi-Agent Systems. SCI, vol. 325, pp. 359–375. Springer, Heidelberg (2010)
2. Bădică, C., Ilie, S., Kamermans, M., Pavlin, G., Scafeş, M.: Using Negotiation for Dynamic Composition of Services in Multi-Organizational Environmental Management. In: International Symposium on Environmental Software Systems ISESS 2011 (submitted, 2011)
3. Hemaissia-Jeannin, M., El Fallah Seghrouchni, A., Labreuche, C.: A New Multilateral Multi-issue Negotiation Protocol and its Application to a Crisis Management Problem. Multiagent and Grid Systems 4(1), 103–123 (2008)
4. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10(2), 199–215 (2001)
5. Paurobally, S., Tamma, V., Wooldridge, M.: A Framework for Web service negotiation. ACM Transactions on Autonomous and Adaptive Systems 2(4) (2007)
6. Pavlin, G., Kamermans, M., Scafeş, M.: Dynamic Process Integration Framework: Toward Efficient Information Processing in Complex Distributed Systems. Informatica 34, 477–490 (2010)
7. Sandholm, T.: An implementation of the contract net protocol based on marginal cost calculations. In: Proceedings of the 12th International Workshop on Distributed Artificial Intelligence, pp. 295–308 (1993)
8. Scafeş, M., Bădică, C.: Framework for Performance Evaluation of Service Negotiations in Agent Systems. In: Proc. of the 8th Int.Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems MSVVEIS 2010, pp. 19–29. INSTICC Press (2010)
9. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29(12), 1104–1113 (1980)
10. Yang, J., Li, W.-L., Hong, C.-Y.: An Improvement to CNCP in Large-Scale Muti-Agent System. Proccedings of the 3rd International Conference on Innovative Computing Information and Control ICICIC 2008, 147 (2008)

Guaranteeing Quality of Service in Globally Distributed Web System with Brokers

Krzysztof Zatwarnicki

Department of Electrical, Control and Computer Engineering,
Opole University of Technology, Opole, Poland
k.zatwarnicki@gmail.com

Abstract. A significant development of Web technologies requires the application of more and more complex methods for maintaining high quality of Web services. This paper presents the application of fuzzy logic and neural networks to HTTP request dispatching performed within a geographically distributed Web system to guarantee quality of service. The proposition of a broker-based system architecture with a global request dispatching method called GGARDiB (Guaranteed Globally Adaptive Request Distribution with Broker) is presented. The method guarantees quality of the Web service on given level, in such way that the page response times do not exceed established time. We demonstrate through the simulations that our algorithm is more effective than popular and reference global dispatching policies.

Keywords: Neuro-fuzzy control, HTTP request distribution, Guaranteeing quality of Web service, Dispatching algorithms, Cluster-based architectures.

1 Introduction

High quality of Web services has a major impact on the profitability of enterprises conducted in the Internet. The quality of Web service can be categorized as best-effort and guaranteed. While best-effort services are unpredictable and do not guarantee that the service will be completed at all, the guaranteed services are predictable and reliable [6]. In this article the method of HTTP request distribution in globally distributed Web system will be presented. Proposed method enables guaranteeing quality of the Web service on given level, in such way that the page response times for all pages and users will not exceed demanded time in case the load of the system is lower than its capacity. There have been many works on how to guarantee the Web service quality. Most of them concern maintaining the quality of the service for individual HTTP requests [2], [5], [8]. Very few papers were dedicated to the problem of designing Web service, which would guarantee servicing the whole WWW pages within limited time [9]. Guarantying quality of service is almost always connected with rejecting requests of users not belonging to privileged group. In the method proposed we use artificial intelligence techniques namely neuro-fuzzy networks to estimate requests service times. In our previous works we have already used those techniques to minimize HTTP request response times in locally distributed Web

P. Jędrzejowicz et al.(Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 374–384, 2011.

cluster system (LFNRD method) [3], globally distributed Web cluster system with broker (GARDiB method) [4] and globally distributed Web cluster system without broker (GARD method) [11]. In our later works we deal with guaranteeing quality of service in Web systems with one Web server (WEDF method) [12] and locally distributed cluster of servers (MLF method) [10]. In this article we show new GGARDiB method guaranteeing the quality of service in globally distributed Web cluster system with broker. According to our knowledge, the GGARDiB is the first method of this sort. The paper is divided into four sections. Section 2 presents the problem formulation and a design of a GGARDiB method together with a description of construction and operation of a neuro-fuzzy model used. Section 3 describes a simulation testbed used in the experiments and the results. The final Section 4 contains concluding remarks.

2 GGARDiB Method Description

The GGARDiB system is designed to distribute HTTP requests among globally distributed Web server clusters. The system consists of following elements: clients sending HTTP requests, broker servers (or simply brokers) placed nearby large concentrations of clients, local services (LS) and DNS servers. The overall view of globally distributed Web cluster system with broker is presented on Fig. 1.a. LSs are locally clustered Web systems containing fully replicated Web servers, back-end servers (application and database servers) and Web switch controlling operations of the cluster. According to adopted assumptions, the GGARDiB system should operate in this way to not let the page response time $\dot{t}_{p_i}$, for each of the pages and clients, be longer than established time $t_{\max}$ under the condition that the load of the system (measured in the number of requests serviced per time unit) will not exceed the capacity of the system. The response time $\dot{t}_{p_i}$ of the Web page p_i is measured from the moment of receiving by the broker the first HTTP request, concerning the given web page sent by the client, up to receiving by the broker the HTTP response concerning the last object belonging to the page, sent by the same client. The response time is reduced by the value of the time, when no other request of the same client, concerning the Web page, was processed within the aforementioned time interval. The decisions in the GGARDiB system are taken in the brokers and web switches. Client before sending the first HTTP request to the GGARDiB system receives from specially constructed DNS system the IP address of the broker located in the nearest proximity to the client. After that the client sends the HTTP request x_i (i is the index of the request) to the broker. The broker is taking into account the information of the load $O_i''^1,...,O_i''^{s''},...,O_i''^{S''}$ of the LSs and the previous request response times $\tilde{t}_1'',...,\tilde{t}_i''$ chooses for the request x_i, the LS w_i'' out of S'' LSs, in this way that $w_i'' = \min_s \left\{ \hat{t}_i''^{s''} : s'' = 1,...,S'' \right\}$, where $\hat{t}_i''^{s''}$ is the estimated response time for i th request and s'' th LS. Subsequently the broker modifies the request x_i to x_i' inserting

to the request information about deadline db_i pointing the moment, when the process of sending the response to the broker should start. Next the modified request x_i' is send to the chosen LS's Web switch. After receiving the request the Web switch determines the deadline ds_i'' indicating the moment the request service on the Web server should start. Then the request x_i' is placed in the queue Q_i in the Web switch. After occurrence of the appropriate conditions the request leaves the queue and the Web switch determines, on the base of the load $O_i^1,...,O_i^s,...,O_i^S$ of the servers in the cluster, as well as the previous request service times $\tilde{t}_1,...,\tilde{t}_i$, Web server w_i out of S servers in the cluster, which will service the request. Next, after completion of request service on the Web server the response is transmitted to the Web switch, then to the broker and further to the client. Request's service time $\tilde{t}_i$ is a time measured since the moment the Web switch starts transmitting the request to the server to the moment the response arrives to Web switch. The decision making process is presented on the Fig 1.b for the broker and on the Fig 1.b for the Web switch.

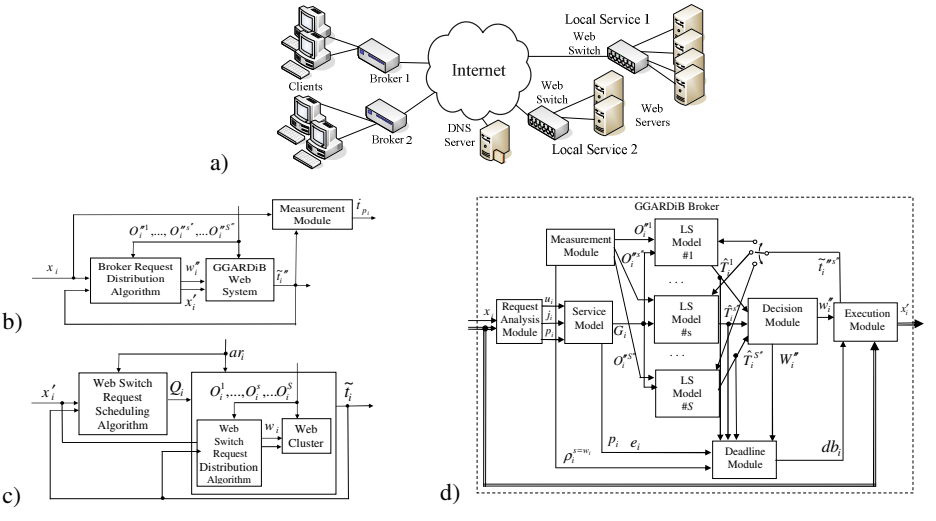


Fig. 1. a) GGARDiB Web system, b) Decision process in the broker, c) Decision process in the Web switch, d) Schema of the broker

2.1 Construction of the GGARDiB Broker Server

The main elements of the broker are: request analysis module, measurement module, service model module, models of LS modules, deadline module, decision module and execution module. The schema of broker is presented on the Fig.1.c.

The request analysis module after receiving the request $x_i = \langle u_i, ck_i \rangle$ analyses it and extracts the address u_i of requested object, identifier j_i of the client and the

identifier p_i of the page to which the requested object belongs. The page and client identifiers are contained in the cookie $ck_i = \{\langle j_i, p_i \rangle\}$ in the request x_i. Service model module stores information about objects available in the Web service, their sizes, kinds and the membership to Web pages. The input information passed to the module are u_i, j_i and p_i. The output information determined in the module is $G_i = [u_{i1},...,u_{il},...,u_{iL}]$, where $u_{i1},...,u_{il},...,u_{iL}$ are addresses of objects belonging to the same page p_i but not yet requested by client j_i, $u_{i1} = u_i$. Additionally the module calculates the time tp_{p_i} of page response time which elapses from the moment of the first request arrival concerning the page p_i to the moment of arrival x_i request. The broker contains S'' modules of LS models, which number is equal to the number of LSs in the service. Each model of LS corresponds to one LS in the system. On the input of each of the LS models the vector G_i of addresses of objects and the load $O_i''^{s''}$ ($s'' = 1,...,S''$) of each of the LS are passed. The output of the module is a vector $\hat{T}_i^{s''} = [\hat{t}_{i1}''^{s''},...,\hat{t}_{il}''^{s''},...,\hat{t}_{iL}''^{s''}]$ of estimated service times for objects pointed in the G_i vector, where s'' is the number of the LS the model correspond to.

The model of LS determines consecutive values $\hat{t}_{il}''^{s''}$, $l = 1,...,L$, one by one for each of the object. The LS model module consists of neuro-fuzzy model which will be described in a later part of this section. The measurement module acquires information about the load $O_i''^{s''}$, $s'' = 1,...,S''$, where $O_i''^{s} = [arl_i^{s''}, btt_i^{s''}]^T$ and $arl_i^{s''}$ is the number of request concurrently serviced in the s''th LS, $btt_i^{s''}$ is a transfer time measured during transferring a test object from the s''th LS to the broker. Both measures are obtained by the load measurement module periodically at fixed time interval. Additionally the module delivers information about effective throughput $\rho_i^{s''}$ on the line between the LS and the broker. Information of estimated by the LS model modules service times $\hat{T}_i^1,...,\hat{T}_i^{s''},...,\hat{T}_i^{S''}$ are passed to decision module. The decision module determine a vector of decisions $W_i = [w_{i1}'',...,w_{il}'',...,w_{iL}'']$, where w_{i1}'' (also denoted as w_i'') is the number of the LS chosen to service the request x_i, and $w_{i2}'',...,w_{il}'',...,w_{iL}''$ are numbers of LSs which may service the other request, $w_{il}'' \in \{1,...,S''\}$, $l = 1,...,L$. Each of the decision w_{il}'' is taken according to formula $w_{il}'' = \min_s \{\hat{t}_{il}''^{s''} : s'' \in \{1,...,S''\}\}$.

The deadline module determines deadline db_i, which is the time the LS should finish servicing the request x_i and starts sending the response to the broker. The deadline db_i is calculated according to the formula $db_i = \Delta db_i + c_i - h_i^{s''}\big|_{s''=w_i''}$, where c_i is the moment of x_i request arrival, Δdb_i is the time which can be spend

on servicing the request x_i and $h_i^{s''}$ is a transmission time of i th request from chosen w_i'' th LS to the broker. The value of $h_i^{s''}$ can be calculated in the following way $h_i^{s''} = v_i / \rho_i^{s''}\big|_{s''=w_i''}$, where v_i is the size of requested object. The Δdb_i is determined on the base of formula $\Delta db_i = \hat{t}_{i1}^{''s''}(t_{\max} - p_i) \Big/ \lambda \sum_{l=2}^{L} \hat{t}_{il}^{''s''}\Big|_{s''=w_i''}$, where λ is concurrency factor and it's value should be determined in experimental way. The execution module modifies the original request x_i to x_i' and sends the modified request to chosen w_i th LS, more precisely to the LS's Web switch. The request x_i' is formed from the x_i request supplemented with the deadline db_i in the cookie information, $x_i' = \langle u_i, ck_i' \rangle$ and $ck_i' = \{\langle j_i, p_i, db_i \rangle\}$. The execution module measures also the request response time $\tilde{t}_i''$ and after completing the measurements passes the time to the model of the LS which corresponds to w_i'' th LS, which updates the knowledge about the LS in adaptation process.

2.2 Construction of the GGARDiB Web Switch

The second, after the broker, most important element of the Web system is the GGARDiB Web switch controlling operations of the LS. The Web switch receives the HTTP request send by the broker, queues the request and redirects it to the Web server chosen by the distribution algorithm. Web server after receiving the request, services it and sends the response to the Web switch. The schema of the GGARDiB Web switch is presented on the Fig. 2.a.

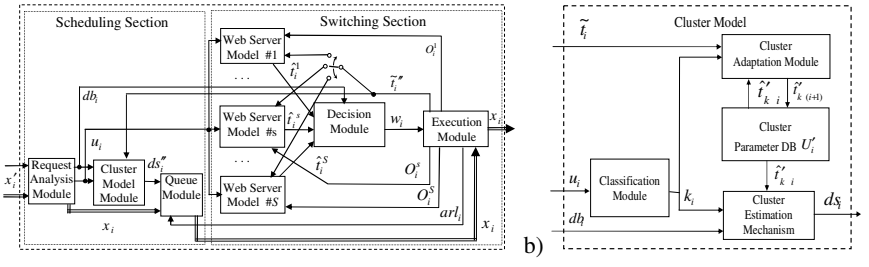


Fig. 2. a) GGARDiB Web switch, b) Cluster model module

The Web switch consists of two sections: scheduling and switching. The scheduling section is composed of request analysis module, cluster model module and queue module. The Web switch after receiving request $x_i' = \langle u_i, ck_i' \rangle$ passes it to the analysis module, which extracts the address u_i of requested object and from the cookie field $ck_i' = \{\langle j_i, p_i, db_i \rangle\}$ the deadline db_i. Additionally from the request x_i' the information about the deadline db_i is removed and in the result the original x_i

request is obtained and passed to the queue module. The cluster model module in the scheduling section determines the deadline ds_i'' pointing the time the i th request service on the Web server should start. The cluster model module is composed of classification module, cluster adaptation module, cluster parameter DB and cluster estimation mechanism. The classification module recognizes the class k_i, $k_i \in \{1,...,K\}$, of u_i th object on the base of the size of the object (in case of static object) or each dynamic object is classified to its own class. Objects belonging to the same class should be serviced in similar time.

The cluster parameter DB module denoted as $U_i' = [\hat{t}_{1i}', ..., \hat{t}_{ki}', ..., \hat{t}_{Ki}']$ stores service times of requests related to objects belonging to consecutive classes. The cluster estimation mechanism determines the deadline ds_i'' on the base of following formula $ds_i'' = db_i - \hat{t}_{ki}'$. $\hat{t}_{ki}'$ is k_i th class request estimated service time offered by the whole cluster. It's value is modified in the adaptation process by cluster adaptation module according to formula $\hat{t}_{k\ (i+1)}' = \hat{t}_{k\ i}' + \hat{\eta}(\tilde{t}_i - \hat{t}_{k\ i}')$, where $\hat{\eta}$ is adaptation ratio and $\hat{\eta} = 0.1$ (the value was determined in preliminary experiments). The queue module in the Web switch schedule requests according to deadlines ds_i'', $i = 1,2,....$ The queue module inserts the request into the queue only if the number arl_i of requests serviced simultaneously by the Web servers in the cluster is greater than adopted constant $arl_{\max}^s$, in other case the request is transferred directly to the execution module. The constant $arl_{\max}^s$ is the maximum number of requests admissible to be simultaneously serviced by the cluster, it is the lowest value for which the cluster maintains the highest performance measured in number of requests serviced per time unit. The request leaves the queue and is transmitted to the execution module when it is first in the queue and the number of requests serviced simultaneously decreases. The switching section of GGARDiB Web switch consists of: Web server model module, decision module and execution module. After transferring the request to the switching section models of server modules determine, on the base of servers load $O_i^1, ..., O_i^s, ..., O_i^S$, estimated requests response times $\hat{t}_i^1, ..., \hat{t}_i^s, ..., \hat{t}_i^S$. Each model of server corresponds to one of S Web servers working in the cluster. The construction of the model of server is similar to the construction of broker's model of LS introduced previously and will be presented in details in a later part of this section.

The decision module in the Web switch makes decision w_i, where $w_i \in \{1,...,S\}$, pointing the server which will service the request x_i. At first a time $\Delta d_i' = \tau_i - db_i$ is determined, which can be spent on servicing the request, where τ_i is the moment of arrival of the request to the execution module. The decision w_i is made according to formula

$$w_i = \begin{cases} \min\left\{s : s \in \{1,...,S\} \wedge \Delta d_i' \le \hat{t}_i^s\right\} \text{ if } ar_i = ar_{\max} \text{ and } \exists_{s\in\{1,...,S\}} \hat{t}_i^s \ge \Delta d_i' \\ \min\limits_s \left\{\hat{t}_i^s : s \in \{1,...,S\}\right\} \text{ in other case} \end{cases}.$$

The Web server of lowest number (index), offering estimated service time lower then $\Delta d_i'$ is chosen. In this way servers with the lowest numbers should be the most loaded while servers with the highest numbers should remain unloaded and thus allow fast service of requests which deadline db_i is coming up. The execution module transmits the request x_i to chosen w_i Web server. During servicing the request by the Web server the execution module measure the service time $\tilde{t}_i$. It also measure the number ara_i of simultaneously serviced requests and the load $O_i^1,...,O_i^s,...,O_i^S$ of individual Web servers in the cluster. The term $O_i^s = [a_i^s, b_i^s]^T$ characterizes a load of the s th Web server and the backend server, where a_i^s is a number of HTTP requests serviced simultaneously by the Web server and b_i^s is a number of HTTP requests serviced simultaneously and related to dynamic objects whose content is created by the application and/or database server when requests are received.

The model of the Web server will be described in details now. Its construction is very similar to construction of the LS model module in the broker, therefore one description of the module will be presented. For clarity of designations the index s of the server will be dropped in the pertaining formulae. This model consists of classification module, a server parameter DB, a server estimation mechanism and a server adaptation mechanism (Fig. 3.b). The classification module classifies requests in the same way as classification module in the cluster model module. The server estimation and adaptation mechanisms form together the neuro-fuzzy model. The neuro-fuzzy model consists of following elements: fuzzification, inference and defuzzification blocks (Fig. 3a).

Server parameter DB is denoted as U_i and stores information $U_i = [U_{1i},...,U_{ki},...,U_{Ki}]^T$ concerning parameters of membership functions for inputs and output, where $U_{ki} = [A_{ki}, B_{ki}, Y_{ki}]$ while $A_{ki} = [\alpha_{1ki},...,\alpha_{lki},...,\alpha_{(L-1)ki}]$, $B_{ki} = [\beta_{1ki},...,\beta_{mki},...,\beta_{(M-1)ki}]$, $T_{ki} = [t_{1ki},...,t_{jki},...,t_{Jki}]$. The fuzzy sets for input a_i were denoted as $Z_{a1},...,Z_{al},...,Z_{aL}$, and similarly fuzzy sets for input b_i were denoted as $Z_{b1},...,Z_{bm},...,Z_{bM}$, where L and M are the numbers of fuzzy sets. The fuzzy sets membership functions for both inputs are triangular and are denoted as $\mu_{Z_{al}}(a_i)$, $\mu_{Z_{bm}}(a_i)$, $l = 1,...,L, m = 1,...,M$, and meet the condition of the unit division. The parameters $\alpha_{1ki},...,\alpha_{lki},...,\alpha_{(L-1)ki}$ and $\beta_{1ki},...,\beta_{mki},...,\beta_{Mki}$ specify the shape of membership functions for individual inputs. The membership functions for input a_i are shown on Fig. 3c. The rule base module calculates the values of the membership to

consecutive rules according to formula $\mu_{R_j}(a_i,b_i)=\mu_{Z_{al}}(a_i)\cdot\mu_{Z_{bm}}(b_i)$ for $l=1,...,L$ and $m=1,...,M$, where $j=(m-1)\cdot L+l$. In the defuzzification block the crisp estimated service time value $\hat{t}_i$ is calculated with use of the Height Method, therefore $\hat{t}_i=\sum_{j=1}^{J}t_{jki}\mu_{R_j i}$, where $\mu_{R_j i}=\mu_{R_j}(a_i,b_i)$, $J=L\cdot M$ and t_{jki} are the values pointed by singleton membership functions for output. In the adaptation process following parameters of membership functions are tuned α_{lki}, β_{mki}, t_{jki}, where $l=1,...,L-1$, $m=1,...,M-1$, $j=1,...,J$. New values are calculated with use of the Back Propagation Method and Gradient descent rule in following way: $t_{jk(i+1)}=t_{jki}+\eta_y(\tilde{t}_i-\hat{t}_i)\cdot\mu_{R_j i}$, $\alpha_{\phi k(i+1)}=\alpha_{\phi ki}+\eta_a(\tilde{t}_i-\hat{t}_i)\sum_{m=1}^{M}\left(\mu_{Z_{bm}}(b_i)\sum_{l=1}^{L}\left(t_{((m-1)\cdot L+l)ki}\,\partial\mu_{Z_{al}}(a_i)/\partial\alpha_{\phi ki}\right)\right)$ and $\beta_{\gamma k(i+1)}=\beta_{\gamma k i}+\eta_b(\tilde{t}_i-\hat{t}_i)\sum_{l=1}^{L}\left(\mu_{Z_{al}}(a_i)\sum_{m=1}^{M}\left(t_{((l-1)\cdot M+m)ki}\,\partial\mu_{Z_{bm}}(b_i)/\partial\beta_{\gamma ki}\right)\right)$, where η_t, η_a, η_b are adaptation ratios, $\phi=1,...,L-1$, $\gamma=1,...,M-1$.

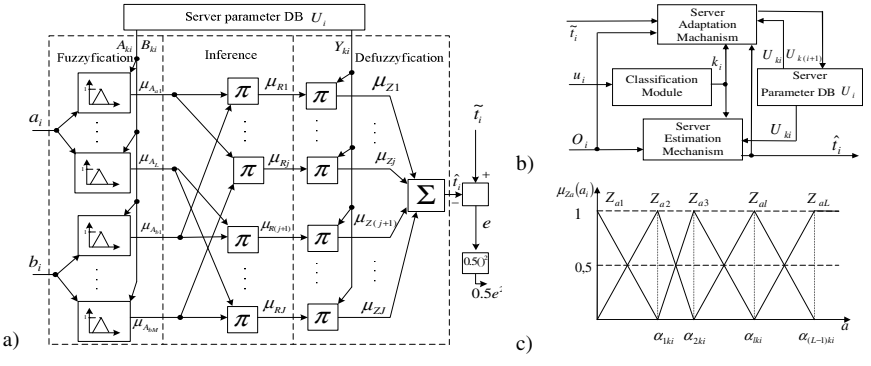


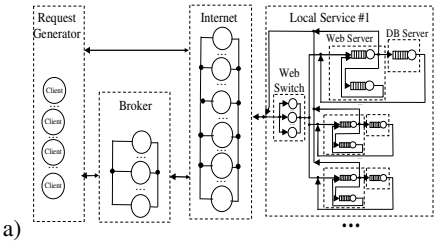
Fig.3. a) The neuro-fuzzy model of the server, b) The server model module, c) Input fuzzy sets membership functions

3 Simulation Model and Experiment Results

In order to evaluate the new GGARDiB method simulation experiments were carried out. The simulation program was build with use of CSIM19 [7] package and consisted of following modules: request generator, broker, Internet, web switch, Web server and DB server modules (Fig 3.1). In the broker module the GGARDiB method was implemented and also other distribution algorithms were employed like Round-Robin (RR) and Weighted Round-Robin in two variants. In the first variant (WRR_T) the weights were calculated on the base of the transfer between the broker and the LS, in the second variant (WRR_L) weights depended on the load. In the experiments, when the broker was working under control of the GGARDiB algorithm the Web switches were operating under control of the LFNRD algorithm. The Internet module was

transferring requests from the broker and the request generator to the Web switches and the responses back to requests sources. The transfer times of requests and responses were calculated according to results obtained after conducting transfer rates tests between the Opole University of Technology (Poland) and other major research centers in Poland, Nederland, USA, Australia and Brazil (centers are listed in order of obtained transfer rates).

The Web server module consisted of processor, hard drive and cache memory sections. The processor and the disk were modeled as queue systems with a single queue with one service. Request service times were determined in experimental tests for the server with an Intel Pentium 4, 2 GHz processor and a hard disk Seagate ST340810A 80GB. Operating system Linux Fedora Core 6 and WWW server software Apache 2.2.4 with modules PHP 5.1 were installed on the server. In the experiments, the request generator modeled work of the clients in this way that the generated request traffic complied with the actual traffic observed in the Internet, which is characterized by bursts and self-similarity (Fig 4.a.) [1]. The modeled system provided service of static and dynamic requests which were serviced by the Web and database server. The dynamic requests were divided into three classes: high intensive, medium intensive and low intensive. Service times of the dynamic requests on the database server were modeled according to hyperexponential distribution.



b)

Category	Distribution	Parameters
Requests per session	Inverse Gaussian	μ=3.86, λ=9.46
User think time	Pareto	α=1.4, k=1
Objects per request	Pareto	α=1.33, k=2
HTML object size	Lognormal	μ=7.630, σ=1.001
	Pareto	α=1, k=10240
Embedded object size	Lognormal	μ=8.215, σ=1.46

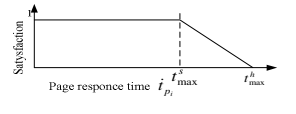


Fig. 4. a) A simulation model; b) Workload model parameters; c) Satisfaction function

The new approach in HTTP request disribution requires indication of adequate quality factor enabling evaluation of new proposed system. As the quality factor the mean value of satisfaction was chosen. This measure is often used in evaluation of soft real time systems. Satisfaction can be calculated on the base of a function presented on Fig. 4.c, where $t^s_{\max}$ is the "soft time", after which the satisfaction starts decreasing to 0 and $t^s_{\max} = t_{\max}$, $t^h_{\max}$ is the "hard time", after which the user will leave the page. It was adopted that $t^s_{\max} = 4s$ and $t^h_{\max} = 8s$.

The first experiment was carried out for broker placed in Opole University of Technology (Pl) and three LSs of following configurations: NL/3/25%; USA/3/25%; AU/3/25%; broker/25%, where on the subsequent positions following information are presented: country where the LS was placed, number of Web and DB servers in the cluster, percentage of requests redirected to the element of the system. In the second experiment following configuration was used: USA/3/16,666%; AU/3/16,666%;

BR/3/16,666%; PL/3/16,666%; broker/16,666%. In the Fig. 5 the mean satisfaction in function of the load (number of new clients created per second) is shown. The highest value of satisfaction was obtained by the GGARDiB method especially for Web system consisting more than tree LSs. Also good results but not greatest were achieved for GARDiB method minimizing individual HTTP request response times. The results of the experiments show that minimizing the request service times doesn't always lead to obtaining the highest quality of service. Results obtained for both WRR algorithms were rather poor.

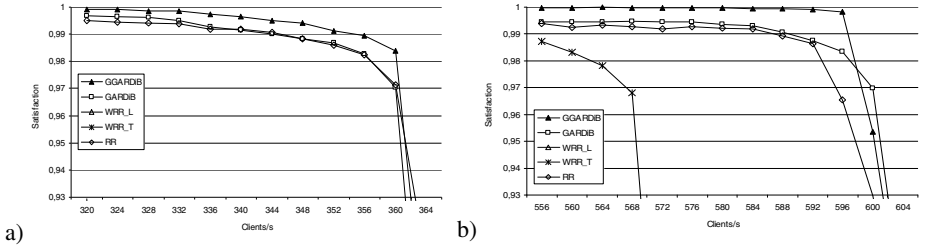


Fig. 5. Satisfaction in load function: a) experiment 1 b) experiment 2

4 Summary

In the paper the new GGARDiB method enabling guarantying quality of service on given level, in such way that the page response times do not exceed established time was described. The method can be applied in globally distributed Web systems and it uses in its construction the neuro-fuzzy decision-making mechanism. The simulation experiments shown that the method can guarantee higher quality than other reference and widely used request distribution methods. Our solution opens up a new group of solutions in the scope of guaranteeing quality of Web service evaluated from the user's point of view.

References

1. Barford, P., Crovella, M.E.: A Performance Evaluation of Hyper Text Transfer Protocols. In: Proc. ACM SIGMETRICS 1999, Atlanta, pp. 188–197 (1999)
2. Blanquer, J.M., Batchelli, A., Schauser, K., Wolski, R.: Quorum: Flexible Quality of Service for Internet Services. In: Proc. Symp. Networked Systems Design and Implementation (2005)
3. Borzemski, L., Zatwarnicki, K.: Fuzzy-neural web switch supporting differentiated service. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4252, pp. 195–203. Springer, Heidelberg (2006)
4. Borzemski, L., Zatwarnicki, K., Zatwarnicka, A.: Adaptive and Intelligent Request Distribution for Content Delivery Networks. Cybernetics and Systems 38(8), 837–857 (2007)
5. Kanodia, V., Knightly, E.: Multi-class latency-bounded web services. In: Proc. of the 8th International Workshop on Quality of Service, IWQOS (2000)

6. McCabe, D.: Network analysis, architecture, and design. Morgan Kaufmann, Boston (2007)
7. Mesquite Software Inc. CSIM User's Guide. Austin, TX (2010), http://www.mesquite.com
8. Schroeder, B., Harchol-Balter, M.: Web servers under overload: How scheduling can help. In: 18th International Teletraffic Congress, Berlin, Germany (2003)
9. Wie, J., Xue, C.Z.: QoS: Provisioning of client-perceived end-to-end QoS guarantees in Web servers. IEEE Trans. on Computers 55(12) (2006)
10. Zatwarnicki, K.: A cluster-based Web system providing guaranteed service. System Science 35(4), 68–80 (2009)
11. Zatwarnicki, K.: Neuro-Fuzzy Models in Global HTTP Request Distribution. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS, vol. 6421, pp. 1–10. Springer, Heidelberg (2010)
12. Zatwarnicki, K.: Providing Web service of established quality with the use of HTTP requests scheduling methods. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS, vol. 6070, pp. 142–151. Springer, Heidelberg (2010)

Customized Travel Information Recommendation Framework Using CBR and Collective Intelligence

Mye Sohn, Su ho Kang, and Young Min Kwon

Department of Industrial Engineering
Sungkyunkwan University
300 Cheoncheon-dong, Jangan-Gu, Suwon, Korea
{myesohn,rpc101,mini8716}@skku.edu

Abstract. This paper proposes an ***O***ntology based ***C***b***R*** system with collective ***I***ntelligence for customized i***N***nformation recommendation (OCRINA), which combines case-based reasoning (CBR) and collective intelligence. Unlike traditional CBR, which consists of 4R (retrieve-reuse-revise-retain), OCRINA is comprised of 5R, which includes the stage of reap additionally, which converts compiled collective intelligence into cases. The system utilizes ontology in order to extract a similar case from the case base, which is a collection of collective intelligence. It also utilizes the technique of Latent Semantic Analysis (LSA) to calculate the extent of the comprehensive similarity between the entirety of all the requests and the whole of all the index values, on the premise that the index values making up the cases will not be independent. Also, in order to demonstrate the superiority of the system, it performs a comparison with the method proposed by Leacock and Chodorow as the best among all the methods calculating the extent of the similarity among concepts.

Keywords: Collective Intelligence, Case-based Reasoning, Latent Semantic Analysis, Ontology.

1 Introduction

Collective Intelligence (CI) is applied extensively to multiply the capacity of an individual to execute his or her decision-making utilizing only limited cognitive ability [1, 2, 3, 4, 5, and 6]. With the rapid growth of the Internet, solving personal problems by borrowing group power is becoming increasingly common. For instance, traveler refers to blogs posted by previous travelers to the area or to information posted on social networking sites to map out a successful journey. At this time, a traveler who maps out a journey are being unconsciously assisted by power of group, called CI. Considering that much of the CI utilized is the experiences of other people who posted relevant information on their blogs or social network sites. The process in which a traveler generates customized information is very similar to the problem-solving procedure of case-based reasoning (CBR). A major difference is that while the traditional CBR system has only applied past experiences of an individual or a small group of experts to the case base [7], web-based CI customizes information

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 385–394, 2011.

using abundant resources produced in a distributed web environment. Furthermore, in the web-based CI, it is the traveler, i.e., a human, who has to customize the collected information, whereas the traditional CBR is loaded with an automated reasoning function. A traveler spends a lot of time and effort searching for information, sorting out the information, and then setting up a travel plan made up of the information.

This paper proposes an ontology based ***C***b***R*** system with collective ***I***ntelligence for customized i***N***nformation recommend***A***tion (OCRINA) in order to reduce the travelers' burden that need to information customization. This paper also conducts the following investigations so as to fulfill the OCRINA system.

- The collection and conversion of CI: Information posted on blogs or social networking sites is treated as cases since such information includes personal experiences. However, because much of CI is unstructured documents that cannot be handled by computers, a pertinent solution is needed. This paper employs a modified Sohn and Lee method [8] as a way to convert unstructured web documents into cases without the burden of treating natural language.
- Calculating the similarity in order to retrieve the most similar case: The methods to calculate the similarity among the traveler request as a new problem and cases in the case base are very diverse. Recently, methods that can arrive at not only the grammatical similarity but also semantic similarity by utilizing ontology are widely used. These methods select similar cases by comparing the weighted sums of those after calculating the term-by-term similarity. Yet, these methods fail to consider the latent association that can exist among indices that make up the traveler requests and cases. This paper proposes a similarity-calculating method that has adapted Latent Semantic Analysis (LSA) to CBR to consider the latent association that can exist among indices.

The makeup of this paper is as follows. Chapter 2 takes a look at related studies, and Chapter 3 explains in detail the overall architecture of the OCRINA system and the stages of 5R. Chapter 4 draws out the similarity by utilizing 15 user requests, 80 cases, and travel ontology in order to demonstrate the superiority of the OCRINA system, and then compares the system with the Leacock and Chodorow method, which has been proved to be the best among methods that calculate the result of similarity calculation among systems and the similarity among concepts. Finally, Chapter 5 presents the result of this study and suggests further research.

2 Literature Review

CI is being studied vigorously in the fields that require group ability rather than individual ability. Franks *et al*. [10] observed the process in which insect groups find their best nests from the perspective of information gathering, evaluation, deliberation, opinion gathering, selection, and execution, discovered how the insects make a rapid decision as a group, and compared the decision-making process with mathematical modeling.

Kube and Zhang demonstrated the superiority of CI through such studies as examined the ways robots equipped with sensors performed their duties without central control [2]. Also, Pham *et al.* proposed an optimization method that uses not a

solution but multiple solutions on the phases of problem-solving to get the optimal solution [6]. This paper, too, demonstrates the superiority of CI by proposing a system generating customized information modeled after the behavior of web users who solve personal questions by using other people's experiences, i.e., CI. In this paper, we define CI as a collection of other people's experiences, refers to them as 'cases,' and generates at customized information by utilizing CI that has been translated into cases. In order to find similar cases, ontology-based similarity calculation is in wide use. As for the methods calculating the semantic-similarity between two concepts using ontology, there are such methods as edge counting, information content, the fuzzy method, and a hybrid method that utilizes all these methods [10 and 11]. A method calculating the distance between two concepts that are the object of ontological comparison, edge counting compares the similarity by calculating the shortest distance between the two concepts [12]. This method is being widely used because its calculation is easy and it elicits the similarity between concepts pertinently [13]. However, this method has a shortcoming in that it cannot consider the latent associations that may exist between words, because it calculates the similarity between words by using only the distance as shown on the graph. The method based on information content and the hybrid method utilizes the probability of the appearance of each concept calculating the frequency of the appearance of the concepts in a particular document. These methods graft the concept of information content onto the assumption of information theory that the more sharing information between two concepts, the more similar they are [14]. The method of information content calculates the importance of a concept as the probability of appearance of a specific concept, and this means that a vocabulary is needed to find the probability. The frequency or probability of appearance of a word cannot be calculated in cases where the document is relatively short and the number of the words contained in the document is small as in information posted on blogs or social networking sites. And the OCRINA system, which generates cases by using information posted on blogs or social networking sites, is not a method fit to measure the probability, because its structural characteristic prevents it from playing the role of a vocabulary book. For that reason, this paper calculates similarity by using LSA [9]. Based on the assumption that semantic structures are inhered in documents, LSA analyzes the relationship between documents, words, or a document and a word. It is being widely used in human-machine interaction and text classification [9 and 15].

3 The Overall Architecture of OCRINA System

As depicted in Fig. 1, the problem-solving process of the OCRINA system consists of the 5Rs – Reap, Retrieve, Reuse/Revise, and Retain.

3.1 Reap Phase

First of all, we collect the CI that is dispersed on the web documents to create a case. A case in the case base of the OCRINA system is structured as in the formula (1). It is made up of a case description that shows a traveler's request and the solution, which is the information that satisfies the traveler's request. The case description takes the

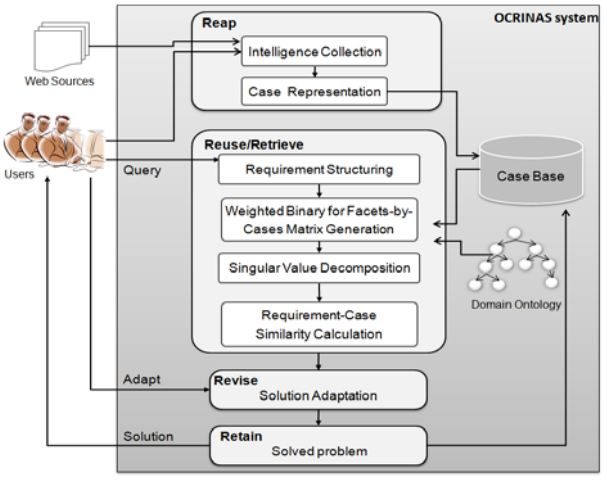


Fig. 1. Problem-Solving Procedure of the OCRINA

triple form of an index, the value, and the traveler's preference for the index. The traveler may have a variety of requests depending on the purpose or destination of the travel, which means the generation of a case consisting of a combination of diverse indices. Priorities are to discern the factors considered most important by the traveler in planning the travel. They are expressed in integers from *1* to *j*, and when they are difficult to discern, the priority is fixed at 1 so as not to be considered.

$$Case\ C_n = \{Des((s_{c_n1}, f_{c_n1}, p_{c_n1}), (s_{c_n2}, f_{c_n2}, p_{c_n2}), \ldots, (s_{c_nj}, f_{c_nj}, p_{c_nj})), \\ Sol(sln_{c_n1}, sln_{c_n2}, \ldots, sln_{c_nk})\} \quad (1)$$

In formula (1), C_n means the nth case in the case base, and $(s_{c_nj}, f_{c_nj}, p_{c_nj})$ is the j^{th}index of the n^{th} case, the value of the index involved, and the priority (j = *1, 2…, i_j*) of the j^{th} index. $(sln_{c_n1}, sln_{c_n2}, \ldots, sln_{c_nk})$ means the set of the solutions (k = *1, 2,…, t_k*).

3.2 Reuse and Retrieval Phase

A traveler who wants to generate customized travel information provides his or her requirements in the system. The requirements take the form of new problem in CBR, and the traveler enters the keywords that s/he is interested in with priority. As has been stated above, the same priority is automatically assigned to the index in which the traveler has not entered a value. LSA is applied in order to find cases that meet the traveler's requirements best. LSA is excellent at distinguishing the meanings of words and the degree of their similarity and also has the advantage of being able to infer the intrinsic relationship between words through powerful mathematical analysis, not their superficial relationship [16].The first step to apply LSA is to transform the value of the traveler's requirement in the m×1 matrix, and the case stored in the case base in the m×n matrix. The process in which these matrices are created is as follows.

In order to apply LSA to CBR, the requirement-by-case matrix$A(m,n)$, which shows whether a particular requirement is included in the cases similar to the term-by-document matrix, which shows whether words appear in a certain document, are produced. As stated above, the row of matrix$A(m,n)$ consists of the m requirements on the list of the traveler's requirements, and the column consists of the *n* cases in the case base. a_{ij}, an entry of$A(m,n)$ can describe whether the requirement of the j^{th} case includes the i^{th}requirement traveler or the frequency. However, since it is meaningless to determine the frequency of a certain requirement for a case with a structured form, the value a_{ij} of $A(m,n)$ amounts to (2) when only the question of the appearance is considered. Here the entrya_{ij} of $A(m,n)$ in formula (2) only considers grammatical similarity.

$$a_{mn} = \begin{cases} 1, & \text{if} v_m = v_{C_n j} \\ 0, & otherwise \end{cases} \quad (2)$$

In the next step, a_{ij} is revised after the semantic similarity like Synonym, Hypernym/Hyponym, and/or Melonym/Holonym is considered. In other words, $A(m,n)$revises by reflecting the conceptual connection between the value of a requirement and the value of a case (Formula (3)). The conceptual connection is discerned by ontology.

$$a_{mn} = \begin{cases} 1, & \text{if} v_m = v_{C_n j} \text{or } v_m \sqsupset v_{C_n j} \\ 0, & otherwise \end{cases} \quad (3)$$

However, formula (3) cannot show the degree of the association that exists between the concepts. That is, it cannot tell whether the value of a requirement and the value of a case are same or different, nor can it show the semantic distance between the two concepts when they are different. In the next step, the semantic distance between the two concepts is considered. The semantic distance is calculated by using ontology, too. To calculate the semantic distance, we adopt a semantic weight (w_s)that is a weight reflecting the semantic connection between the two values of $v_{C_n j}$and v_m. Since the semantic distance between two different values, i.e., the number of the concepts that exist between the two values in ontology can vary depending on the situation, it is necessary to adjust the entry so that it reflects such a difference. The semantic weight for matrix$A(m,n)$is arrived at as in Formula (4). In order to calculate the semantic weight, first, find the two semantically related concepts in ontology. Second, the shortest path that has the two concepts as the terminal node is drawn out of the ontology. Third, the semantic weight is calculated using the number of the concepts that exist in the shortest path. To compensate the range of the variation of the weights due to increasing of the numbers of concepts the shortest path and maximize the weights of concepts that are conceptually closed are evaluated using square root.

$$w_{s_{mn}}(v_m, v_{C_n j}) = \frac{1}{\sqrt{node(v_m, v_{C_n j})}} \quad (4)$$

In Formula (3), $w_{s_{mn}}$ is the semantic weight for the entry of $A(m,n)$ and $node(v_m, v_{c_nj})$ is the number of the nodes between v_m and v_{c_nj}. The entry of $A(m,n)$ that considers the semantic weight is shown in Formula (5).

$$a_{mn} = \begin{cases} 1 \times w_{s_{mn}}, & \text{if} v_m = v_{C_nj} \text{or} v_m \sqsupset v_{C_nj} \\ 0, & otherwise \end{cases} \quad (5)$$

A(m, n) is revised by the priority of the traveler. Priority weight (w_p) is a weight reflecting the difference that takes place between priority p_m (which is one of the travel's requirements) and p_{c_nj} (which is the priority of v_{c_nj}, which make up the cases). The priority weight is first measured in order to reflect in the calculation the differences in the indices and their values depending on the preferences of the traveler. The way that w_p is arrived at is shown in Formula (6).

$$w_{p_{mn}}(v_m, v_{C_nj}) = \frac{1}{|p_m - p_{C_nj}| + 1} \quad (6)$$

Input : Req{$(s_1, f_1, p_1), (s_2, f_2, p_2), \ldots, (s_m, f_m, p_m)$} and cases in the case base
Output : Values-by-Cases Matrix filled with appropriate values

1. **create** Requirement-by-Cases matrix $A(m,n)$
2. **for** each a_{mn} **do** // for all cells in A
3. **Fill** a_{mn}
4. **if** $v_m = v_{C_n j}$ **then** // when two facets exactly match
5. **set** $a_{mn} = 1$
6. **else if** $v_m \sqsupset v_{C_n j}$ **then** // when v_m subsumes $v_{C_n j}$
7. **set** $a_{mn} = 1$
8. **else**
9. **set** $a_{mn} = 0$
10. **endif**
11. **endfor**
12. **for** each $a_{mn} = 1$ **do** // for cells with value of 1
13. **check** subsume relations between v_m and $v_{C_n j}$
14. **if** $v_m \sqsupset v_{C_n j}$ **then** // when v_m subsumes $v_{C_n j}$
15. **set** $w_{s_{mn}}(v_m, v_{C_nj}) = 1/\sqrt{node(v_m, v_{C_nj})}$
16. **endif**
17. **check** priorities between v_m and $v_{C_n j}$
18. **set** $w_{p_{mn}}(v_m, v_{C_n j}) = 1/(|p_m - p_{C_n j}| + 1)$
19. **reset** $a_{mn} = 1 \times w_{s_{mn}} \times w_{p_{mn}}$
20. **endfor**
21. **for** each $\sum_{h=1}^{m} a_{hn} = 0$
22. **delete** C_n
23. **endfor**

Fig. 2. Algorithm arriving at the entry a_{mn} of $A(m,n)$

In Formula (6), $w_{p_{mn}}$ is the priority weight for the entry of$A\ (m,n)$, $\left|p_m - p_{C_nj}\right|$is the difference between the two priorities. The entry a_{mn} progression, which reflects the two weights drawn out above, is shown in Formula (7) as follows.

$$a_{mn} = \begin{cases} 1 \times w_{s_{mn}} \times w_{p_{mn}}, \text{if} v_m = v_{C_nj} \text{or} v_m \sqsupset v_{C_nj} \\ 0, otherwise \end{cases} \quad (7)$$

The process in which the value of $A(m,n)$is arrived at can be summarized as in Figure (2). The most optimal case is determined by calculating the cosine similarity in the most optimal vector space, after$A(m,n)$, which is drawn above, is analyzed in vector space. The drawn-out solution is finally put into the case base after it is adapted and used by the traveler to suit his purpose or its result is posted on his blog. The 5R process of CBR is concluded with the Retain process, which stores the proposed result in the case base.

4 Experiment and Evaluation

This paper has performed an experiment with a travel domain to prove the superiority of the OCRINA system. For this purpose, a case base containing 80 cases and an ontology consisting of 185 classes and 2,884 instances have been utilized. For the experiment, 15 types of traveler requirements have been provided by 5 travelers who were planning their travels.

req = {(Region, Seoul, 1), (tourType, historyTour, 2), (hotelRate, 4Star, 3), (touristParty, 4, 4)}

As for this requirement, shown in Table (1) is the result when the OCRINA system is compared with the Leacock and Chodorow (L/C) method, which have been known to be the best at calculating the similarity between concepts.

Both systems elicited the 31st case as a similar case, and the L/C method drew out the 33rd case at the same time. As to the 33rd case, whereas the value of similarity

Table 1. Comparison Results of the Calculation of Similarity

	Case no.	Sim.	Region	tourType	Hotel	Solution
Requirement			Seoul	History	4Star	
OCRINA	C_{31}	0.996	Jongro	History	4Star	dest: Gyeongbok_Palace, Deoksu_Palace, Gwanghwamun accm: Grand_Hayatt_Hotel_Seoul cost: $620
Leacock& Chodorow	C_{31}	0.674				
	C_{33}	0.674	Gangnam	Convention	4Star	dest: SETEC, KINTEX,COEX; accm: Novotel_Gangnam cost: $715

was high, the degree of the traveler's satisfaction was very low. With regard to a total of 15 requirements, both the OCRINA system and the L/C method judged the identical requirements, *Req1, Req4, Req7* and *Req14*, to be the most similar. However, the OCRINA system can't be claimed to be superior to the other methods just because the result of the OCRINA system is larger than that of the L/C method. Therefore, in order to prove the superiority of the OCRINA system, additional experiments such as the following were conducted. That is, after a comparison of the value of the index and the value of the requirement that make up each case with regard to one outstanding case drawn out by the two methods, their added weights were figured out, and the values of the final results were compared.

The value of the index, which make up the case, and the value of the requirement were compared, and an arbitrary weight was granted. The total score (TS) was found for the value of index by utilizing this weight, and the final results were compared (Formula (8)).

$$\mathrm{TS} = \frac{\sum_{j=1}^{n}\sum_{i=1}^{4} w_i t_j}{\sum_{j=1}^{n} t_j} \tag{8}$$

In Formula (8), w_i is the weight indicating the degree to which the values and the priorities match. Assuming that $w_2 = 0.75$, $w_3 = 0.5$, $w_4 = 0.25$, the total score was calculated for the case drawn out after the calculation of the similarity. The result of the calculation shows that the OCRINA system drew out a larger value for six requirements (Req_3, Req_6, Req_7,Req_{10}, Req_{12}, Req_{14}), and for another six requirements (Req_1, Req_2, Req_4Req_6, Req_8, Req_{15}), the two methods drew out an identical value. On the other hand, the L/C method produced a larger value only for the case drawn out with regard to three requirements (Req_1, Req_2, Req_4Req_6, Req_8, Req_{15}) than the OCRINA system. This means that the OCRINA sytem is superior to the L/C method in distinguishing between similar cases in connection with a new problem. A comparison of the final total scores of the OCRINA system and the L/Cmethod can be formulated as in Fig. 3.

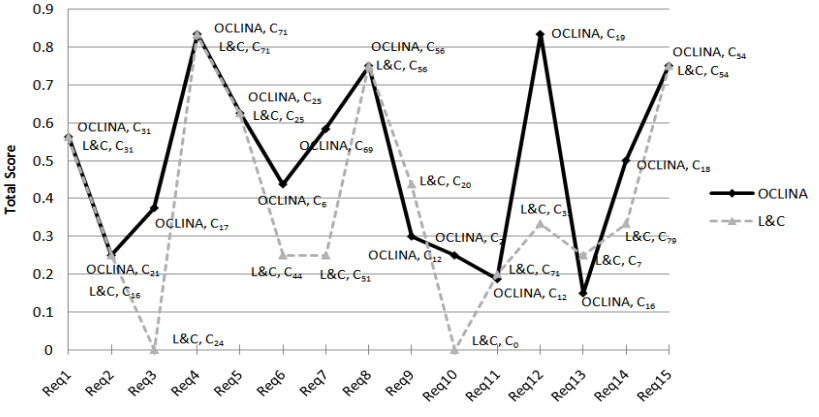


Fig. 3. Comparison of the Total Scores for the Cases

5 Conclusion and Further Research

This paper proposed the OCRINA system, which combines traditional CBR and web-based CI, in order to generate information customized for the traveler. The OCRINA system saw information posted on blogs or social network sites as an independent individual experience, collected such experiences, converted them into cases, compiled them, and translated them into collective intelligence. The OCRINA system utilized LSA and ontology at the same time in order to calculate the similarity that considers the latent connection and semantic connection that exists between the values that make up the requirements and cases. In order to prove the superiority of the OCRINA of calculating the similarity, a case base was established utilizing information posted on blogs or social network sites, experiments were conducted to draw out similar cases after requirements were entered by five users. The result of the experiments demonstrated the effectiveness of the method proposed in this paper as compared with the L/C method. The OCRINA system has the advantage of collecting web information most suitable for the traveler's requirements without going through the process of confirming all the search results individually. However, since the search for the case with the largest similarity alone cannot satisfy the user requirements, there is a need for studies to generate grammatically and semantically coherent information. Also, the result of the above experiments cannot be generalized because the experiments were conducted after the weight for the index was fixed at an arbitrary value. Therefore, in order to prove the superiority of the OCRINA system, additional experiments are needed with a diverse combination of weights.

Acknowledgments. This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract UD080042AD, Korea.

References

1. Heylighen, F.: Collective Intelligence an its Implementation on the Web: algorithms to develop a collective mental map. Computational and Mathematical Theory of Organizations 5(3), 253–280 (1999)
2. De Shutter, G., Theraulaz, G., Deneubourg, J.L.: Animal–robots collective intelligence. Annals of Mathematics and Artificial Intelligence. Kluwer Academic Publishers, Dordrecht (2001)
3. Kube, C.R., Zhang, H.: Collective Robotics: From Social Insects to Robots. In: Proc. Simulation of Adaptive Behavior, pp. 460–468 (1992)
4. Grant, L.L., Venayagamoorthy, G.K.: Swarm Intelligence for Collective Robotic Search. SCI, vol. 177, pp. 29–47 (2009)
5. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics–Part B 26(1), 29–41 (1996)
6. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. In: Proceedings of IPROMS 2006 (2006)

7. Chan, F.T.S.: Application of a hybrid case-based reasoning approach in electroplating industry. Expert Systems with Application 29(1), 121–130 (2005)
8. Sohn, M., Lee, H.J.: Context-based Similarity Calculation using CBR on Ontology. In: 2nd International Conference on Internet (ICONI) (December 2010)
9. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
10. Slimani, B.B., Yaghlane, K.M.: A New Similarity Measure based on Edge Counting. World Academy of Science. Engineering and Technology 23, 34–38 (2006)
11. Song, L., et al.: Fuzzy Semantic Similarity Between Ontological Concepts. In: Advances and Innovations in Systems, Computing Sciences and Software Engineering, pp. 275–280 (2007)
12. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics 19(1), 17–30 (1989)
13. Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G.M., Milios, E.: Information Retrieval by Semantic Similarity. International Journal on Semantic Web & Information Systems 2, 55–73 (2006)
14. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proc. 14th International Joint Conference on Artificial Intelligence, vol. 1, pp. 448–453 (1995)
15. Zelikovitz, S., Hirsh, H.: Integrating Background Knowledge into Nearest-Neighbor Text Classification. In: Advances in Case-Based Reasoning, EC CBR Proceedings, pp. 1–5 (2002)
16. Landauer, T.K., Foltz, P.W., Laham, D.: An Introduction to Latent Semantic Analysis. Discourse Processes 25, 259–284 (1998)

Integration of Collective Knowledge in Fuzzy Models Supporting Web Design Process

Jaroslaw Jankowski

Faculty of Computer Science and Information Technology
West Pomeranian University of Technology
ul. Zolnierska 49, 71-410 Szczecin, Poland
jjankowski@wi.zut.edu.pl

Abstract. The process of applications design focused on the Internet is usually based on continuous optimization and analysis of response from web users. The knowledge acquired from the target environment improves this process and can be used in design space modeling. This paper proposes an approach for fuzzy modeling of interactive objects' structure to determine the dependencies and provide the additional input towards increasing website effectiveness.

Keywords: collective design, website effectiveness, human-computer interaction, fuzzy modeling.

1 Introduction

The evolution of social media influences the development of platforms, which use the potential of collaboration. Collective intelligence adopted a different scope together with the development of the Internet and can be considered from the perspective of technical systems and philosophy or psychology as well [5]. It has a special significance in relation to social platforms in the trend of a so-called Web 2.0, virtual worlds and massive multiplayer systems [18][22]. The increasing engagement and the motivation of users result from various reasons and lead to different applications [14]. One of the developing fields in this area is a collaborative design, where the potential of community is used during the design process [23]. The methodical basis was known earlier, but the development of interactive platforms influenced its evolution and improvement [13]. Collaborative design systems generate multi-dimensional design space and require the appliance of decisions-support methods, among others, in the area of negotiations and expert systems [31]. They are applied in the field of manufacturing processes and distributed platforms [3]. In the situation where direct engagement of recipients is not possible, the collective knowledge extraction can be used to improve a design process [17]. This also relates to websites' participatory design [26]. In recent years, the important trend has been the design of effect-oriented websites, especially when they are a key element of business processes or sales platforms [24]. Available solutions relate to different areas connected with the increase of effectiveness and concentrate on the usage of experiment planning methods and limitation of decision space. The main goal of this article is to extend

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 395–404, 2011.

current approaches by integration of collective knowledge in web design processes with monitoring system and fuzzy models, which represent response for design space of effects-oriented websites. The knowledge acquired from the fuzzy model reflects the aggregated preferences of interactive content recipients. It constitutes a significant supplement of expert knowledge and enables adjusting the website parameters to the recipients' characteristics. The main result of the research presented in this paper is a new approach to testing and studying the interactive content and it provides the possibility of project solution assessment in the scope of its effectiveness.

2 Literature Review

In the web design process, there are different methodologies introduced. They depend on the area of applications, from websites, transactional systems to complex social platforms [16][10]. Design should fulfil the usability requirements analysed with the use of evaluation methods, heuristics and other approaches [6]. During the design of interactive elements the information architecture and user-centred design plays a crucial role [29]. It refers to the process of organising information and to the scope of the user's needs [4]. Apart from user-centred aspects, issues connected with design were outlined in the case of Internet websites or marketing content, whose main task is the generation of interactions [27]. The analysis of effects within the website is most commonly realized in relation to user actions that are desired by the website operators like new signups, redirections to target websites or transactions [28]. There are several elements of analysis, which should be measured and they extend behavioural analysis in the frames of session and dynamic modeling of navigation and temporal patterns [8]. Here are observed interdisciplinary occurrences from the area of human-computer interaction, which combines such fields as computer science, sociology, and social engineering [25]. The interactive messages are in such a case integrated with persuasion elements. There are identified elements connected with website design that influence the acquired results, like highlighted text, images, animations, headers, call to action messages with different colours and sizes [24]. All of them can be used with different combinations and variants and are resulting in a multidimensional design space. Here are available different approaches focused on increasing website effectiveness. As it is emphasized by M. Pearrow, the psychology of human factors, which includes the examination of various factors' influence on design, evaluation, and product usage, is more commonly applied in this field [19]. Intentional use of the dissonance theory of L. Fistinger and elaborates on the likelihood model developed by R. E. Petty and J. T. Cacioppo is also discussed [20]. The research confirmed that the perception of attached messages is one of the elements indicating effectiveness and achievement of goals assumed for a website. I. Lópeza and S. Ruiz proposed a conceptual website analysis model in which cognition and emotions are combined to aid in understanding behaviour in an online environment [12]. E. W. Welch and S. Pandey conducted research related to identification of website effectiveness in the area of target groups, content, technological issues, and consumer orientation [30]. J.H. Marcel and others, conducted research on effectiveness of corporate websites connected with employment, and they introduced assessment measures for websites from this field

[1]. With all this efforts the effects' maximization may lead to a situation where the increase of persuasion factors does not lead to further increase of website effectiveness, because of the limited users' ability to process information [9]. It may also generate negative side effects which are observed by a decrease of system usability. On the other hand, designers must take into account the occurrence of limited perception and sensory adaptation [2]. There is limited opportunity to catch the web user's attention, because interaction occurs in the form of quick content scanning and seeking significant information [11]. The combination of those areas leads to an increase of website invasiveness, which results on the other side in the decrease of effectiveness. The available publications relate to different methods of increased effectiveness, but the issue concerning the level to which persuasion should be increased has not yet been solved. One of the possibilities proposed in this article is a balanced approach in a design. It enables the acquisition of an acceptable level of effects without having excessive influence on users. As a basis for this approach, we propose in the next parts of this article the use of fuzzy models in the design process, which represent collective knowledge and preferences from web audiences.

3 Methodological Aspects of Proposed Approach

The website's components and interfaces can be represented in a form of interactive objects. They integrate the elements that influence users and provide the possibility of generating individual interactions. For every object there is defined a set of $E=\{E_1,E_2,\ldots,E_n\}$ elements and a number of possible project variants for each of them as $E_i=\{v_{i,1},v_{i,2},\ldots,v_{i,n(i)}\}$. The number of variants for the *i-th* element is represented by n(i). We assume that for each variant $v_{i,j}$, there can be defined the level of persuasion $p_{i,j}$, dependent on text messages, colours, and animations. The object constructed in this way can be used with different combinations of variants for gathering collective characteristics of target users. The main goal is to produce a final design with optimal set of components with acceptable effectiveness and persuasion without negative side effect on usability. Our approach uses collective knowledge from web environment to build an inference system that can be included into the design process. Due to imprecise character, it is not always possible to define measures for levels of persuasion based on crisp numbers. It is more natural to formulate them using linguistic terms like *low persuasion*, *medium persuasion*, *high persuasion* and representation based on fuzzy sets defined by L.A. Zadeh, where a single element can belong to different sets with different strength based on a membership function [32]. One of the applications of fuzzy sets are expert systems usually integrated with inference models introduced for the first time by Mamdani, and their extensions [15]. The main assumption for knowledge processing is to reflect the model inputs in the form of vector X on the output Y^* with minimization of mean square error for reflection of X on Y output from the real system. The structure of general inference system with rule base, fuzzyfication and defuzzyfication modules is presented in Figure 1. For the purpose of modeling the collective knowledge from web users, the adaptive neuro fuzzy inference system introduced by R. Jang was used [7]. Its structure is based on five levels and parallel information processing with the possibility for adaptation.

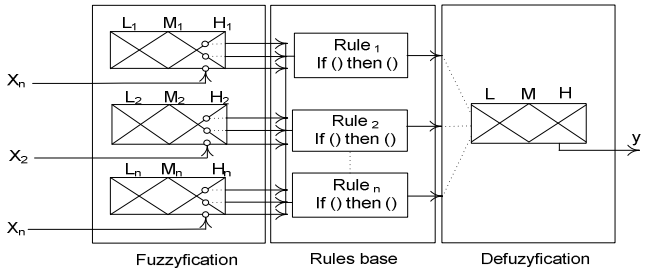


Fig. 1. General fuzzy model structure [21]

In the next step, fuzzy models where integrated with the measurement system and a design environment, with main goal to generate a response surface and deliver knowledge about web users' characteristics. Fig. 2 shows the structure of the system focused on testing different variants and generation of models. The interactive object structure, which consists of elements $E_1,E_2,\ldots,E_i$ was parameterized. For each of them, variants $v_{i,j}$ where determined with quantitative and qualitative characteristics, as well as the persuasion levels $p_{i,j}$. The selection module SM generates a project variant and it is directed to the user's, located in the interactive environment IE. In the measurement system MS, there are collected data about individual expositions and acquired results. Data and response R are passed over as data feed for the fuzzy modeling system FM, where the model structures are supplied in a feedback process. In the decision support system module, there occur integration of data from inference models and inputs from design teams D, management M, and marketing experts S.

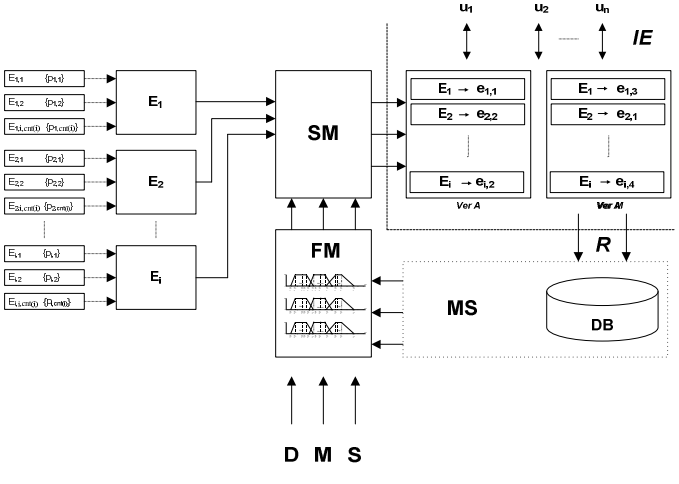


Fig. 2. Structure of measurement system integrated with fuzzy models

Results obtained from models are used in decision processes related to design and website optimisation towards higher level of effectiveness, and are redirected to the

SM module. The main purpose is to determine the optimal range of effects and its relation to the levels of persuasion. Knowledge acquired from the system represents target users' characteristics and it extends inputs from experts available during the design phase. The presented approach can be used in effectiveness monitoring and detection of situations when the level of message persuasion grows without significant improvement of effects coefficients. In that situation, increased invasiveness can result in a negative influence on a user, and as a result a lower level of interest in a given service or website. That's why the maximization of effects is assumed, as well as achievement of compromise level where there can be minimized level of persuasion. System based on fuzzy models captures characteristics with more natural way than other available solutions, especially when it comes to inputs, where it is not possible to define crisp sets. In the remaining part of this article, the experimental research was made in a real environment with main goal to generate the data feed for the fuzzy model and to verify the presented approach with the use of real data.

4 Integration with Online Environment and Empirical Research

The following phase of research included engineering component construction, integration with website, initial data analysis, and construction of a fuzzy inference model. The interactive object was part of a website with the main purpose to redirect users to informative landing pages of a social platform within the VirtuLabs project. The designed interactive object consisted of three sections: S_1, S_2 and S_3. In section S_1, the text influence elements were located; in section S_2, a graphical call to the action element; and in section S_3, there were graphical elements with informational functions. In these sections there were designed four influence elements $E=\{E_1,E_2,E_3,E_4\}$, which were subject to versification. In the object area the text element T_1 was located, which was not subject to versification and a permanent graphical element I_1. The additional component, turned on and off for individual project variants, was B_1, which represented the message elimination button. Fig. 3 presents the object structure.

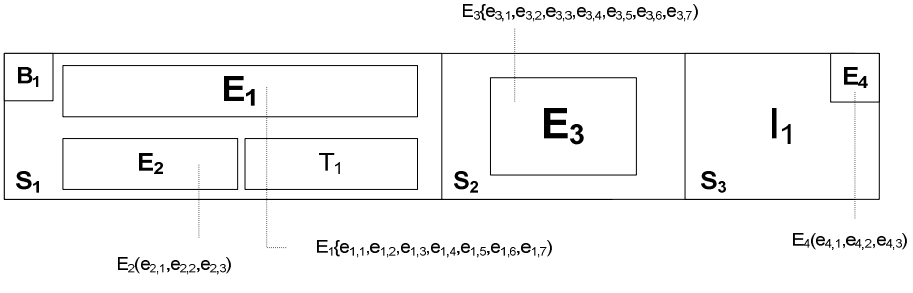


Fig. 3. The structure of an experimental interactive object

For element E_1 there were seven versions of text variants $\{E_{1,1}, E_{1,2}, E_{1,3}, E_{1,4}, E_{1,5}, E_{1,6}, E_{1,7}\}$ with different levels of influence and integration of call to action expressions. For text element E_2 there were three text variants $\{E_{2,1}, E_{2,2}, E_{2,3}\}$ with

variable influence levels and gradual call to action. Text object $E_{2,1}$ included call to action message, $E_{2,2}$ included a strengthened call to action. The element $E_{2,3}$ was integrated with a message with the additional determination of services free of charge. Element E_3 was designed in the form of a graphical button with a call to action text in seven project variants $\{E_{3,1}, E_{3,2}, E_{3,3}, E_{3,4}, E_{3,5}, E_{3,6}, E_{3,7}\}$. The first two variants $E_{3,1}$, $E_{3,2}$ were static, $E_{3,4}$ was based on flashing animated text, variant $E_{3,5}$ was flashing more intensively up to $E_{3,6}$, $E_{3,7}$ with the highest flashing frequency and contrast and high impact on user. The last element E_4 with persuasion functions, included information about free accounts in three design variants. Element B_1 was an additional element with two possible states, with the option to remove the interactive object from the user's screen. It helped to measure invasiveness level and detect at what stage users where more willing to disable this content. As a result, the presented object had V=882 possible design variants. In the next phase, experiment realization in a real environment was determined and the object was displayed 249,149 times within a test web page. Every possible combination of elements was exposed approximately 282 times. The message was generated for 27,338 unique users. In the analysed period the system registered 698 interactions with different response R measured as (number of interactions/number of impressions) * 100%. Table 1 shows the chosen variants for the highest and lowest response values.

Table 1. Subsets of response values for selected variants

High response instances				
E_1	**E_2**	**E_3**	**E_4**	**R[%]**
1	1	1	2	1.4652
1	2	2	3	1.4652
6	1	7	2	1.4705
6	2	5	1	1.6326
2	2	6	3	1.7421
4	1	6	2	1.7482
4	2	3	1	2.2222
5	1	6	3	2.4054

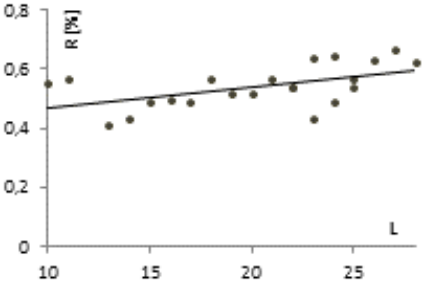


Fig. 4. System response depending on the aggregated value of influence measure

The analysed response of the R system was characterized by a click through ratio coefficient determining the relation of redirections from an individual project variant to its number of expositions. In the next phase an analysis of the aggregated persuasion levels L_i was made, based on the summarized persuasion of each element, and the average response R values where determined. Fig. 4 shows the interaction distribution depending on the influence levels. It is possible to observe, according to the trend line, the increase of influence from the level of R=0.49 to the level R=0.62, which gives an approximate increase of 26% of the system response during the increase of the aggregated persuasion level. Combination of influence elements does not significantly cause an increase of R in the geometrical way. This increase is small, taking into account the differences in response between the variants with the smallest influence level without animated and flashing element. It indicates the possibility of the occurrence of limited perception: when the influence element is already visible,

other effects are not efficient. The presented analyses reflected the characteristics of a system and indicated the occurring dependencies between persuasion levels and response from the system. Information gathered in this process can be used in design optimisation and makes additional sources of knowledge in the design process. The aim of the next stage is to build knowledge representation for gathered data sets and to construct a system based on fuzzy inference models with linguistic description of design components and their relation to output values.

5 Representation of Collective Knowledge in Fuzzy Models

The initially conducted analysis determined dependencies between the components, their persuasion, and response levels. Based on these inputs, the structure of the fuzzy model was developed, for which the supply of a measurement system was established for individual parameters. The aim was to build an inference system that would reflect the registered occurrences and enable the modeling of dependencies between system inputs and output. It was possible to generate a surface of system response and to determine the system characteristics in a form of collective knowledge. At this stage Matlab^{TM} environment was used. The input parameters reflected the individual components of vector $E_i=[e_1,e_2,e_3,e_4]$, whose values were acquired from empirical data. In order to determine the influence level, the values of R coefficients were adapted for individual sets of parameters. On the output the response levels were obtained, which were determined for the parameters' vector. Fig. 5 shows the system structure, for which four inputs were determined E_1, E_2, E_3 and E_4 relating to the components of the interactive object. For individual elements the fuzzy representation was determined by influence level. For example, Fig. 6 presents the distribution of the membership function in the Gaussian shape for element E_1 with the division of seven influence elements into three blocks. It helped to reduce the number input parameters and it was possible to replace all possible variants with a membership function and as a result *low*, *medium* and *high persuasion* levels were determined.

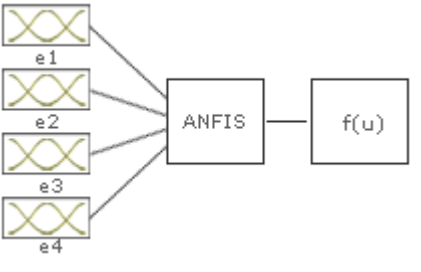


Fig. 5. Structure of fuzzy inference system

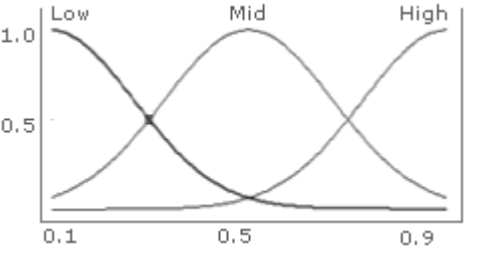


Fig. 6. Membership functions for parameter e_1

The model's accuracy was tested for three different shapes of the membership function. The best results were acquired for the Gaussian function and such

distribution was applied in the final model. For every input the number of membership functions was also analysed. The satisfying results were also acquired with three membership functions for individual inputs. The training process was tested for 20, 50, 100, 200 and 300 epochs. Training above 100 epochs did not cause improvement and was stopped in order to avoid the occurrence of overtraining. Here was achieved a form of representation of collective knowledge from audience. It can be used in design process to analyse responses and helps in components selection. Fig. 7 presents the surface of system response depending on the input E_3, E_4 which shows the dependencies between system input and output and determines the influence of individual components E_1 and E_4 on response surface.

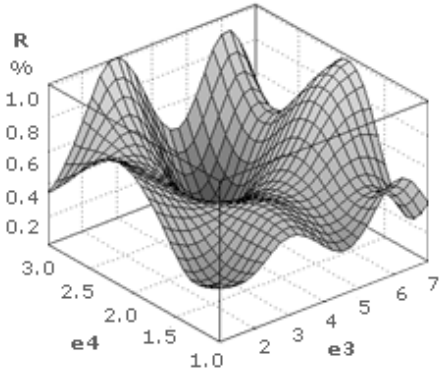


Fig. 7. Dependencies between parameters E_4 and E_3 and output R of fuzzy model

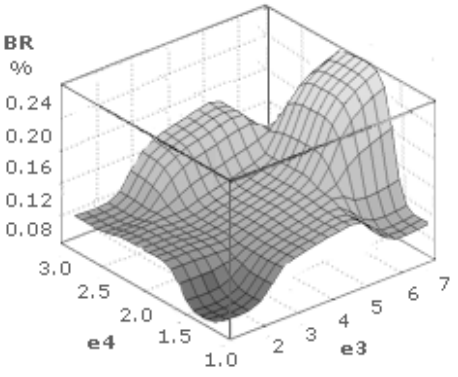


Fig. 8. Model of blocking response BR for e_3 and e_4 parameters

Fig. 8 represents the model generated for the blocking response BR, calculated by the number of blocking actions, divided by the number of impressions as a measure of invasiveness. The highest number of attempts to remove interactive content where obtained for element e_4 with variants number two and six with the seventh variant for element e_3. The models deliver information on which variants are not increasing the total response from the system, and what level of response can be obtained with smaller persuasion. The presented approach can be used as additional source of knowledge in the design process, and the response surfaces from models show dependencies between used text, graphical components and animations. Response from the target environment can be other than that expected by designers, and without modeling it would be difficult to determine the optimal set of parameters. Fuzzy approach can be used to prepare response surface on limited dataset and make approximation for other unexpected inputs. The construction of a fuzzy model that reflects the dependencies in an interactive system provides the possibility of determining the levels of influence of different project variants, and the system response enables the potential evaluation of individual influence elements and their relation to the acquired results.

6 Summary

Designing websites requires knowledge of audience characteristics and selection of influence factors in such a way so as to ensure a certain level of effect realization. Collective knowledge integrated into fuzzy inference models can be used during the design process and is an extension to available solutions. The obtained inference systems enable determination of the levels of system response for provided sets of values, representing the strength of persuasion of interactive object components represented by fuzzy numbers. The presented solutions show the alternative manner of interactive message construction, which ensures the possibility of influence level testing. This approach can be a basis for compromise design, where the determined level of effects is guaranteed, but the excessive message persuasion does not occur. The acquired results based on fuzzy modelling ensure the effects on the acceptable level without causing a significant increase in system invasiveness.

References

1. Birgelen, M., Martin, G.M., Dolen, W.M.: Effectiveness of Corporate Employment Web Sites: How Content and Form Influence Intentions to Apply. International Journal of Manpower 29(8), 731–751 (2008)
2. Burke, M., Hornof, A., Nilsen, E., Gorman, N.: High-Cost Banner Blindness: Ads Increase Perceived Workload, Hinder Visual Search, and are Forgotten. ACM Transactions on Computer-Human Interaction 12(4), 423–445 (2005)
3. Dai, X., Qin, Y., Lingxiao, M.L., Juster, N.: A Knowledge-Based Design Advisory System for Collaborative Design for Micromanufacturing. The International Journal of Advanced Manufacturing Technology 47(9), 973–979 (2010)
4. Green, D., Pearson, J.M.: Development of a Web Site Usability Instrument Based on ISO-9241-11. The Journal of Computer Information Systems 47(1), 66–72 (2006)
5. Halpin, H.: Foundations of a Philosophy of Collective Intelligence. In: Guerin, F., Vasconcelos, W.W. (eds.) AISB Convention: Communication, Interaction and Social Intelligence, AISB, pp. 12–19. University of Aberdeen, UK (2008)
6. Hassan, S., Li, F.: Evaluating the Usability and Content Usefulness of Web Sites: A Benchmarking Approach. Journal of Electronic Commerce in Organizations 3(2), 46–67 (2005)
7. Jang, J.S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. IEEE Transactions on Systems, Man and Cybernetics 23(3), 665–685 (1993)
8. Kelly, S.: Determining Effectiveness of Websites by Refining Transaction-Oriented Views through Supervised Competitive Clustering. Web Effectiveness Research Group, Research report, The University of Georgia (2008)
9. Lang, A., Potter, R.F., Bolls, P.: Where Psychophsiology Meets the Media. In: Bryant, J., Oliver, M.B. (eds.) Media Effects: Advances in Theory and Research, pp. 185–206. Taylor & Francis, Abington (2008)
10. Lin, Y.-C., Yeh, C.-H.: User-centered interface design of social websites. In: Yang, C.C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P.S., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K.M., Mao, W., Zhan, J. (eds.) ISI Workshops 2008. LNCS, vol. 5075, pp. 366–376. Springer, Heidelberg (2008)
11. Lindgaard, G., Fernandes, G., Dudek, C., Brown, J.: Attention Web Designers: You Have 50 Milliseconds to Make a Good First Impression. Behavior and Information Technology 25(2), 115–126 (2006)

12. Lópeza, I., Ruiz, S.: Explaining Website Effectiveness: The Hedonic–Utilitarian Dual Mediation Hypothesis. Electronic Commerce Research and Applications 10(1), 49–58 (2011)
13. Maher, M.L., Paulini, M., Murty, P.: Scaling up: From Individual Design to Collaborative Design to Collective Design. In: Gero, J.S. (ed.) Design Computing and Cognition, DCC 2010, pp. 581–600. Springer, Heidelberg (2010)
14. Malone, T.W., Laubacher, R., Dellarocas, C.N.: Harnessing Crowds: Mapping the Genome of Collective Intelligence. MIT Sloan Research. No. 4732-09 (2009)
15. Mamdani, E.H.: Applications of Fuzzy Set Theory to Control Systems: A Survey. In: Gupta, M.M., Saridis, G.N., Gaines, B.R. (eds.) Fuzzy Automata and Decision Processes, pp. 77–88. North-Holland, Amsterdam (1977)
16. Mohamed, T.I., Hamdollah, M., O'Brien, P.T.R.: Analysis and Design of e-Commerce Applications on the Web: A Case Study of OO Techniques and Notations. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2003. LNCS, vol. 2738, pp. 315–327. Springer, Heidelberg (2003)
17. Nikolopoulos, S., Chatzilari, E., Giannakidou, E., Papadopoulos, S., Kompatsiaris, I., Vakali, A.: Leveraging Massive User Contributions for Knowledge Extraction. In: Bessis, N., Xhafa, F. (eds.) Next Generation Data Technologies for Collective Computational Intelligence. Studies in Computational Intelligence, Springer, Heidelberg (2011)
18. O'Reilly, T.: What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, http://oreilly.com/web2/archive/what-is-web-20.html
19. Pearrow, M.: Web Site Usability Handbook. Charles River Media, Rockland (2000)
20. Petty, R.E., Wegener, D.T.: The Elaboration Likelihood Model: Current Status and Controversies. In: Chaiken, S., Trope, Y. (eds.) Dual Process Theories in Social Psychology, pp. 41–72. Guilford Press, New York (1999)
21. Piegat, A.: Fuzzy Modeling and Control. Physica, Heildelberg (2001)
22. Preece, J., Shneiderman, B.: The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. AIS Transactions on Human-Computer Interaction 1(1), 13–32 (2009)
23. Sanders, L.: Collective Creativity. LOOP: AIGA Journal of Interaction Design Education 6, 14–26 (2002)
24. Schlosser, A.E., White, T.B., Lloyd, S.M.: Converting Web Site Visitors into Buyers: How Web Site Investment Increases Consumer Trusting Beliefs and Online Purchase Intentions. Journal of Marketing 70(2), 133–148 (2006)
25. Sears, A., Jacko, J.: Human-Computer Interaction Handbook. CRC Press, New York (2007)
26. Spinuzzi, C.: The methodology of participatory design. Technical Communication 52(5), 163–174 (2005)
27. Tarafdar, M., Zhang, J.: Analysis of Critical Website Characteristics: A Cross-Category Study of Successful Websites. The Journal of Computer Information Systems 46(2), 14–24 (2005)
28. Udo, G.J., Marquis, G.P.: Factors Affecting e-Commerce Web Site Effectiveness. The Journal of Computer Information Systems 42(2), 10–16 (2001)
29. Vredenburg, K., Isensee, S., Righi, C.: User-Centered Design. Prentice Hall, New York (2002)
30. Welch, E.W., Pandey, P.: Multiple Measures of Website Effectiveness and their Association with Service Quality in Health and Human Service Agencies. In: Proceedings of the 40th Hawaii International Conference on System Sciences, pp. 115–135 (2007)
31. Xuan, F.Z., Du, H.: Knowledge-Intensive Collaborative Design Modeling and Support. Computer In Industry 57(1), 56–71 (2006)
32. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)

WordNet Based Word Sense Disambiguation

Andrzej Siemiński

Institute of Informatics, Wroclaw University of Technology
Wybrzeże St. Wyspiańskiego 27
50-370 Wrocław, Poland
Andrzej.Sieminski@pwr.wroc.pl

Abstract. Due to the intrinsic ambiguity of a natural language the word sense disambiguation or WSD is a challenging task. The paper uses WordNet for (WSD) for that purpose. Unlike many others approaches on that area it exploits the structure of WordNet in an indirect manner. To disambiguate the words it measures the semantic similarity of the words glosses. The similarity is calculated using the SynPath algorithm. Its essence is the replacement of each word by a sequence of WordNet synset identifiers that describe related concepts. To measure the similarity of such sequence the standard tf-idf formula is used. At the last stage a modification of Ant Colony Optimization for the Traveling Salesman Problem is responsible for word disambiguation.

Keywords: word sense disambiguation, WordNet, tf-idf, Ant Colony Optimization, SynPath.

1 Introduction

The task of Word Sense Disambiguation or the WSD is to assign a sense to an instance of a polysemous word in a particular context. The WSD is considered to be one of the most difficult problems in Artificial intelligence. By analogy to NP-completeness in complexity theory WSD has been described as an AI-complete problem.

A sense is selected from a set of pre-defined word senses. Each WSD approach relies on a source of knowledge for words meaning. The source must to be precise, comprehensive and reliable. All these requirements are fulfilled by WordNet which could be interpreted as a lexical ontology. The development of Internet has brought among others the availability of a vast amount of unstructured data, such as Web pages or blog corpora. Traditional techniques for text mining and information retrieval of such data are not adequate as they are based on lexical, syntactic and statistical analysis of text. They do not go beyond the surface appearance of words. Often they fail to find important texts and due to the differences in used vocabulary. On the other hand, not being able to identify the proper meaning of a word, results in providing documents that are clearly not relevant to the user needs. Reliable and fast text disambiguation can help also in achieving a major breakthrough in the realization of the so-called semantic Web. Other application areas include machine translation (MT), information extraction, speech recognition, semantic mapping (SM), semantic annotation (SA), and ontology learning (OL).

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 405–414, 2011.

The main contribution of the paper is the presentation of a blend new and modified algorithms that are capable of rudimentary WSD in an efficient manner that will make it useful for a variety on-line applications.

The remainder of the paper is organized as follows. Section 2 presents a brief survey of different approaches to the Word Disambiguation. Section 3 is devoted to the problem of measuring the text similarity. It introduces the core concept of WordNet SynPaths and rationalizes their usage for the WSD. Section 4 describes the modification of the ACO algorithm used for disambiguation of the word senses. The discussion of preliminary experiment results is presented in Section 5. A short conclusion in Section 6 finishes the article.

2 Survey of Word Sense Disambiguation Data Sources and Techniques

WSD techniques differ substantially in the used source of knowledge, the way of operation and achieved error ratio. The errors are inevitable, even humans make errors. Having all the knowledge of the language and the world around us, the consistence of word tagging performed by trained linguists is not as high as would be expected [1-2]. The selection a technique must take into account: the availability of language data, the acceptability of required time and typically achieved error ratio. A concise survey of WSD techniques is presented in [3] and a comprehensive, in depth and up-to date overview could be find in [4]. The knowledge sources used for the WSD come from lexical knowledge released to the public (e.g. WordNet) or are learned from a training corpus.

The lexical knowledge usually has form of some sort of dictionary and is used in unsupervised WSD approaches. The sources include:

- Sense Frequency. The most frequently used sense is always selected. Astonishingly, the performance of such a naive approach is not very bad and it often serves as the benchmark for the evaluation of other WSD algorithms.
- Sense glosses. A gloss is a brief explanations of a word sense with definitions and examples. To asses the similarity of glosses the techniques developed for text retrieval are used.
- Concept Trees. The trees connect words to concepts and exploit a variety of relations between the concepts. The most notable source of such data is WordNet [5]. The similarity of word senses is measured from the hierarchal concept tree.
- Selectional Restrictions. The principle used here is somewhat opposite to that of Concept trees. In that case the semantic restrictions placed on the word sense are used [6]. That kind of information is acquired from e.g. by the Longman Dictionary of Contemporary English.
- Subject Codes. They refer to the category to which one sense of the target word belongs [7]. Typically, subject code alone is not enough to tag a word sense. Weighted indicative words for each subject code are used to determine the subject code that the current instance of the word is likely to belong to. The sense is selected according to the association between word sense and subject code.

  The proposed here approach uses a combination of sense glosses and concept trees.

The learned word knowledge constitutes a second kind of data source. It is acquired from the context of training corpora. It includes such data as:

- Indicative Words. They are meaningful words that surround the target and can serve as the indicator of target senses.
- Syntactic Features. Knowing whether a specific word appears in the position of subject, direct object or indirect object could make the disambiguation more straightforward.
- Domain-specific Knowledge. It can only be acquired from training corpora and can only be attached to the WSD by empirical methods, rather than by symbolic reasoning. The limitation is that it could be defined only for specific and narrow domain such as e.g. terrorism [8].
- Parallel Corpora. These are multilingual corpora of texts translated into different languages. First the verbs and nouns in different languages are aligned. The translation process implies that aligned pair words share the same sense or concept so we can use this information to sense the meaning of words [9].

The used algorithms are divided into two broad groups: the supervised and unsupervised approaches. The members of the former group do not require a training corpus and needs less computing time and power. It is suitable for online machine translation and information retrieval.

Simple approaches algorithms reference only one type of lexical knowledge. The precision and recall are typically not satisfactory. Using a combination of two or more types of lexical knowledge looks attractive [10]. It mimics the way people work but the questions of the selection of knowledge types and their relative weight remains unanswered yet.

The iterative [11] and recursive filtering [12] both assign senses not in one step but rather gradually purging the irrelevant senses and leaving only the relevant ones. The process should stop within a finite number of processing cycles. The recursive filtering is more flexible, it does not disambiguate the senses of all words until the final step whereas the iterative approach starts with the tagging of the words with high maximum confidence.

The supervised models are divided into two classes: hidden models and explicit models. All of them use sense-tagged corpora to train the sense model, which makes it possible to link contextual features (world knowledge) to word sense. The increasing availability of training corpora make such approaches attractive to the research community. They are more complex than the unsupervised algorithms and even a brief discussion goes beyond the scope of the paper. The proposed method applies an unsupervised model.

3 WordNet

A WordNet is a comprehensive dictionary and lexical data base for a language. It can be interpreted and used as a lexical ontology in the computer science. It includes nouns, verbs, adjectives and adverbs. Its elementary building block is a synset - a set of words or collocations (word sequences) that are all mutually synonym. A sysnset represents a unique word meaning. In what follows word disambiguation simply

means selecting a proper sysnet. In a WordNet the sysnset meaning is described by the set of words that form it and it is further explained by a gloss.

Synsets are connected to other synsets via a number of semantic relations e.g. the nouns synsets are connected by 5 relations: hypernym, hyponyms, coordinate terms, holonym and meronym. In the study the Princeton WordNet (PWN) was used. It is the first WordNet ever created and is commonly used as a reference for other wordnets and for other wordnet-related activities. After years of development it covers in a detailed manner virtually the whole English language.

The proposed approach attempts to disambiguate words senses using their systesets' glosses. The assumption is that the sysnests with related meaning should be described by glosses that are semantically related to each other. The nouns 'king' and 'queen' have both 10 glosses each, some of them are shown in the Table 1.

Table 1. Selected synsets for the words king and queen

Word	SynsetID	Synset members	Gloss
king	king#1	king, male monarch, Rex	a male sovereign; ruler of a kingdom
	king#2	king, queen, world-beater	a competitor who holds a preeminent position
	king#3	king	chess a checker that has been moved to the opponent's first row where it is promoted …
	king#4	King	the weakest but the most important piece
queen	queen#1	queen	the only fertile female in a colony of social insects such as bees and ants and termites; its function is to lay eggs
	queen#2	queen, regnant	a female sovereign ruler
	queen#3	queen	one of four face cards in a deck bearing a picture of a queen
	queen#4	queen	chess the most powerful piece

Cleary in the context of monarchy the synsets king#1 and queen#2 are appropriate, whereas for chess playing the synsets king#4 for king and queen#4 should be selected. This is reflected by the semantic similarity of respective glosses. The gloss texts are short and to compensate the unavoidable differences in the used vocabulary semantic methods must be employed.

The first step is to measure the similarity of pairs or words. The hierarchical structure defined by hypernym and hyponym relations is often used for that purpose. Proposed algorithms [13] usually exploit one or more of the following features:

1. the number of links separating the words sysnsets;
2. the location of the Most Specific Common Abstraction, that is the synset most deeply located in the hierarchy that subsumes both words;
3. the number of all subordinate sysnets of each word.

Calculating all of the features requires many accesses to the database that represents the WordNet structure. This could pose a serious efficiency problem especially for the WordNets which are implemented using an intrinsically slow XML databases.

Having computed the similarity between all words pairs we can move to the even more complex task of measuring the text similarity. One of the most precise algorithms for calculating the similarity of texts t_1 and t_2 was presented in [14] and it uses the following formula:

$$Sim\ (t_1, t_2) = 0.5 * \left(\frac{mSim(w, t_2) * idf(w)}{\sum_{w \in \{t1\}} idf(w)} + \frac{mSim(w, t_2) * idf(w)}{\sum_{w \in \{t2\}} idf(w)} \right)$$

where mSim(w,t) is the maximum similarity of word w to any of words from text t As a consequence it requires the calculation of similarity of n*k/2 word pairs where n and k are the number of words in compared texts. It is clearly not feasible for any on-line application. A slightly less complex formula was proposed in [15] but it still requires many access to the WordNet data base.

The computational complexity of all of the traditional approaches has prompted the usage of the standard tf-idf (term frequncy – inverse document frequency) measure [18]

$$sim(t_j, t_k) = \frac{\sum_{i=1}^{N} w_{ij} * w_{ik}}{\sqrt{\sum_{i=1}^{N} w_{ij}^2} \sqrt{\sum_{i=1}^{N} w_{ik}^2}} \quad (1)$$

where:

$$w_{t,d} = tf_{t,d} \log \frac{|D|}{|\{dx \in D \mid t \in dx\}|} \quad (2)$$

where:

$t_j = (w_{1j}, w_{2j}, ..., w_{nj})$
$tf_{t,d}$ is the frequency of term t in text d
|D| id the is the number of all texts
$|\{dx \in D \mid t \in dx\}|$ if the number of texts that contain the term t.

The standard method is purely statistical in nature. It uses only the term frequencies in the compared texts and in the entire database. It totally lacks a semantic component. The similarity of different words that belong to the same synset is not taken into account.

In order to add the semantic component the words are replaced by their SynPaths. A SynPath is a sequence of synset identifiers that contains the identifier of the words' systeset and all identifiers of its superior synsets. Such a method of operation clearly increases the level of text similarity e.g. all words belonging to the same synset are considered identical. Therefore the retrieval recall increases. The question is whether the replacement will not introduce to many common identifiers from the top of WordNet hierarchy structure thus damaging the precision . In what follows we discuss the impact of introducing SynPaths on the text similarity as measured by the tf-idf algorithm.

The theoretical analysis assumes several simplifying assumptions. This is due to the complexity of modeling the real life WordNet structure and the term frequency of occurrence pattern. The assumptions are:

- The WordNet has a regular structure and at each level (excluding the last level of the so called leaf synsets) a synset has exactly b branches.
- A texts consist of only one term and the term appears only once in the text.
- Each leaf synset appears in only once in the text database.
- A term belongs to only one leaf synset.
- The base of the logarithm used in the equation (1) is equal to b.

Let:

- l_{max} denote the level of the leaf synsets.
- l(s) denote the level of synset s
- synSet(t) denote the synset of term t

Taking into account the above assumptions and Equations 1 and 2 are converted to:

$$|D| = b^{l_{\max}-1}$$

the weight of a term t that appears in text d is equal to:

$$w_{s,d} = 1 * \log_b \frac{|D|}{b^{l_{\max}-l(s)}} = \log_b b^{l(s)-1} = l(s)-1$$

The weight of a synsets increases linearly with the level at they are located in the structure. The root synset has the value of 0. The top level sysets identifiers are so common that they do not influence the search precision or recall. They are like a stop words in traditional indexing. It is worth noting that the first text retrieval systems used medium sized list of stop words which later grow up in size. Nowadays most systems do not used them at all [17].

4 Ant Colony Optimization

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. The ACO mimics the behavior of real world ants. The ants, which could be regarded as simple agents, walk through the search space in a more or less random manner laying pheromone trail. The amount of the pheromone deposits depends on the quality of the found solution. The pheromone trails in ACO serve as a communication medium fort the ants. It was initially proposed by Marco Dorigo in 1992 in his PhD Thesis [18] for solving the Traveling Salesman Problem. The problem belongs to the class of NP-complete problems. Since then the ACO was successfully applied is such diverse areas such as vehicle routing, image processing or job scheduling [19].

The paper [20] describes a Java ACS framework and two its implementations. The first one (ACS4TSP) is for the Traveling Salesman Problem (TSP) and the second one (ACS4MP) is for the Multicasting Problem in a network. During the study another

implementation (ACS4Disamb) was developed that extends the ACS4TSP. The aim of the ACS4Disamb is to identify the a set of most similar glosses. We believe that the paper describes the first attempt to use the ACO to disambiguate words senses.

In order to accomplish the task the following modifications were introduced:

1. The set of cities was replaced by a set of identifiers of synsets of all the disambiguated words.
2. The distances between the cities we replaced by the similarity of respective glosses. Prior to the calculation of similarity the words in glosses were replaced by their SynPaths.
3. Selecting one synset identifier of the word w removes automatically all remaining synset identifiers from the list of available identifiers.
4. The original ACS4TSP algorithm favors exploration over exploitation; the ACS4Disamb algorithm the has opposite preferences.
5. The function eval used to evaluate the quality of found solutions has the following form:

$$eval(\{t_1,...t_m\}) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} sim(t_i, t_j)$$

where:

{t1, ..tm} is the set of glosses of the selected systets.

The similarity of glosses was calculated by the tf-idf formula. The 3rd modification is especially important as it reduces dramatically the complexity of the task – the length of the path is equal to the number of disambiguated words and not the total number of all their synsets. The modification number 5 is also important. In the original TSP algorithm we are interested in the shortest possible way whereas in area of sense disambiguation we need to identify nodes related to each other.

5 Study

During the study several sequences of nouns were disambiguated using varying number of ants and iterations. Both the number ants and iterations ware in the range from 2 to 50. The obtained results were compared with the disambiguation prepared by a human expert.

Let $Pr_{AN,IT}(s)$ denote the number of properly selected synsets of the sequence s by a TSP4Disamb using AN ants and IT iterations. The algorithm for the selection of paths ACO uses random number generator and therefore to be able to interpret the results in a statistically meaningful manner each sequence was disambiguated 50 times. In what follows this is called a run. The discussed below average values and standard deviation of $Pr_{AN,IT}(s)$ refer to the results obtained for 5 such test runs.

Three distinct patterns were observed during the study. They are exemplified by the sequences that are shown in the Table 2. The number in brackets after a word is the number of the words' systets. The table contains the sequences accompanied by the number of nodes in the graph. The number is equal to the sum of all words synsets.

The experiment shows that performance was more influenced by the number of ants than the number of iterations. The number of iterations decreased the standard deviation of decreased the standard deviation but of $Pr_{AN,IT}(s)$ but had little effect on its average value. The latter factor was grow with the number of ants. The relationships for S3 are shown on the Fig. 1, other sequences behave in a similar manner.

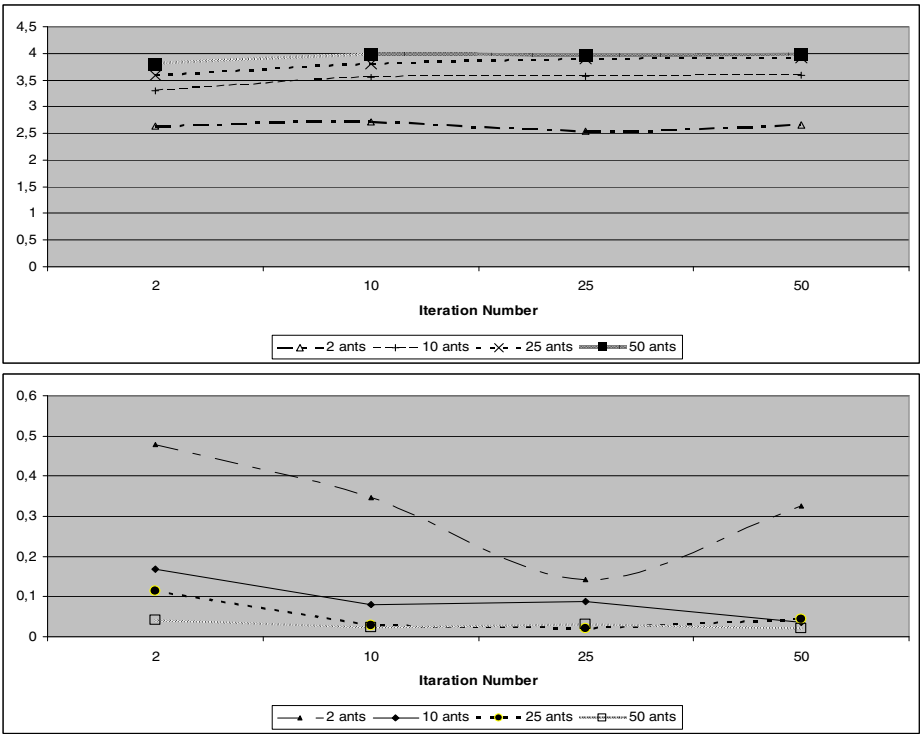


Fig. 1. The average number and standard deviation of the number of proper disambiguation's for the string s3

Using small number of iteration lowers the computational effort and does not effect significantly the disambiguation precision. The next Fig. 2 shows the average value of Pr for the three test sequences. In each cease the iteration number equal to 2 and a number of ants is in the range from 2 to 50. For small numbers of ants the selection of the starting node influences greatly the performance. The node is chosen in a random manner and a selection of a wrong node (sysnet) could not be rectified afterwards. This could be compensated by increasing the number of ants.

The glosses for s1 does not allow for a precise disambiguation no matter what the number of iterations or ant is used. On the other side of the spectrum is it sequence s2 for which an almost flawless disambiguation is achieved for relatively small amount of ants and iterations. The sequence s3 goes in between. The disambiguation precision

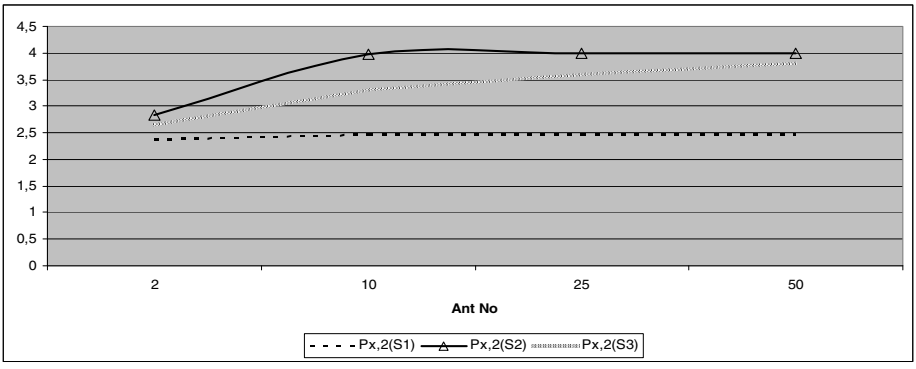


Fig. 2. The average number of the number of proper disambiguation's for the test strings

Table 2. The test sequences for disambiguation

Code	Words	Node number
S1	King(10) knight(2) scepter(2) crown(11)	25
S2	king pawn rook bishop	19
S3	Tree leaf trunk root	19

increases steadily with the increase of ant number. Taking it all into account increasing the number of ant beyond 10 is not justified. The precision for s1 could not be improved anyway and the precision for s2 is not bad. The average value for $P_{20,2}(s3)= 3.3$ which means that on the average 82% of all selections are correct.

6 Conclusions

Provided with a proper set of glosses the ACO4Disamb algorithm disambiguates words using modest number of ants and low number of iterations. Replacing the words by their SynPaths makes it possible to capture the semantic text similarity using efficient tools developed for measuring the syntactic similarity. All this make the approach suitable for on-line applications.

The glosses are short pieces of text and were not written to facilitate automatic disambiguation and in some cases this limits the performance of the proposed approach. A remedy is to extend the glosses. A research project now underway attempts to augment the glosses by sentences retrieved by the Google Search Engine.

References

1. Fellbaum, C., Palmer, M.: Manual and Automatic Semantic Annotation with WordNet. In: Proceedings of NAACL 2001 Workshop (2001)
2. Ng, H.T., Lim, C., Foo, S.: A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation. In: Proceedings of the ACL SIGLEX Workshop: Standardizing Lexical Resources (1999)

3. Xiaohua, Z., Hyoil, H.: Survey of Word Sense Disambiguation Approaches. In: 18th FLAIRS Conference, Clearwater Beach, Florida (May 15-17, 2005)
4. Navigli, R.: ACM Computing Surveys 41(2), article 10 (2009)
5. Fellbaum, C.: WordNet: An electronic Lexical Database. MIT Press, Cambridge (1998)
6. Stevenson, M., Wilks, Y.: The Interaction of Knowledge Sources in Word Sense Disambiguation. Computational Linguistics 27(3), 321–349 (2001)
7. Magnini, B., Cavaglià, G.: Integrating Subject Field Codes into WordNet. In: International Conference on Language Resources and Evaluation LREC 2000 (2000)
8. Hastings, P., et al.: Inferring the Meaning of Verbs from Context. In: Proceedings of the Twentieth Annual Conference of the Cognitive Science Society (1998)
9. Bhattacharya, I., Getoor, L., Bengio, Y.: Unsupervised Sense Disambiguation Using Bilingual Probabilistic Models. In: Proceedings of the Annual Meeting of ACL (2004)
10. Agirre, E., et al.: Combining Supervised and Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. Kluwer Academic Publishers, Dordrecht (1999)
11. Mihalcea, R., Moldovan, D.: An Iterative Approach to Word Sense Disambiguation. In: Proceedings of Flairs 2000, Orlando, FL, pp. 219–223 (2000)
12. Kwong, O.Y.: Word Sense Selection in Texts: An Integrated Model, Doctoral Dissertation, University of Cambridge (2000)
13. Seco, N., Veale, T., Hayes, J.: An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In: Proceedings of the European Conference of Artificial Intelligence (2004)
14. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and Knowledge-based Measures of Text Semantic Similarity. American Association for Artificial Intelligence, 775–780 (2006)
15. Siemiński, A.: Verifying Text Similarity Measures for Two Layered Retrieval. In: Advances in Multimedia and Network Information System Technologies, pp. 245–256 (2010)
16. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)
17. Manning, C., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
18. Dorigo, M.: Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie (1992)
19. Dorigo, M., Stuetzle, T.: Ant Colony Optimization: Overview and Recent Advances, IRIDIA – Technical Report Series (2009) ISSN 1781-3794
20. Chirico, U.: A Java Framework for Ant Colony Systems. In: Ants2004: Forth International Workshop on Ant Colony Optimization and Swarm Intelligence, Brussels (2004)

Further Tests with Click, Block, and Heat Maps Applied to Website Evaluations

Kazimierz Choroś

Institute of Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
kazimierz.choros@pwr.wroc.pl

Abstract. Usability is defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of user. Usability of a website is a very crucial problem. For the Web pages usability is a set of layout, structure, arrangements, typography, and many other properties that makes a website simple and easy to use. We are using many techniques for the evaluation of the websites. One of the new methods is based on the analysis of click, block, and heat maps mainly leading to the detection of these parts of the website which are not used or where the user is intuitively expecting a link to the next part of the site visited. Nowadays, there are many professional software packages to generate these maps, but their deep interpretation seems to be still a problem to consider. The results obtained from the tests performed for two different websites using click, block, and heat map techniques enabled us to draw some interesting conclusions on user behavior. Another problem discussed is to what extent the observations and correlations detected for one website, with the population of dominant visitors relatively well predictable can be applied to another website with unknown population of visitors.

Keywords: usability evaluation, click maps, block maps, heat maps, website pages, Web designing, usability in practice, website visitor population.

1 Introduction

Usability is defined as the extent to which a product can be used by users to achieve specified goals with effectiveness, efficiency, and satisfaction. The satisfaction of the user is a priority also in the Web. It has been observed that there is a great influence of the usability on the user's loyalty to websites that they already visited. The results of the empirical analysis [1] confirm that the trust of the user increases when the user perceived that the system was usable and it stimulates a consequent increase in the degree of website loyalty [2]. A website success is significantly associated with website download delay (speed of access and display rate within the Web site), navigation (organization, arrangement, layout, and sequencing), content (amount and variety of product information), interactivity (customization and interactivity), and responsiveness (feedback options and FAQs).

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 415–424, 2011.

Many feedback methods are based on the observations of user behaviors when visiting websites. The promising approaches for user behavior observations use click, block, and heat map techniques for website evaluations.

The paper is organized as follows. The next section briefly describes the main related works in the area of website usability. Moreover, some related research works are cited. The third section describes the usability evaluation methods, mainly the most popular in practice: testing, inspection, and inquiry. The forth section discusses the clicks, block and heat maps showing the website areas which are most frequently clicked on by the users. It is due to the fact that there is a interesting link for a user or because the user expected a link in this page area. In the sections 5 the experimental results for two tested websites are reported. Then the observations and conclusions that have been formulated after the analysis of click, block, and heat maps, as well as of other data gathered during the tests are presented. The final remarks and the future research work areas are discussed in the last 6th section.

2 Related Works

Various measures for evaluating user satisfaction have been developed [3-6]. Although these measures use different definitions of user satisfaction, all of them are based on an evaluative response provided by a user.

Nowadays, the usability of many specific websites is evaluated. For example, academic library websites [7,8], e-learning systems [9], hypermedia systems [10], commercial websites [11,12], or museum websites [13] have been examined. We expect that the analysis and the evaluation of the usability of websites lead to the improvement of the user satisfaction at website navigation time.

In our former experiments [14] click and heat maps have been applied for analyzing the usability, mainly the layout design of the website of the Institute of Informatics of Wrocław University of Technology. Then the block maps have been defined and their usefulness has been verified.

3 Usability in Practice

Usability measures a product's potential to accomplish the goals of the user. It refers to the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of a user. In the Web world usability is mostly defined as set of layout, structure, arrangements, typography, and some other properties that make website simple and easy to use. Usability reflects the ease with which visitors can employ a particular website in order to achieve their particular goal. Usability is not a single, one-dimensional property of a user interface. Usability is a combination of factors including [15]:

- ease of learning – how fast can a user who has never seen the user interface before learn it sufficiently well to accomplish basic tasks?
- efficiency of use – once an experienced user has learned to use the system, how fast can he or she accomplish tasks?

- memorability – if a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start once more the learning everything?
- error frequency and severity – how often do users make errors while using the system, how serious are these errors, and how do users recover from these errors?
- subjective satisfaction – how much does the user like using the system?

There are many usability evaluation methods leading to the best website designing. Three types are the most popular in practice: testing, inspection, and inquiry [16,17].

In usability testing approach, representative users work on typical tasks using the system and the evaluators use the results to see how the user interface supports the users to do their tasks. Testing methods include the following: coaching method, co-discovery learning, performance measurement, question-asking protocol, remote testing, retrospective testing, shadowing method, teaching method, and thinking aloud protocol.

In usability inspection approach, usability specialists, sometimes software developers, users and other professionals, examine usability-related aspects of a user interface. Commonly used inspection methods are: cognitive walkthroughs, feature inspection, heuristic evaluation, pluralistic walkthrough, and perspective-based inspection.

In inquiry methods, usability evaluators demand information about users' likes, dislikes, needs, and understanding of the system by talking to them, observing them using the system in real work (not for the purpose of usability testing), or letting them answer questions verbally or in written form. Inquiry methods include: field observation, focus groups, interviews, logging actual use, proactive field study, and questionnaires.

4 Click, Block, and Heat Maps

Relatively new techniques of the evaluation of websites are click, block, and heat maps. Although, the idea of heat maps came out of a relatively long history of matrix displays, before and after the computer era [18]. As with many graphical methods, the cluster heat map involved a creative synthesis of different graphical representations devised by a number of statisticians.

Maps show the website areas which are most frequently clicked on by the users. The data may be displayed in two ways: click map and heat map [19]. The map of clicks is represented by crosses, whereas the heat map is represented by circles. Click map shows the position of every click on a page, creating a virtual map of visitor activity. Many clicks on the same point are represented by one cross. The number of crosses on the map is equal to the number of the different points on the website clicking by the user. Whereas, on the heat maps the areas that are clicked most often appear in red (these are so-called hot spot), while the areas clicked least often appear in blue or with no color at all. Therefore, a heat map is a visualization of data which uses color to represent data values in a two-dimensional image. Data are gathered during a normal functioning of the website. Monitoring program works in background and is totally invisible and transparent for the user.

However, it has been observed that the data received could be better understood if some part of the screen would be analyzed together. Therefore, blocks as some parts of the Web page have been defined [14] and then also block maps can be generated. So, the blocks are defined as such a part of area which contains several links, placed every one near another one, forming logical entity.

Click, block, and heat maps are [20]:

- tools which help us to identify the most popular areas of the website,
- tools which allow us to enhance the usefulness of the website,
- knowledge of how most efficiently the content and the advertising components should be organized on the website,
- information on the users' behavior during website visits,
- information on the website presented directly on Web pages,
- help in e-marketing activity,
- verification of changes on the website,
- assessment of advantages and disadvantages of the website.

Naturally, the question remains why people are clicking where they shouldn't. Probably, they expect there a specific link leading to the information they are seeking.

Click maps, block maps, as well as heat maps are nowadays used for the evaluation of the website. These techniques have been used in our experiments to evaluate two websites, that is of the Institute of Informatics (www.ii.pwr.wroc.pl) and of the Association of Polish Alumni of French Universities (www.sage.pl). The websites are different because the users of the first one, i.e. of the Institute of Informatics site can be in majority predictable, they are mainly students and lecturers of the Institute. They form two relatively homogenous groups of users. Lecturers work together, they have quite similar rhythm of their activities, similar software, and similar computer equipments, usually other than in the group of students. Whereas the users of the second website are rather strongly heterogeneous, they live in different places, work in different institutions, and they have totally different reasons to visit this website, there are also many casual users, and practically we cannot even guess who they are.

5 Test Results and Discussion of Results

The main goal of map creations is to decide how we can improve the website. Such an approach has been applied in our former research and practical investigations [14]. This approach brings many benefits. Maps identify usability and information architecture issues on the website. Click density maps as well as block maps are not another statistics or Web analytics tool. They allow us to analyze the user experience and identify usability and information architecture issues.

The maps are useful for many reasons, some of which include:

- knowing exactly how Web viewers use a specific page,
- seeing parts of a page that are most frequently used, are the most attractive,
- seeing which parts of a page are totally unused,
- recognizing such part of the page that are not linked but where the users are clicking on, so, identifying areas where a link is expected,

- understanding typical patterns of use on the site as a whole,
- predicting how people will use the website in the future.

The visualizations in the form of click, block, and heat maps make them ideal for presentation to non-technical audiences. The results of any change made on the page could be easily observed by comparing a map before and after change. We can exactly see where the users are clicking. But we can also see where people are clicking even if they should not. It seems that Web analysts cannot easily detect such situations, but with click, block, or heat maps it immediately becomes obvious.

The data gathered during our tests carried out during three weeks contain not only the position of clicks but also the information which Web browser has been used, on screen resolution, and finally on time interval since the beginning of the visit on a given page to the first click on this page. Hereby, we hoped to detect user behavior.

The users have been not anticipated and did not know that their clicks have been registered. So, their behavior was natural and not influenced by anyone nor anything.

The problem that we see is to what extent the observations and correlations detected for one website, with the population of dominant visitors relatively well predictable can be applied to another website with unknown population of visitors. So, how the conclusions drawn from the test results of one website could be used to analyze or then to improve another website.

5.1 Analysis of the Website of the Institute of Informatics

The click map presented in the Figure 1 shows the distribution of clicks on the main page of the Institute of Informatics. This map enables us to observe which parts of the page are mostly clicked. It is not surprising that these parts are the links to the subpages specifically designed for lecturers or students, participants of programming competition, forum of IT companies, or people from outside looking for contact data.

Fig. 1. Click map (a small cross represent one click) for the main page of the Institute of Informatics website

Can we observe correlations between the Web browser and the preferred parts of the website? The answer is yes. The majority of the Internet Explorer browser users visited the part of the Institute page oriented to the scientific staff. Whereas the majority of the FireFox browser slightly more frequently clicked on the link leading to these parts of the Institute website which mostly present the contents for students (Fig. 2). How to explain it? Students are users preferring the FireFox browser. It confirms the other investigations based on the telephone survey which showed that the proportion of people of the Institute scientific staff using the IE to those using the FireFox browser is like 10:3.

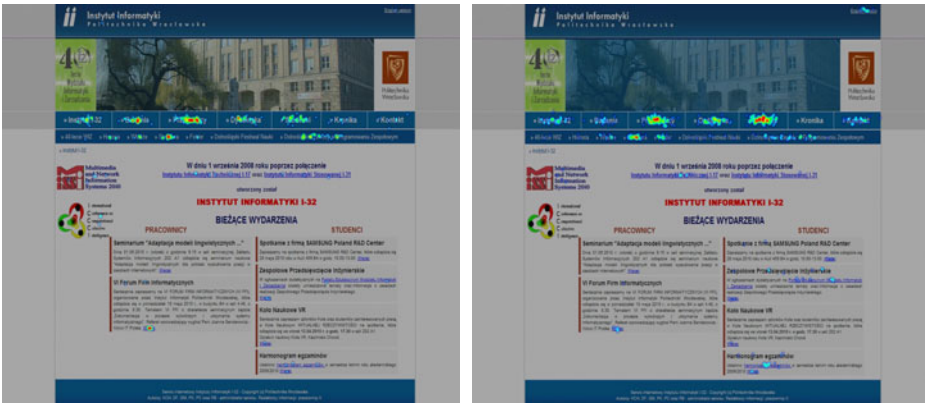


Fig. 2. Heat map for the main page of the website of the Institute of Informatics for the users using the Internet Explore browser (left) and the Firefox browser (right)

The preferences are not justified by the opportunities of browsers. Rather the lecturers tend to the browser that comes preinstalled with theirs computers, and do not seek out other browser alternatives. May be because they are concentrated much more on the content than on the efficiency of a browser. May be because they use a browser much more for browsing textual pages, for example for reading scientific papers and any other materials useful for teaching and scientific investigations. Young people much more frequently use computers also for fun and they much more frequently and easily accept any technical innovations.

Much more frequently student parts of the website have been visited by the users also of the Google Chrome browser. As it was mentioned it can be due to the fact that students are more likely to use a new software, especially if they are free applications.

The next characteristic observed during the tests was a time elapsed since the beginning of the visit on the Institute website to the first click on the page. The majority of visits shorter than one second were continued on the bookmark “scientific staff”. It can be explain in the following way. Users visiting the Institute website know very well this website and immediately choose the most interesting part for them of the website. But it has a quite important consequence. It may suggest that such users do not perceive the changes on the main page of the website. It is important because the main page presents the news of the Institute activities. On the

other hand the changes of the page layout are not desirable because the rapid clicks of users prove that they do it by heart, being used to hitherto scheme.

Longer visits, when the user clicks on the link between one second and five seconds since the beginning of the visit, have been not concentrated on any part of the Web page, so no preference has been observed.

Next, the resolution of the user screen has been registered. In general it is difficult to detect a correlation between the resolution of user screen and the browser used. All the most popular browsers have been used as well as all standard screen resolutions:

- 640 x 480 pikseli – zero clicks
- 800 x 600 – only few clicks
- 1024 x 768 - Firefox, IE, Mozilla, Opera, and Chrome
- 1152 x 864 - Firefox, IE, Opera
- 1280 x 960 - Firefox, IE, Opera
- 1280 x 1024 - Firefox, IE, Mozilla, Opera, Netscape, and Chrome
- 1600 x 1200 - Firefox and IE.

However, an interesting observation concerns the highest screen resolution 1600 x 1200. The majority of clicks on the link to the subpage of the Silesian Competitions in Team Programming, competitions in which high school students are the major part of participants, are made by the FireFox and IE browsers with this highest resolutions. The hypothesis can be that high school students much younger than the scientific staff of the Institute have relatively better eyesight and they prefer the highest resolutions. It is much less probable that the high school students have better and much more modern computer equipment.

5.2 Analysis of the Association Website

The website of the Association of Polish Alumni of French Universities has a totally different task than that of the Institute of Informatics. The Association website is not so dynamic than that of the Institute. News are occasional and the website is rarely updated. People visiting this website are frequently different from those of graduates. The website is rather a kind of showcase informing on Association activities, it is addressed not only for the graduates associated but may be even more for not affiliated and it is also a kind of invitation for those potentially interested to associate.

The most clicking area in the main menu of the Association website are links to *News*, *Authorities*, *Contact data*, and also *Mission* (Fig. 3).

It is very difficult to predict and guess who is visiting the website. It seems that all parts of websites are equally frequently visited. Almost all links are used, only the news are slightly more dominant. How can we explain it? May be the majority of visitors, as it is expected, are people not affiliated to the Association, so they click to almost every link to find exhaustive information on the Associations. Whereas the graduates affiliated to the Association are interested only on news because other parts of the website, well-known and being only occasionally changed are less interesting for them.

Only three browsers have been used by the visitors of the Association website: IE, Firefox, and Chrome. The most popular was the Internet Explore browser. If we try to

Fig. 3. Heat map (left) for the main menu (right) of the Association website

expand the conclusions on the use of browsers by visitors of the Institute website we could suppose that the visitors of the Association website are less frequently students or high school students. The majority of visitors are people with computer behavior similar to that of university lecturers. On the other hand we know that university lecturers are strongly represented in the population of Association members.

The users of the Chrome browser mainly visited this part of the Website with *News*. Whereas *Goals* and *History* were significantly less visited. If we similarly expand the conclusions on the use of browsers by visitors of the Institute website to the Association website the youngest visitors are less interested in the past activities of the Association.

An average click time that is the period of time between a page loading and the visitor clicking is significantly different for the Association website. It is much greater. The majority of visits are of quite long duration about 60 seconds. It can signify that visitors are not very familiar with the Association website or even visit it for the first time. To choose a link on the long list of links in the main menu, some more time is needed. It confirms the conclusions of heat maps of the Association website. On the other hand it may suggest that the main menu of the Association main page is not very friendly for new visitors or that new visitors quite carefully analyze the information on the first main page.

Only two screen resolution have been registered for the Association website: 1024 x 768 and 1280 x 1024. Interestingly, the highest resolution registered for the Institute website 1660 x 1200 is not observed for the Association website. It may suggest that very young people are not interested in the Association website nor in the Association activities. This may be true because alumni associations are not in the interests of people only at the beginning of their careers. So, the conclusion is similar to that on the basis of heat maps.

Amazing is in turn a large number of unproductive clicks that we call “empty clicks”, that is clicks on the parts not only without links but frequently without any textual or graphical elements. Some of them are due to the well-known problem that

users expected for example a link attached to the logo of the Association, but the logo is not linked to anything, not even to the home page.

The problem of empty clicks was examined more carefully. During the observations of people browsing the Web it has been noticed that when they use several windows it happens that to change the window and to make another window active an effective and simple way is to click on the empty place on the page of a desired window. In this case the empty clicks are not caused by the same reasons like clicks on the words or icons (graphical elements) not-linked.

6 Final Conclusion and Further Studies

The click, block, and heat maps are very useful tools for usability evaluation. The maps are visual location of clicks on a given Web page, showing attractive, frequently used zones where the user is expecting and in general finds links as well as unused areas. Maps record the exact x and y coordinates of every click. The map techniques are totally invisible and transparent for the users. Their behavior is natural and not influenced by anyone nor anything.

In the test performed all three maps have been done: click, block, and heat map. The block map should be – beside click maps and heat maps – a third indispensable tool for each webmaster, administrator or those who care for effectiveness of their websites and increasing their websites' usability. A block is an area of a Web page which contains several links, placed next to one another, forming logical entity. Such a solution allow us to aggregate the statistical results obtained for click or heat maps.

Maps are very useful for those who search usability improvements. Nevertheless, at the same time other data have been also gathered: which Web browser has been used, screen resolution, and finally time interval since the beginning of the visit on a given page to the first click on this page.

It was also found that to some extent the observations and correlations detected for one website, with the population of dominant visitors relatively well predictable can be applied to another website with unknown population of visitors.

And then, it may be expected that such a technique could be also applied in a personalization process for websites. We are going to examine if the maps have a personal specificity.

The process of usability evaluation can be automated [21] and can provide quick suggestions for website improvements.

We are also going to study the usefulness of click, block, and heat maps in a dynamically developing Web2.0 environment.

References

1. Flavian, C., Guinaliu, M., Gurrea, R.: The role played by perceived usability, satisfaction and consumer trust on website loyalty. Information & Management 43, 1–14 (2006)
2. Palmer, J.W.: Web site usability, design, and performance metrics. Information Systems Research 13, 151–167 (2002)
3. Muylle, S., Moenaert, R., Despontin, M.: The conceptualization and empirical validation of web site user satisfaction. Information & Management 41, 543–560 (2004)

4. Brinck, T., Gergle, D., Wood, S.D.: Usability for the Web: Designing Web Sites that Work. Morgan Kaufmann, San Francisco (2002)
5. Nielsen, J.: Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis (2000)
6. Nielsen, J., Loranger, H.: Prioritizing Web Usability. New Riders Press, Berkeley (2006)
7. Battleson, B., Booth, A., Weintrop, J.: Usability testing of an academic library Web site: a case study. The Journal of Academic Librarianship 27, 188–198 (2009)
8. Hilyer, L.A.: User-Centred Library Websites: Usability evaluation methods. The Journal of Academic Librarianship 35, 101–102 (2009)
9. Triacca, L., Bolchini, D., Botturi, L., Inversini, A.: MiLE: Systematic Usability Evaluation for E-learning Web Applications. In: Cantoni, L., McLoughlin, C. (eds.) Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 4398–4405 (2004)
10. Matera, M., Costabile, M.F., Garzotto, F., Paolini, P.: SUE Inspection: An effective method for systematic usability evaluation of hypermedia. IEEE Transaction on Systems, Man and Cybernetics 32, 93–103 (2002)
11. Benbunan-Fich, R.: Using protocol analysis to evaluate the usability of a commercial web site. Information and Management 39, 151–163 (2001)
12. Zviran, M., Glezer, C., Avni, I.: User satisfaction from commercial web sites: The effect of design and use. Information & Management 43, 157–178 (2006)
13. Cunliffe, D., Kritou, E., Tudhope, D.: Usability evaluation for museum web sites. Museum Management and Curatorship 19, 229–252 (2001)
14. Choroś, K., Muskała, M.: Block map technique for the usability evaluation of a website. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 743–751. Springer, Heidelberg (2009)
15. Usability.gov. Your Guide for Developing Usable & Useful Web Sites, http://www.usability.gov/basics/
16. Folmer, E., Bosch, J.: Architecting for usability: a survey. Journal of Systems and Software 70, 61–78 (2004)
17. Usability Evaluation, http://www.usabilityhome.com/
18. Wilkinson, L., Friendly, M.: The history of the cluster heat map. The American Statistician 63(2), 179–184 (2009)
19. Brown, S.: Click Survey Analysis & Heatmap. Interpretations and ramifications from a web design point of view (2006), http://modernl.com/article/click-survey-heatmap-analysis
20. Graphical Map of Clicks on a Website, http://stat24.com/en/statistics/clickmap/
21. Feuerstack, S., et al.: Automated usability evaluation during model-based interactive system development. In: Forbrig, P., Paternò, F. (eds.) HCSE/TAMODIA 2008. LNCS, vol. 5247, pp. 134–141. Springer, Heidelberg (2008)

A Research Study on Business-Oriented Quality-Driven Request Service in a B2C Web Site

Grażyna Suchacka[1] and Leszek Borzemski[2]

[1] Chair of Computer Science, Opole University of Technology,
Sosnkowskiego 31, 45-272 Opole, Poland
g.suchacka@po.opole.pl
[2] Institute of Informatics, Wrocław University of Technology,
Janiszewskiego 11/17, 50-370 Wrocław, Poland
leszek.borzemski@pwr.wroc.pl

Abstract. In the paper, a Quality of Web Service (QoWS) method for a Web server system hosting a retail online store is discussed. The method is based on service differentiation between so-called key customers (*KC*), evaluated based on their purchase histories, and other, ordinary customers (*OC*). The second aspect of service differentiation is connected with revenue-related attributes of user sessions at the B2C (*Business-to-Consumer*) Web site. We consider several scheduling policies for the proposed method and implement them in a simulation environment for a business Web server system. Finally, simulation results of the system performance for these policies compared to FIFO (*First In First Out*) scheduling are discussed.

Keywords: QoWS, Web server, e-commerce, B2C, admission control, scheduling, simulation.

1 Introduction

Due to the low quality of Web service (QoWS), manifesting itself mainly in long times Internet users have to wait for receiving the Web content, a number of mechanisms have been proposed for Web components: Web clients, Web servers, and network devices. These mechanisms have all aimed at improving QoWS while applying different approaches and taking various criteria into consideration. They are designed for highly accessed Web servers subject to very changeable and unpredictable Web traffic, which may lead to Web server overloads.

In the paper, a QoWS method for an e-commerce Business-to-Consumer (B2C) Web server system is presented. The method is called KARO (*Key customers And Revenue Oriented admission and scheduling*), and a corresponding request service algorithm is called KARO-Rev (*KARO – Revenue Maximization*). It introduces priority-based admission control and scheduling of requests in the Web server system in order to achieve two business-oriented goals. The first goal is to give users a chance for successful completions of their purchase transactions and thereby generating revenue at a given moment. The second goal is to identify and evaluate

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 425–434, 2011.

key customers of the online store and to offer the highest possible quality of service to them, thus boosting their loyalty and increasing the retailer's revenue in the long run.

The basic scheduling policy is based on strict priority scheduling whereas other policies are based on weighted priority scheduling with various weights assigned to individual priorities. The aim is to choose the most adequate policy with respect to business-oriented control goals and overall system performance. Scheduling policies proposed for KARO method are evaluated through simulation.

The remainder of this paper is organized as follows. Section 2 overviews related work. Section 3 discusses KARO method and KARO-Rev algorithm along with scheduling policies under consideration. In Sect. 4, simulation results for KARO-Rev with various scheduling policies are analyzed. Finally, Sect. 5 concludes the paper.

2 Related Work

There has been a lot of research on Web server architectures being able to prevent server overload and support differentiated QoWS. Control mechanisms have been introduced at the level of an operating system's kernel, at the application level or in a middleware in front of the server. On the one hand, mechanisms which do not prioritize incoming traffic have been proposed, e.g. request admission control based on feedback in order to ensure bounded request response times from a Web server [1] or SRPT (*Shortest Remaining Processing Time*) scheduling to improve the mean request response time at the server [2]. On the other hand, service differentiation based on request prioritization has been proposed [3, 4], including business-oriented request service [5, 6, 7].

3 Business-Oriented Quality-Driven Request Service in a B2C Web Server System

The key idea of KARO method is introducing preferential treatment of the most valued customers of the online store, like in a large part of traditional trade. Based on research on CRM (*Customer Relationship Management*) we decided to define *key customers* (*KC*) of the Web store as users who had made any purchase in the store sometime, as opposed to *ordinary customers* (*OC*). We characterize them using RFM (*Recency, Frequency, Monetary value*) analysis [8], a segmentation method which allows one to describe customers with *RFM scores* based on behavioral data related to customers' past purchases. The behavioral data for each key customer include: the time interval from the customer's last purchase until now, the total number of times the customer has made a purchase, and the total amount of money spent by the customer in the store. A way of applying RFM analysis to key customers' evaluation is described in [9]. The RFM data is stored in key customer database on the Web server and it is periodically recalculated.

All users entering the site start their sessions as ordinary customers and have values of their session ranks equal to 0. However, when a user logs on and is identified as a key customer, the value of his/her session rank is equal to his/her RFM score, read from key customer database.

In order to support differentiated QoWS for different customers, we introduce session priorities and apply the priority-based request admission control and scheduling for key system resources. For the purpose of request service control we use a coarse-grained definition of system resources, encompassing the Web server(s) and the back-end server(s), in contrast to a fine-grained definition, considering low-level server resources, such as CPU, disc, or network interface. The rationale for such approach is an attempt to design a more general and portable solution apart from any specific system architecture. The second advantage of considering the system resources at a high level is a possibility to implement the proposed algorithm as a middleware at the inputs of the Web and back-end servers (or groups of servers). Such middleware mechanisms may be then modularly integrated into the open source Web server software.

A model of request service in a B2C Web server system according to KARO method is depicted in Fig. 1. Two basic components of the system under consideration are a Web server and a back-end server.

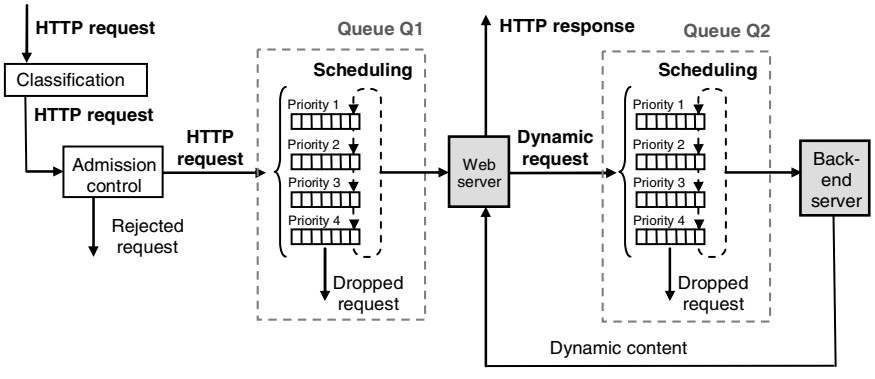


Fig. 1. Model of request service in a B2C Web server system according to KARO method

At any time, a number of users interact with a B2C site, so one can distinguish a number of *user sessions* in the Web server system. Within the session a user visits consecutive Web pages composing the B2C site, i.e. he/she performs different *Web interactions* at the site, such as entering the site, logging in, browsing the site content, searching for products according to specific keywords, reading detailed information on a selected product, adding a product to a shopping cart, registration, or finalizing a purchase transaction. What is important from the system point of view, each single Web interaction involves processing many individual *HTTP requests*, i.e. the first request for an HTML description file and the following requests for Web objects embedded in the page. Requests in the system are processed independently of one another. Moreover, some requests require generating dynamic content, which means that a dynamic request is generated by the Web server and sent to the back-end server.

3.1 Session Priorities

We define four priority levels for a user session at any moment, ranging from 1 to 4. Then, we propose a dynamic session priority scheme, based on a method known in the literature [5].

A priority of a user session (i.e. the priority of a customer interacting with the B2C Web site) changes dynamically with the session progress depending on some session attributes: a type of the current Web interaction being performed in the session, the number of Web pages visited in the session so far, the contents of the customer's shopping cart, and the fact whether the customer is a key one or not. The idea of priority assignment is as follows.

Priority 1, which is the highest possible priority, is reserved for all key customers, as well as for ordinary customers finalizing their purchase transactions. We consider such sessions of paramount significance for the sake of business-oriented control goals and thus give a precedence of service to them.

Priority 2 may be assigned to a session in two cases. First, all customers starting their sessions receive this high priority and retain it for some time. The motivation is to give all users a chance for a successful interaction and to allow key customers to log into the site. Second, this priority is also assigned to all ordinary customers who have some goods in their shopping carts, regardless of the number of Web pages visited in the session. We propose giving them lower priority than to customers who are just about to finalize a purchase because only some of them will finally decide to make a purchase.

Priorities 3 and 4 are assigned to customers who have stayed at the site for a long and very long time, respectively, and have no item in their shopping carts. In reality, such sessions have very low probability of ending up with a purchase, and that is why they are considered the least valuable from a business point of view.

The session priority is verified and updated on arrivals of HTTP requests for new Web pages, i.e. for HTML page description files. All subsequent HTTP requests for Web objects embedded in the page and all corresponding dynamic requests have the same priority.

The session priority and a session rank determine an order of servicing requests of that session in the system in relation to other sessions' requests, and consequently, they affect the resulting quality of service perceived by the customer. Differences in QoWS levels provided by the Web server system are especially evident when the offered load much exceeds the system capacity.

3.2 Admission Control

Admission control consists in rejecting some HTTP requests belonging to low-priority sessions at peak times in order to prevent the system overload and to ensure effective processing of all requests belonging to high-priority sessions. A request rejection means that the request will not be processed at the system.

In order to enforce admission control decisions, we define two thresholds for the system load intensity. When the current system load exceeds the first threshold, the system starts to reject HTTP requests for new Web pages belonging to sessions with priority 4. Above the second threshold, HTTP requests for new Web pages belonging

to sessions both with priority 4 and 3 are rejected. All requests for embedded objects are accepted regardless of the session priority, as well as all requests for new Web pages in sessions with priority 1 and 2.

3.3 Request Scheduling

Accepted HTTP requests are sent to queue Q_1, where they wait for service from the Web server. Similarly, dynamic requests are sent to queue Q_2, where they wait for service from the back-end server. To support differentiated QoWS, we apply priority scheduling to both queues, which enables us to change the order of request execution at the Web and back-end servers. Both Q_1 and Q_2 may be depicted as a multi-level structure of priority queues, where each priority queue is destined for requests with a different priority (which can be seen in Fig. 1). Requests waiting in individual priority queues may be differently ordered and different scheduling policies between the priority queues may be applied. Therefore, various scheduling policies within KARO method may be proposed. Depending on the given performance indexes, one can schedule requests taking different session attributes into consideration.

Ways of request ordering according to KARO-Rev algorithm, which are discussed in this paper, are summarized in Tab. 1. KARO-Rev is oriented towards revenue maximization and the additional performance criterion is related to key customers. Therefore, requests with priority 1 are ordered decreasingly according to shopping cart values to give shorter waiting times to requests from customers with more valuable goods in carts. Requests from the sessions with the same shopping cart values are additionally ordered decreasingly according to session ranks. Requests with priority 2 are also ordered decreasingly according to shopping cart values. However, session ranks are not taken into consideration here because all sessions with priority 2 are from ordinary customers and thus, all the session ranks are equal to zero. Both queues with priority 3 and 4 are ordered simply according to FIFO order.

Table 1. Request scheduling in queue Q_1/Q_2 according to KARO-Rev algorithm

Priority queue	**Request ordering in a priority queue**	**Request scheduling between priority queues**
1	Decreasingly according to shopping cart values; for sessions with equal shopping cart values decreasingly according to session ranks	*Strict priority* or *Weighted priority*
2	Decreasingly according to shopping cart values	
3	FIFO	
4	FIFO	

Between four priorities in a four-layer structure, the following scheduling policies may be applied:

1) *Strict priority*. All higher priority requests are scheduled before lower priority requests. It means that a request with priority 2 may be taken to service only if no request is waiting in the priority queue 1, etc. Such a policy allows us to give better service to premium requests but it may lead to starvation of low priority

requests under a very high system load. A version of the algorithm with the *strict priority* scheduling applied both for Q_1 and Q_2 is referred to as KARO-Rev *S*.

2) *Weighted priority*. A weight is assigned to each priority queue and the priority queues are handled according to their weights. The higher the queue weight, the more requests with the corresponding priority may receive service compared to other priority queues. We assume that if there are not enough requests in higher priority queues, adequately more requests can be serviced from lower priority queues. This version of the algorithm is referred to as KARO-Rev *W* (w_1-w_2-w_3-w_4), where w_1, w_2, w_3, w_4 are weights of priority queues 1, 2, 3 and 4, respectively, both for Q_1 and Q_2.

The way of request scheduling in queues Q_1 and Q_2 determines the order of request service at the Web server and the back-end server, respectively. On the one hand, strict priority scheduling allows one to give an absolute precedence of service to requests with the highest priority. Consequently, it allows one to strongly favor key customers who are logged on, as well as ordinary customers who are making a purchase. On the other hand, weighted priority scheduling allows one to manipulate a degree of QoWS differentiation between requests with different priorities. In the next Section, we present some results of simulation experiments performed in order to assess these dependencies.

4 Simulation Results

To verify the efficiency of KARO method and KARO-Rev algorithm we worked out a simulation model of a B2C Web server system, which is described in [9] in detail. The model includes a session-based, HTTP-level workload model and a queuing network model of a multi-tier Web server system. We implemented the model as a discrete-event simulation tool using C++ and CSIM19 package [10], a toolkit for modeling complex systems. Using the simulator we have done a performance study of a Web server system under FIFO scheduling and under different versions of KARO-Rev algorithm. To compare relative benefits of our approach, exactly the same input workload was generated in all simulation experiments. We assumed a user tolerance of page delay equal to 8 seconds according to the 8-second rule [11]. It means that if a Web page is not completed within 8 seconds at the Web server system, the user will give up the interaction and the corresponding session will be aborted. A share of *KC* sessions in the generated workload was 10%. Each experiment was run for a constant session arrival rate, i.e. for a constant number of new user sessions initiated per minute. The system performance during each experiment was monitored in a 3-hour observation window following a 10-hour preliminary phase (this period refers to the internal simulation time, different from the "real world" time).

First, experiments for FIFO and KARO-Rev *S* have been performed. System performance has been evaluated in terms of the following business-oriented performance metrics:

- revenue achieved per minute through successfully completed buying sessions,
- potential revenue lost per minute, computed based on the total financial value of products which had been in shopping carts of sessions aborted due to low QoWS,
- percentage of successfully completed key customer sessions.

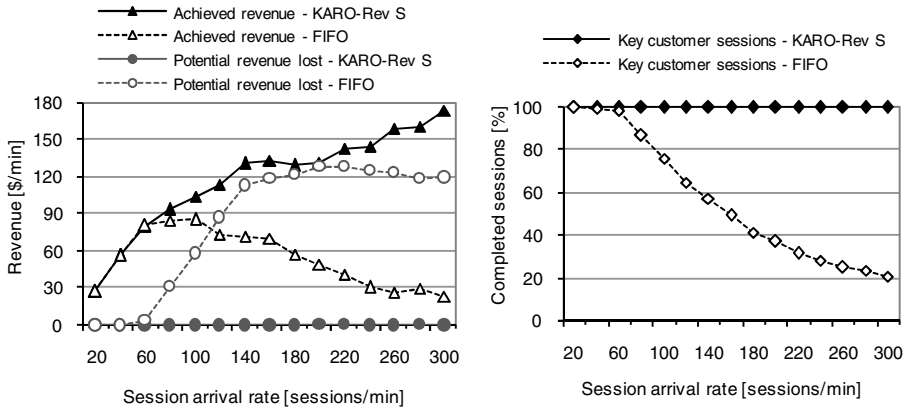


Fig. 2. Revenue achieved and potential revenue lost per minute (left), percentage of successfully completed *KC* sessions (right) for KARO-Rev *S* and FIFO scheduling

Fig. 2-left presents a variation of achieved and lost revenue for all customers as a function of the session arrival rate, ranging from 20 to 300 sessions per minute. Each point at the graph corresponds to a single experiment run for a constant session arrival rate for a given scheduling policy. As it can be seen, the revenue achieved for KARO-Rev *S* increases with the system load virtually throughout the whole load range, as opposed to FIFO, for which it starts decreasing at about 100 sessions generated per minute. For KARO-Rev *S* there is almost no revenue lost - thanks to giving a precedence of service to sessions with goods in carts (which constitute relatively small percentage of all sessions, in fact), it is possible to process such sessions quickly and to prevent potential revenue losses. Moreover, it may seem to be a paradox that potential revenue lost for FIFO decrease above 220 new sessions/minute. The reason is that the heavier the workload, the more sessions aborted at early stages and thus, less users have a chance of adding a product to their shopping carts. Fig. 2-right shows that KARO-Rev *S* has also been able to ensure the very high percentage of successfully completed key customer sessions even for the maximum system load.

The cost of these improvements has been a little bit lower QoWS for ordinary customer sessions in most cases. That is why we decided to run experiments for KARO-Rev algorithm with *weighted priority* scheduling and evaluate impact of different priority queue weights on the system performance. Parameters of KARO-Rev *W* policies are presented in Tab. 2.

As expected, changing the scheduling policy from *Strict priority* to *Weighted priority* caused deterioration of QoWS for requests with the highest priority (i.e. the priority 1); in this group there are requests from key customers. Simultaneously, it led to QoWS improvements for ordinary customer requests with priorities 2, 3 and 4. As it can be seen in Tab. 3, in the case of KARO-Rev *W*, the bigger the weight of the priority queue 1, the bigger the number of successfully completed key customer sessions in the observation window. This number is the highest for KARO-Rev *S*. However, the differences are not dramatic – they are of the order of a few percent.

Table 2. Parameters of KARO-Rev scheduling policies under consideration

Scheduling policy	Scheduling	Weights of priority queues			
		1	**2**	**3**	**4**
KARO-Rev *W* (4-3-2-1)	*Weighted priority*	4	3	2	1
KARO-Rev *W* (7-5-3-1)	*Weighted priority*	7	5	3	1
KARO-Rev *W* (10-7-4-1)	*Weighted priority*	10	7	4	1
KARO-Rev *S*	*Strict priority*	–			

Table 3. Number of completed sessions for KARO-Rev scheduling policies for session arrival rate of 300 sessions/min

Scheduling policy	Number of completed sessions		
	KC	***OC***	**All**
KARO-Rev *W* (4-3-2-1)	5465	8343	13808
KARO-Rev *W* (7-5-3-1)	5468	8477	13945
KARO-Rev *W* (10-7-4-1)	5488	8523	14011
KARO-Rev *S*	5526	8384	13910

Fig. 3-left presents the percentage of successfully completed *KC* sessions as a function of the session arrival rate for KARO-Rev scheduling policies. The best values of this metric have been achieved for KARO-Rev *S*. However, we have been surprised to observe that among the *Weighted priority* policies the worst results have been achieved for the case with the most differentiated weights, i.e. for KARO-Rev *W* (10-7-4-1). Such results for this case have been caused by the very high number of *OC* requests competing with *KC* ones. A bigger number of completed *OC* requests has lead to a bigger number of completed *OC* sessions, which can be seen in Tab. 3. Generally, for more differentiated priority queue weights the load of the Web server system has been more intensive than for less differentiated weights. With relatively high weights of priority queues 2 and 3, the system has not been able to successfully process all requests with priority 1. Most probably, better QoWS for *OC* requests with priority 2 and 3 has resulted in a bigger number of *OC* sessions with goods in carts. Consequently, more *OC* requests with priority 2 have been queued before requests from unlogged key customers, which then have been more likely to timeout.

The revenue achieved per minute has turned out to be very similar for all four scheduling policies and none of them has gained a clear advantage over the others. However, after analyzing the percentage of achieved revenue (which takes the total potential revenue into account) some differences among the policies became visible (Fig. 3-right). *Strict priority* policy has been the most effective in preventing losses of potential revenue. The curves for *Weighted scheduling* policies in Fig. 3-right reveal

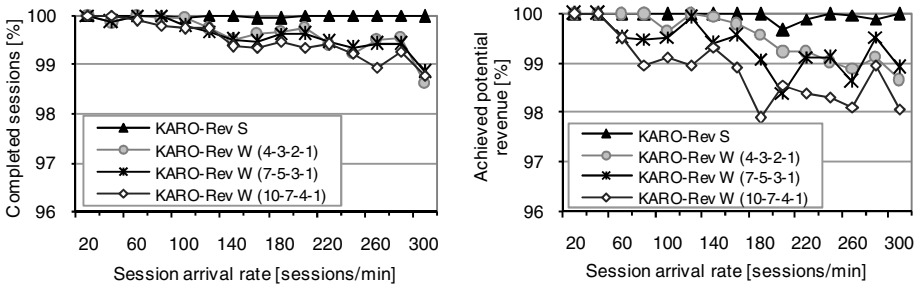


Fig. 3. Percentage of successfully completed *KC* sessions (left), percentage of achieved potential revenue (right) for KARO-Rev *S* and KARO-Rev *W* scheduling

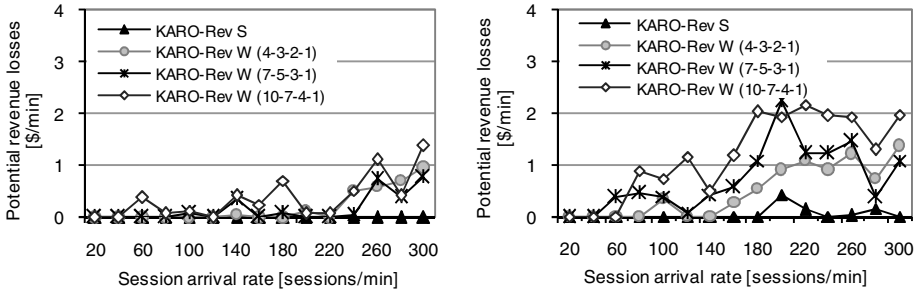


Fig. 4. Potential revenue lost per minute from key customers (left) and from ordinary customers (right) for KARO-Rev *S* and KARO-Rev *W*

a similar tendency as in Fig. 3-left, i.e. more differentiated priority queue weights gave worse results. Our interpretation for this result is that a bigger differentiation between the weights acts as an advantage of *OC* sessions, especially when the spare capacity, reserved for higher priority requests, can be assigned to lower priority ones. In such a case, ordinary customers gain a bigger chance of continuing their sessions and adding products to the carts, which implies higher potential revenue. However, the system with *Weighted priority* scheduling is not able to successfully process all the sessions with goods in carts. Consequently, more potential revenue is lost and the resultant percentage of achieved revenue is lower.

Fig. 4 presents potential revenue lost per minute as a result of aborted *KC* and *OC* sessions. One can observe that a little bigger potential revenue is lost due to aborted *OC* sessions than due to *KC* ones virtually throughout the whole load range. However, given that *KC* sessions have presented only 10% of all sessions, the amount of revenue losses per *KC* session is actually higher than the corresponding value per *OC* session. As it can be seen, *Strict priority* policy is the only one which is able to entirely prevent revenue losses from *KC* sessions, as well as able to ensure the lowest revenue losses from *OC* sessions.

5 Concluding Remarks

In the paper, a few scheduling policies have been considered in the context of the proposed QoWS method, which aims at improving the B2C Web server system performance in terms of business-oriented goals. Using our simulation environment, operation of the system under these policies has been evaluated.

Experiments have shown that compared to the basic version of KARO-Rev algorithm, which applies *Strict priority* scheduling between requests with different priorities, none of the *Weighted priority* scheduling policies have brought in significant improvements in terms of business-oriented performance metrics, regardless of priority queue weights. Moreover, all these policies caused deterioration of QoWS for key customers, only slightly improving QoWS of ordinary customers. Even though they allowed ordinary customers to proceed more effectively with their sessions, the system was not able to turn the bigger amount of potential revenue into the actual revenue of the online retailer. In the meantime, more key customers suffered from long response times and more *KC* sessions were aborted finally.

References

1. Qin, W., Wang, Q.: An LPV Approximation for Admission Control of an Internet Web Server: Identification and Control. Control Eng. Pract. 15(12), 1457–1467 (2007)
2. Schroeder, B., Harchol-Balter, M.: Web Servers under Overload: How Scheduling Can Help. ACM TOIT 6(1), 20–52 (2006)
3. Wei, J., Xu, C.-Z.: Design and Implementation of a Feedback Controller for Slowdown Differentiation on Internet Servers. In: WWW, pp. 1006–1007 (2005)
4. Huang, C.-J., Cheng, C.-L., Chuang, Y.-T., Jang, Y.-S.R.: Admission Control Schemes for Proportional Differentiated Services Enabled Internet Servers Using Machine Learning Techniques. Expert Syst. Appl. 31(3), 458–471 (2006)
5. Menascé, D.A., Almeida, V.A.F., Fonseca, R., Mendes, M.A.: Business-Oriented Resource Management Policies for E-Commerce Servers. Perform. Eval. 42(2-3), 223–239 (2000)
6. Shaaban, Y.A., Hillston, J.: Cost-based Admission Control for Internet Commerce QoS Enhancement. Electronic Commerce Research and Applications 8, 142–159 (2009)
7. Yue, C., Wang, H.: Profit-aware Admission Control for Overload Protection in E-commerce Web Sites. In: 15th IEEE IWQoS, pp. 188–193 (2007)
8. Miglautsch, J.R.: Thoughts on RFM Scoring. The Journal of Database Marketing 8(1), 67–72 (2000)
9. Borzemski, L., Suchacka, G.: Business-Oriented Admission Control and Request Scheduling for e-Commerce Web Sites. Cybernet. Syst. 41(8), 592–609 (2010)
10. Schwetman, H.: CSIM19: A Powerful Tool for Building System Models. In: WSC, pp. 250–255. IEEE, Los Alamitos (2001)
11. 8 Seconds to Capture Attention. Silverpop's Landing Page Report (June 2007), `http://www.silverpop.com/practices/studies/landing_page` (access date: March 15, 2008)

Collective Intelligence Approach to Measuring Invisible Hand of the Market

Pawel Skrzynski[1], Tadeusz Szuba[1], and Stanisław Szydło[2]

[1] AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, IT and Electronics, al. Mickiewicza 30, 30-059 Krakow, Poland
pawel.skrzynski@agh.edu.pl, geszuba@cyf-kr.edu.pl
[2] AGH University of Science and Technology, Faculty of Management
sszydlo@zarz.agh.edu.pl

Abstract. The paper presents an invisible hand process simulation using a multi agent approach and its measure with a IQS – collective intelligence factor used to calculate the intelligence of a given social structure. Since it is widely assumed that in a free market, market participants create such a structure the authors claim that it is possible to calculate the effect of the invisible hand process by defining the IQS of the market. The agents behavior and interaction is modeled by adapting classical models from a theory of microeconomics which describes the ways goods are exchanged among agents. The paper also introduces in details the concept of a multi-agent market simulator which was used in experiments.

Keywords: collective intelligence, IQS, multi-agent systems, theory of exchange, invisible hand process, theory of utility, microeconomics.

1 Introduction

Adam Smith introduced the concept of the Invisible Hand process by describing that as a process in which each individual (market participant) labors to render the annual revenue of the society as great as he can [7]. Smith proved that the individuals generally, indeed neither intend to promote the public interest, nor know how much they are promoting it. His theory explains and uncovers structures and processes which people unintentionally and without recognition use as if they were guided by an Invisible Hand. Nowadays, something much more general is meant by the expression Invisible Hand. An Invisible Hand process is one in which the outcome to be explained is produced in a decentralized way, with no explicit agreements between the acting agents [4]. The second essential component is that the process is non-intentional. The agents' aims are neither coordinated nor identical with the actual outcome, which is a by-product of those aims. The process should work even without the agents having any knowledge of it. This is why the process is called invisible. The system in which the Invisible Hand is most often assumed to work is the Free Market [4].

Such a definition is surprisingly compliant with the concepts underlying the collective intelligence computational model [9] which lead authors to designing

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 435–444, 2011.

market simulation system based on molecular model of computation [5]. Earlier works proved that it has been reasonable to apply formalism, which is well described to analyze and study market processes in general and Invisible Hand in particular.

Despite the fact that invisible hand is a term that has become classical and belongs to paradigms of economy no-one has yet described a way to calculate its effects. Both the classical definition given by A. Smith [7] and general definition mentioned earlier emphasize the same thing: the genesis of the process is market participant behavior which is aimed on promoting its own interest. If we consider micro-economic models that describe market participant behavior such as theory of preferences or theory of market exchange we might notice that these models are a hundred per cent compliant with the origins of the invisible hand process.

Economists use the term utility to measure relative satisfaction. Given this measure, one may speak meaningfully of increasing or decreasing utility, and thereby explain economic behavior in terms of attempts to increase one's utility. Utility is usually modeled to be affected by the consumption of various goods and in most models is described by mathematical functions which calculates the utility depending on the level of consumption of each good. While exchanging goods in a free market, market agents choose goods that offer maximum possible utility levels. In terms of collective intelligence each exchange between agents is an act of communication which is well defined by the following general equilibrium principle: the "Pareto efficiency principle" [10]. Given an initial allocation of goods among a set of (market participants), a change to a different allocation that makes at least one individual better (which is equal to the increasing utility level of possessed goods) off without making any other individual worse off (decreasing utility level) is called a Pareto improvement. An allocation is defined as "Pareto efficient" or "Pareto optimal" when no further Pareto improvements can be made.

The question is how to measure the invisible hand process effects? As we mentioned the fundamental element of the invisible hand process is agent behavior which is modeled at a micro level by adapting microeconomic models. Economics can be treated as a science which provides the answer for the basic question: what is best way to use the resources provided by the environment and distribute them among economic agents that socially participate in the production, exchange, distribution, and consumption of goods and services in the environment. On a macro level it introduces GDP (Gross Domestic Product) factor which refers to the market value of all goods and services produced within a country in a given period. The authors claim that invisible hand processes which are derived from microeconomic models affect the GDP on a macro level, hence it is possible to calculate those effects by adapting the GDP. We can even go further in our observation by treating market exchanges between two agents as a communication act which creates social structure among market participants and then provide a way to measure its "intelligence" by adapting collective intelligence IQS. The IQS definition is general so we might define IQS_{market} based on GDP. The whole concept is discussed in further sections of this paper.

2 Market Simulation System – Concept and Architecture

Formal model of market simulator system uses multi-agent approach[1] – M-Agent architecture [2][3]. Such an approach seems to be natural as this architecture focuses on modeling whole system from the single agent point of view - each agent observes the surrounding world and builds its model which is used as input for making decisions about choosing the best strategy to achieve its goals. Hence, adapting models taken from microeconomics to multi-agent market simulator should be relatively easy. Furthermore, such an approach is compliant with the invisible hand paradigm.

The essential element of each agent architecture is the environment which provides agents with a space to live by providing resources which are used by the agents.

The formal definition of a Multi-Agent Market Simulation System (MAMSS) is as follows:

$$MAMSS =< agents, env, t > \tag{1}$$

where:

agents - set of agents (market participants)
env - environment
t - time

Environment provides agents with the resources which might be gathered by agents to become goods - the subject of exchange between agents. Each resource is countable. Moreover, the environment provides agents with some information and introduces space within which agents exists:

$$env =< Res, Inf, space > \tag{2}$$

where:

Res - set of resources
Inf - set of information, in the MAMSS system (in the experiments presented in the paper the only information is: R_i interaction radius)
space - two dimensional space

Adding space to the above definition has some effects:

- system has topology: possible location of agents and resources is provided,
- defining location attribute for each agent
- defining location attribute for each resource
- adding interaction radius which defines the "market visibility distance" for each agent - agent is aware of existence of other agents which are within its reach in the term of R_i radius (in the given metrics).

[1] Extensive discussion on multi-agents approaches to build market simulation systems might be found in [5][6].

Space in the MAMSS system is modeled as a two dimensional mesh and the formal definition is given below:

$$space =< Pl, Tr, Aloc, Rloc > \tag{3}$$

where:

Pl - set of possible locations
Tr - relation defining possibility of agent migration: $Tr \subset Pl \bigotimes Pl$
Aloc - injective function defining agents' location (agent might be at only one place at a given time)
Rloc - injective function defining resources' location

Because, as it was said above, space is a two dimensional mesh:

$$Pl \equiv \{(x, y) : x, y \in \mathbb{N}\} \tag{4}$$

relation Tr might be defined as follow:

$$Tr \equiv \{((x_1, y_1), (x_2, y_2)) :\mid x_1 - x_2 \mid \leq 1, \mid y_1 - y_2 \mid \leq 1\} \tag{5}$$

where:

K - finite subset of $\mathbb{N}$ *- natural numbers from 0 to k*
Pl - set of possible locations $pl = (x_i, x_j) \subset Pl$

Each node might be described by resources (countable) that are available within it. It is further assumed that information about the resources is available only for agents which are located in that node.

The last important thing to be defined in the space is metrics since we define interaction radius as visibility range for the agents. We can define the distance between two points within mesh as:

$$d\left(p_{i_1 k_1}, p_{i_2 j_2}\right) = min\left(\mid i_1 - i_2 \mid, \mid j_1 - j_2 \mid\right) \tag{6}$$

2.1 Market Agent Modeling

The agent in the MAMSS system represents market participant and is described by a set of actions that is capable of performing and some attributes and algorithms which we call internal structure. The internal structure of an agent is responsible for the decision process which is responsible for choosing the action to be performed and the way the action is performed. The internal structure is defined as a set of profiles. We can distinguish between physical profiles which are related to the assets possessed, intellectual profile which describe agent preferences and all the information which represent an agent's knowledge about the surrounding world (ex. information about preferences of other agents that are visible to the agent) and the migration profile as agent can change its location within space (mobility).

Agent might be summarized as follows (the more complete description was given in previous papers [5] [6]):

- preferences - utility function, mathematical formula describing agent preferences about the goods which are available,
- utility level - utility level is increased by the consumption of goods, if the utility level drops to zero a market participant is removed form the market (in that sense it might be treated as a generalization of the term "energy". Moreover, utility level drops in each turn by a certain value which is defined by the environment which might be interpreted as "life costs" [1])
- assets - goods possessed - consumed by an agent and exchanged between agents,
- consumption strategy - defining what and how much to consume,
- production function - mathematical formula describing what agent might produce and what are the costs (measured in utility which is the authors' assumption and further explains the energy relation of the utility attribute),
- migration strategy - mathematical formula which allows for the computation of the next location for the agent,
- information - set of information about the surrounding world including the preferences of other agents that are visible to the agent (located within reach in the sense of metric and R_i) and transaction history,
- transaction algorithm - defining the way an exchange of goods between agents is performed.

In economics it is widely assumed that the best way to express human preferences is by the Cobb-Douglas functional formula:

$$u\left(x_1, x_2, ..., x_n\right) = x_1^{c_1} * ... * x_n^{c_n} \tag{7}$$

where:

u - *utility function*
x_i - *the amount of good* i
$c_1, ..., c_n \in \mathbb{R}$ *and* $c_1, ..., c_n > 0$

2.2 Agents' Interaction – Transaction Modeling

As mentioned in the Introduction the main communication mechanism among market participants is goods exchange. The general rule that agents follow is achieving Pareto improvement – each agent tries to improve its situation which is equal to increasing utility level, in the same time there is no agent that worsen its situation after the transaction.

Given denotation as below:

n - *no of goods defined by the environment*
$\{ag_1, ag_2, ..., ag_m\}$ - *set of agents engaged in transaction*
$\{u_1, u_2, ..., u_m\}$ - *set of utility functions of the agents* (u_i *is related to* ag_i)
$G_0 = \left\{G_1^0, G_2^0, ..., G_m^0\right\}$ - *initial allocation of goods among agents where*
$G_i^0 = \left\{g_{i1}^0, ..., g_{in}^0\right\}$ *and* g_{ik}^0 *is the amount of good* k *possessed by agent* i *before transaction*

$G_F = \{G_1^F, G_2^F, ..., G_m^F\}$ - *final (after transactions) allocation of goods among agents where* $G_i^0 = \{g_{i1}^F, ..., g_{in}^F\}$ *and* g_{ik}^F *is the amount of good k possessed by agent i after transaction*

We can write the formal conditions for transaction which to be Pareto efficient. First, the allocation after a transaction must not be worse than before the transaction:

$$\sum_{i=1}^{m} u_i(g_{i1}^0, ..., g_{in}^0) \leq \sum_{i=1}^{m} u_i(g_{i1}^F, ..., g_{in}^F) \tag{8}$$

Second, there is no better allocation G_K than G_F:

$$!\exists\, G_K : \sum_{i=1}^{m} u_i(g_{i1}^K, ..., g_{in}^K) > \sum_{i=1}^{m} u_i(g_{i1}^F, ..., g_{in}^F) \tag{9}$$

To simplify the transaction implementation some further assumptions have been made: the number of agents involved in the transaction is limited to two. The whole procedure is given below:

```
1. procedure Transactions
2.    foreach (agent)
3.       N := findNeighbours(interactionRadius)
4.          foreach (neighbor: N)
5.             transaction(agent, neighbour);
6.          end foreach;
7.    end foreach;
8. end procedure
```

Procedure `transaction` which is called in line 5 has been implemented using a simple genetic algorithm according to the conditions given above.

2.3 Agent Migrations

Due to the fact that the utility level is usually increased by the transaction (as the utility level of goods possessed by an agent is increased) migration should be modeled in the way that allows for market participants to search for a "better market". If no transaction has been performed the agent makes a random move to the neighboring field, and in that case the new position is calculated as follows – given the denotations below:

p_1, p_2 - *agents' performing transaction location,*
p_1', p_2' - *new locations*
$d(p_1, p_2)$ - *distance between* p_1, p_2
u_1 - *utility level of goods possessed by agent 1 before transaction*
u_2 - *utility level of goods possessed by agent 2 before transaction*
u_1' - *utility level of goods possessed by agent 1 after transaction*
u_2' - *utility level of goods possessed by agent 2 after transaction*
$\| \overrightarrow{p_1, p_2} \|$ - *unit vector for* p_1 *and* p_2

the new position is:

$$p_1' = \begin{cases} u_1' - u_1 < 2u_1 & p_1 + d(p_1, p_2) * ||\overrightarrow{p1, p2}|| * (u_1' - u_1)/2u_1 \\ u_1' - u_1 \geq 2u_1 & p2 \end{cases} \tag{10}$$

3 Defining the IQS of the Market

The definition of IQS (IQ Social) given by Szuba [8] which allows for the measurement of collective intelligence for any given social structure is very general thus applying it to a specific problem requires applying adaptive procedures. To define IQS for the market we have to refer to the GDP definition which explains this term as the market value of all goods and services produced within a country in a given period. It is computed by adding the value of all goods produced and deducting the value of goods that were used in the production process. However, in a real economy it is difficult to calculate GDP. According to this formula, in a MAMSS simulation system it is relatively easy as the only way production is present in the model is gathering resources from the environment by the agents (the other way, which is not used in experiments, is the production process in which agent uses goods possessed to produce another good). Hence, we might write that:

$$GDP_{MAMSS} = \sum_{a_i \in Agents} \left[\sum_{j=1}^{k} (p_{a_i j} - \sum_{n=1}^{m} g_{a_i jn}) + \sum_{p=1}^{r} e_{a_i p} \right] \tag{11}$$

where:

$p_{a_i j}$ – *the amount of good j produced by agent i*
$p_{e_i p}$ – *the amount of resource p gathered by agent i from the environment*
$g_{a_i jn}$ – *the amount of good n used in the production of good j by by agent i*

Having defined GDP we might define IQS for the market as:

$$IQS_{market} = GDP_I - GDP_{Ref} \tag{12}$$

where:

GDP_I – *GDP in simulation with interactions enabled (transactions)*
GDP_{Ref} – *GDP in simulation with interactions disabled (transactions not allowed, each agent is producing goods to consume by itself)*

The interpretation of this formula is that it measures profits from interactions which is equal to GDP increase comparing to reference GDP achieved in the simulation with transactions disabled. It would be useful if the additional measure based on the IQS would be defined which refers to a relative GDP increase caused by agents' interactions (collective intelligence). Such a factor might be interpreted as a way to measure market efficiency resulting in social mechanisms. Hence, we define ΔIQS as:

$$\Delta IQS = IQS_{market} = \frac{GDP_I - GDP_{Ref}}{GDP_{Ref}} \tag{13}$$

4 Experiments

Based on the provided fundamentals of the MAMSS system we can define an experiment which is supposed to measure the collective intelligence of the market using the metrics that have been defined in previous sections. Using the prototype MAMSS implementation we define the environment setting by the following attributes: space size: 100 x 100, goods/resources available: G1, G2, G3, G4, resources distribution: random – centers of 20 "islands" defined arbitrary (approximate size 15 x 15) and all goods are distributed randomly within island. Each island has 25 fields providing resources. The initial amount of a resource in each field was arbitrary set at 10 and each turn produces a further 10 quantities of each resource.

"Island distribution" was assumed to be arbitrary to make migration profile sense. In such an environment 200 agents are distributed randomly. There are two classes of agents (100 agents each):

1. Agents of class A – preferences defined by some variation[2] of Cobb-Douglas functional form for two goods: G1 and G2, production function has two parameters as well: G1 and G3 (only one good which agent is able to produce may be consumed by itself).
2. Agents of class B – preferences defined in the same manner but for goods: G3 and G4, production function has two parameters as well: G2 and G4 (again only one good which agent is able to produce may be consumed by itself).

Utility function defining class A agents preferences (analogical formula with different parameters is defined for class B):

$$u(g_1, g_2) = \frac{g_1 g_2}{2} + g_1 + g_2 \tag{14}$$

where $g1$, g_2 denotes the amount of goods: G_1, G_2. Production function for agents of both classes was defined as below:

$$y = \begin{cases} \frac{2}{5}x & x \leq 10 \\ 2 * log_{10}10x & x > 10 \end{cases} \tag{15}$$

where x denotes the amount of goods – G_1 and G_3 for agents of class A, G_2 and G_4 for agents of class B. Such a definition allowing for a "scale effect".

Initial utility level for both classes was set at 10 and utility loss per turn set at $0,5$. For each interaction radius R_i 20 simulations were made, 100 iterations each. Table 1 presents the results – "Ref." is reference simulation where interactions (transactions between agents) were disabled. The purpose of running simulations with different interaction radius was to verify the impact of the interaction strength (bigger R_i translates to stronger communication – more transactions are made) on the economy.

[2] It is the simplest form of the Cobb-Douglas functional form with a $g_1 + g_2$ component added.

Table 1. Results – average values *Avg.* and standard deviation *SD* after 100 turns

Market factor	Ref.	$R_i = 1$	$R_i = 2$	$R_i = 3$
	Avg. | SD	Avg. | SD	Avg. | SD	Avg. | SD
GDP	158,161|1,536	272,976|1,776	289,305|1,738	294,326|1,812
Agents population	148.95|7.37	146.85|2.37	156.80|1.41	159.23|1.52
Average utility level	798|33	3,596|123	3,612|132	3,734|145
Transactions value	*NA*	101,123|1,212	146,591|2,134	167,234|1,923
GDP per capita	1,054|98	1,869|112	1,852|134	1,849|141
	$R_i = 4$	$R_i = 5$	$R_i = 7$	$R_i = 10$
	Avg. | SD	Avg. | SD	Avg. | SD	Avg. | SD
GDP	301,143|1,798	307,875|1,524	312,124|1,523	318,300 |1,564
Agents population	163.12|1.56	167.95|1.22	169.61|1.26	173.90|1,31
Average utility level	3,784|156	3,812|182	3,841|181	4,012|189
Transactions value	179,415|2,156	189,315|2,162	195,312|2,321	201,321|2,658
GDP per capita	1,846|131	1,838|141	1,846|143	1,827|141

It is clear that significant improvement appears for $R_i = 1$ to $R_i = 5$ – from then on the growth is stable but slow. The interesting issue is the analysis of the GDP during simulation as function of time – the GDP produced within one turn becomes stable after 15-20 turns and from that point on growth becomes stable but very slow.

The results show that it was possible to configure all the required parameters of the system to achieve stability. The final step to do is to compute IQS_{market} and ΔIQS_{market} which is shown in table 2.

Table 2. IQS_{market} and ΔIQS_{market} for chosen R_i values

	$R_i = 1$	$R_i = 2$	$R_i = 3$	$R_i = 4$	$R_i = 5$	$R_i = 7$	$R_i = 10$
ΔIQS	114,815	131,144	136,165	142,982	149,714	153,963	160,139
ΔIQS_{market}	0.73	0.83	0.86	0.90	0.95	0.97	1.01

IQS roots are in IQ factor (taken from psychology) which is used to measure human intelligence. Comparing those results to humans' IQ – 73% increase means moving from below average human being to ultimate genius.

5 Conclusions

This paper describes a model for a multi-agent market simulation system *MAMSS* which allows for the analysis of the invisible hand process because the market participant behavior was modeled by adapting microeconomic models to achieve compliance with the invisible hand paradigm. The effect of the process is measured by

the collective intelligence IQS metrics by adapting the general definition of the IQS to market specificity – this is achieved by using GDP general definition to measure the outcome of the economy. Such a definition, in the authors' opinion, might be treated as a bridge between microeconomics (agent behavior models on micro level) and macroeconomics (the GDP factor measuring the way microeconomic models affect the economy on macro level).

Moreover, the MAMSS model described in this paper might be easily extended by adapting other economic models – ex. adapting "money" to the model which might be treated as an interaction mechanism which further improves the collective intelligence of the market. One of the main advantages of this model is that it allows for market efficiency research based on different factors for example different consumption/production/migration strategies etc.

References

1. Blaug, M.: Economic Theory in Retrospect, 5th edn. Cambridge University Press, Cambridge (1996)
2. Cetnarowicz, K., Gruer, P., Hilaire, V., Koukam, A.: A formal specification of M-agent architecture. In: Dunin-Keplicz, B., Nawarecki, E. (eds.) CEEMAS 2001. LNCS (LNAI), vol. 2296, pp. 62–72. Springer, Heidelberg (2002)
3. Cetnarowicz, K.: M-Agent Architecture Based Method of Development of Multiagent Systems. In: Proceedings of International Conference on Physics Computing (1996)
4. Joyce, H.: Adam Smith and the invisible hand, Millennium Mathematics Project. Plus Magazine 3 (2001)
5. Skrzynski, P., Szuba, T.: Invisible hand process simulation based on collective intelligence computational model. In: Bolc, L. (ed.) Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science, pp. 541–550. Warsaw Academic Publishing House EXIT, Warsaw (2009) ISBN 9788360434598
6. Skrzynski, P., Szuba, T., Szydlo, S.: Agent based market simulation using collective intelligence computational model. In: Lebkowski, P. (ed.) Production engineering in making, pp. 127–136. AGH University of Science and Technology Press, Krakow (2010) ISBN: 9788374643702
7. Smith, A.: An Inquiry into the Nature and Causes of the Wealth of Nations. University Of Chicago Press, Chicago (1977) ISBN 0226763749
8. Szuba, T.: A Formal Definition of the Phenomenon of Collective Intelligence and its IQ Measure. Future Generation Computing Journal 17, 489–500 (2001)
9. Szuba, T.: Computational Collective Intelligence, New York. Wiley Series on Parallel and Distributed Computing (2001)
10. Varian, H.R.: Intermediate Microeconomics. A modern approach. W. W. Norton & Company, New York (1987)

Collective Intelligence of Genetic Programming for Macroeconomic Forecasting

Jerzy Duda[1] and Stanisław Szydło[2]

[1] Department of Applied Computer Science, AGH University
[2] Department of Economy, Finance and Environmental Management
AGH University of Science and Technology, Faculty of Management, ul. Gramatyka 10,
30-067 Kraków, Poland
{jduda,sszydlo}@zarz.agh.edu.pl

Abstract. A collective approach to the problem of developing forecasts for macroeconomic indicators is presented in the paper. The main advantage of genetic programming over artificial neural networks is that it generates human readable mathematical expressions that can be interpreted by a decision-maker. Gene expression programming used in the paper is an example of collective adaptive system, but we propose to use a collective intelligence to develop not only one forecasting model, but a set of models, from which the most suitable one can be chosen automatically or manually by the decision-maker.

Keywords: collective intelligence, genetic programming, forecasting models.

1 Introduction

One of the basic problems policy makers in a company have to face is uncertainty about the macroeconomic environment. This is particularly true for strategic decisions such as sales volume planning, investment planning or choosing the right source of business financing. On the other hand, every major disruption in the business environment creates an opportunity or a threat to the company. Therefore, it is important to gain the knowledge about the anticipated changes in the macroeconomic environment.

One possibility is to follow the forecasts made by experts (e.g. published by the European Commission), but these forecasts apply only to certain indicators, and often differ significantly. In addition, it is not clear on what basis the calculations were made. As an alternative decision-makers may use econometric models, such as NECMOD for Polish economy, but firstly they are usually published with some delay, and therefore do not always reflect the current changes in macroeconomic trends and, secondly, the company may not have adequate data to supply such a model.

Another alternative is to use artificial intelligence methods such as neural networks or genetic programming. While the use of neural networks has been widely reported in the literature (sample reviews of ANN applications for economic forecasting can be found in [1],[6] or [7]), the papers on applications of genetic programming in this area are much rarer. Few examples of such papers will be presented in section 3.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 445–454, 2011.

Section 2 provides a brief description of genetic programming method, followed by the description of a collective approach to multi-model finding.

Finally, in section 4, three examples are provided in order to show the effectiveness of the proposed method for short-term prediction of selected macroeconomic indicators. For each indicator one concrete model was automatically chosen from the set of models generated by genetic programming agents. However, decision-makers with appropriate economic knowledge could choose a different model based on their preferences concerning the number of variables, the set of variables or the set of functions used in specific models.

2 Genetic Programming as Models Generation Method

Genetic programming (GP) is a general name for the class of evolutionary methods, which can solve problems without requiring the user to know or specify the form or structure of the solution in advance [12]. In this sense they are similar to artificial neural networks, however, thanks to the representation of the solutions used in genetic programming they can be applied to much wider range of problems, including linear, tree-based or graph-based problems. The problem itself in genetic programming can be seen as any computer program usually the set of terminal and non-terminal symbols (e.g. mathematical operators and functions) as long as the fitness function for this program (i.e. how good is the program comparing to another one) can be evaluated in any time.

Early forms of genetic programming can be found in Cramer [4] who proposed tree-based representation of computer programs. For the first time the name "genetic programming" was used by Koza in [9] and since then the method had been successfully applied to various problems from many scientific disciplines. Genetic programming proposed by Koza operated on representation of computer programs in the form of trees, particularly so called syntax (or grammar) trees. Such syntax tree comprises the sequence of functional components in such sense that tree root represents computer program output, nodes represent functions and their arguments are placed in subsequent nodes which are also functions or leafs which represents terminal symbols e.g. name of variables or constants. Thus the solution can be easily translated to LISP expression as proposed by Koza. Fig. 1 illustrates a simple computer program, which can be read as f(x,y) = x*x + y/2 and written in LISP language as (+ (* x x)(/ y 2)).

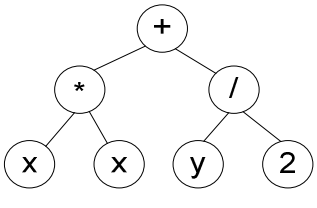


Fig. 1. A computer program represented as the syntax tree

The sets of terminal symbols and functions determine how computer program can look like. The main algorithm of genetic programming is outlined in Fig. 2.

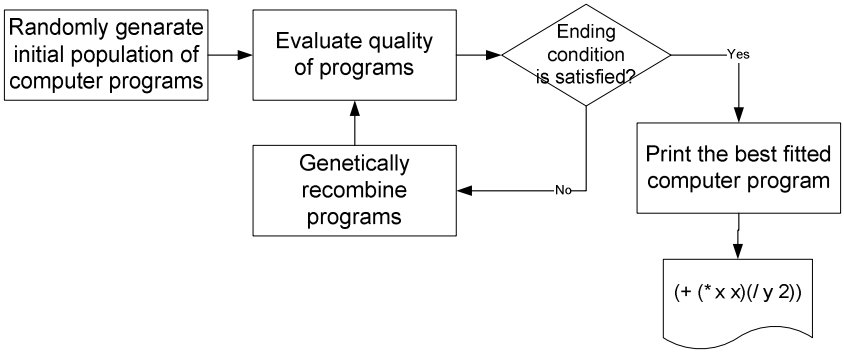


Fig. 2. Main algorithm of the genetic programming

The initial population consists of solutions (called in evolutionary methods individuals) which are random combination of functional components and terminals from the set of symbols allowed. Next generation is created by genetic recombination of individuals in a current population. The two basic genetic operators in evolutionary computation are crossover and mutation of genes in individuals. Fig. 3 illustrates two parent computer programs and two offspring created as the result of genes exchange. Subtrees in offspring are swapped below randomly selected crossover point.

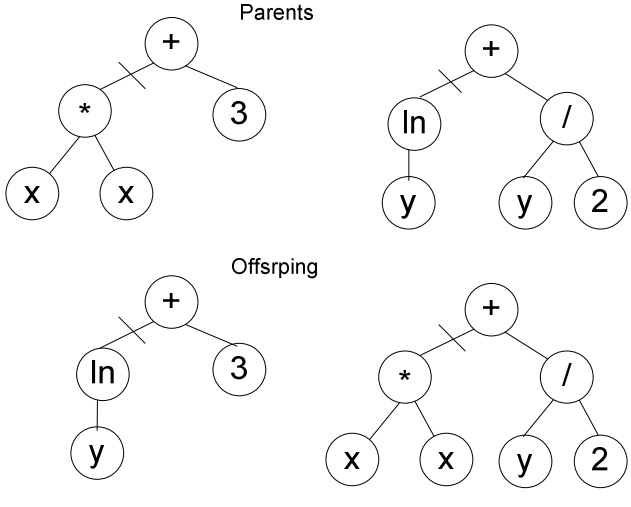


Fig. 3. Crossover operator creating two offspring from two parents

Some genes in offspring can be altered with a given probability by the means of two kinds of mutation operators. Point mutation simply changes a randomly chosen function or terminal symbol into another from the set of possible symbols. In the subtree mutation a subtree below mutation point is changed into another randomly generated subtree. This is shown in Fig. 4.

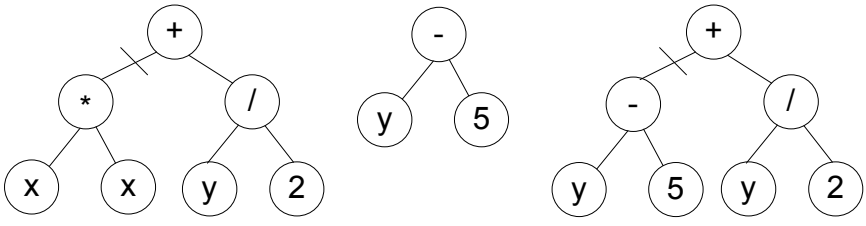


Fig. 4. Subtree mutation operator

After the process of genetic recombination new individuals are evaluated according to the fitness (quality) function. Evolution is continued until satisfactory value of the fitness function is found or a given number of generations is reached.

The problem of finding a forecasting model for such complex phenomenon like macroeconomic environment with a lot of explanatory variables requires an effective mechanism of guiding the search process of genetic programming. Gene expression programming method (GEP) proposed by Ferreira [5] and used in this paper utilizes a collective adaptive system allowing the genetic programming to generate the best possible model in each iteration on the basis of the models found so far.

GEP method, however, requires a decision-maker to arbitrary chose some startup parameters like the set of allowed functions, acceptability of constants, type and range of constants, number of components (subprograms) and the operator connecting them. Depending on these parameters a completely different model can be generated. Thus we propose to use a multi-agent approach in order to collect many models and then choose the most suitable for the decision-maker either automatically or manually.

3 Genetic Programming in Economic Modeling

One of the first attempts of applying genetic algorithms to economic modeling was carried out by Koza in [10]. He examined the well know formula of non-linear exchange equation P= MV/Q relating the price level (P), money supply (M), gross national product (Q) and velocity of money (V) for American economy. The terminal symbols where Gross National Product expressed in 1982 dollars (GNP82), Gross National Product Deflator (GD), M2 money supply and monthly interests rates yields of 3-month Treasury bills averaged for each quarter (FYGM3). Koza used the following set of functional symbols F={+, -, *, %, log ,exp) in order to "rediscover" the exchange equation on the basis of 120 quarterly data from Q1 1959 to Q4 1998. The LISP expression received by him was (* 0.885 (* 0.885 (% (- FM2 (- (- (* 0.885 FM2) FM2) FM2)) GNP82))) and could be transformed into simple equation:

GD=M2*1.6565/GNP82, which was almost identical with the equation developed by American economists GD = (M2 * 1.6527)/GNP82.

Cobweb model is probably one of the most examined economic models by researchers dealing with evolutionary computation. Arifovic [2] examined the convergence results of the model obtained by Wellford (1989) by applying two versions of genetic algorithms to this issue. She found that in each simulation of the basic GA, individual quantities and prices exhibited fluctuations for its entire duration and do not result in convergence to the rational expectations equilibrium values, which is quite inconsistent with the Wellford's experimental results with human subjects. Second GA version (augmented GA) converged to the rational expectations equilibrium values for all sets of cobweb model parameter values, including both stable and unstable cases, and could capture several features of the experimental behavior of human subjects better than other simple learning algorithms.

Chen and Yeh [3] found that the rational expectations equilibrium price can also be discovered by genetic programming. Furthermore, genetic programming required much less prior knowledge than genetic algorithms. The reasonable upper limit of the price and the characteristic of the equilibrium which is assumed as the prior knowledge in genetic algorithms can all be discovered by genetic programming. In addition, GP-based markets have a self-stabilizing force which is capable of bringing any deviations caused by mutation back to the rational expectations equilibrium price. All of these features showed that genetic programming can be a very useful tool for economists to model learning and adaptation.

4 Short-Term Forecasting of GDP, Investments and Loan Rates in Poland Using Genetic Programming

On the basis of more than 50 parameters describing financial and real nature of the economy, the authors tried to discover models that best describe GDP, investments and loan rates in Poland using selected predictors. Quarterly time series data came from period of Q1 1995 to Q4 2008. Macroeconomic indicators used for modeling have been chosen on the basis of essential analysis and they describe chosen, most important macroeconomic trends that influence enterprises. Explanatory variables have been selected in such way that description by a single variable should be unique and that data obtained should be reliable and comparable in corresponding periods. Dimensions of financial and real spheres have been analyzed in nominal, quarterly values.

The following variables were used (terminal symbols for GP):

- GDP - nominal GDP (millions PLN)
- M0, M1, M2, M3 - money supply (avg. in quarter, millions PLN)
- CPI - price index of consumer goods and services (Dec.1994=1, avg. in quarter)
- PPI - price indices of sold production of industry (Dec.1994=1, avg. in quarter)
- PPB - price indices of construction and assembly production (Dec.1994=1, avg. in quarter)
- IR - current inflation rate (y/y, source: NBP, avg. in quarter)
- IE - avg. expected inflation rate in next 12 months (source: NBP, avg. in quarter)

- CR - commercial loan interest rates (source: Rzeczpospolita, avg. in quarter)
- DR - deposit interest rates in commercial banks (source: Rzeczpospolita, 6-months fixed, avg. in quarter)
- LR - lombard rates (source: NBP, avg. in quarter)
- WS - average monthly gross wages and salaries (nominal, in PLN)
- FCF - gross fixed capital formation (current prices, millions PLN)
- SPI - sold production of industry (current prices, millions PLN)
- SPB - sale of construction and assembly production (current prices, millions PLN)
- UR - unemployment rate (avg. in quarter)
- PEE - average paid employment in enterprise sector (avg. in quarter, thousands)
- GTR - gross turnover profitability ratio (avg. in quarter)
- NTR - net turnover profitability ratio (avg. in quarter)
- NFR -net financial results of non-financial enterprises (in quarters, millions PLN)
- SB - the state budget result (aggregated, quarter end, millions PLN)
- EDB - balance of external debts (aggregated, end of the quarter, millions PLN)
- ORC - changes in official reserve assets (avg. in quarter, millions PLN)
- ORA - official reserve assets in PLN (USD/PLN rate at month end, avg. in quarter, millions PLN)
- UP - average USD/PLN exchange rate in the quarter
- UE - USD/EUR ratio since 1999 (month end, avg. in quarter)
- EP - Average EUR/PLN exchange rate in the quarter
- W1, W3, W6 - WIBOR 1M, 3M and 6M rate, respectively (avg. in quarter)
- NDA - net domestic assets (avg. in quarter, millions PLN)
- DGC - domestic claims on the central government, net (avg. in quarter, millions PLN)
- ADN - amounts due from the non-financial sector (avg. in quarter, billions PLN)
- NFA - net foreign assets (avg. in quarter, billions PLN)
- EL - loans in the enterprise sector
- HL - loans in the household sector
- MLN - liabilities of monetary financial institutions for non-financial sector (avg. in quarter, millions PLN)
- MLE - liabilities of monetary financial institutions for enterprise sector (avg. in quarter, millions PLN)
- MLH - liabilities of monetary financial institutions for household sector (avg. in quarter, millions PLN)
- FOB - export value as FOB (source: GUS, avg. in quarter, millions PLN)
- CIF - import value as CIF (source: GUS, avg. in quarter, millions PLN)
- FTT - foreign trade turnover (quarterly balance, millions PLN)
- EPR - transaction price rate of export (aggregated, avg. in quarter)
- IPR - transaction price rate of import (aggregated, avg. in quarter)
- MFI - import of mineral fuels, lubricants, and related materials in quarters (millions PLN)
- MTI - import of machinery and transport equipment in quarters (millions PLN)
- EG - electricity generation (quarterly, in GWh)

- PI - import of petroleum (quarterly, in Kt)
- PP - petroleum price (avg. in quarter in USD/barrel)
- WST - Warsaw Stock Exchange turnover (source: NBP, monthly, avg. in quarter)
- WSC - Warsaw Stock Exchange capitalization (source: NBP, month end, avg. in quarter)
- WIG - WIG Index (source: NBP, month end, avg. in quarter)

4.1 Model for GDP

First the genetic programming method was applied to find a model describing gross domestic product (GDP). After performing 10,000 generations for each agent and finding 10 different models the following one described by equation (1) has been chosen with the minimum number of variables (13) and the smallest function set F= {+, -, *, sqrt, cubert}.

$$GDP_t = \sqrt[3]{M2_{t-4} + MLN_{t-4}} + \sqrt{FTT_{t-4} + HL_{t-4} - MTI_{t-4}} + SPB_{t-2} - SPB_{t-3} + \sqrt[6]{EG_{t-1}}\,SPB_{t-4} + SPI_{t-1} + \sqrt[3]{M3_{t-1}(ADN_{t-1} - CIF_{t-1} + MTI_{t-4})NTR_{t-4}UR_{t-4}} \quad (1)$$

Observed and fitted values have been plotted in Fig. 5. Coefficient of determination of the model for the data provided was $R^2 = 0.998$.

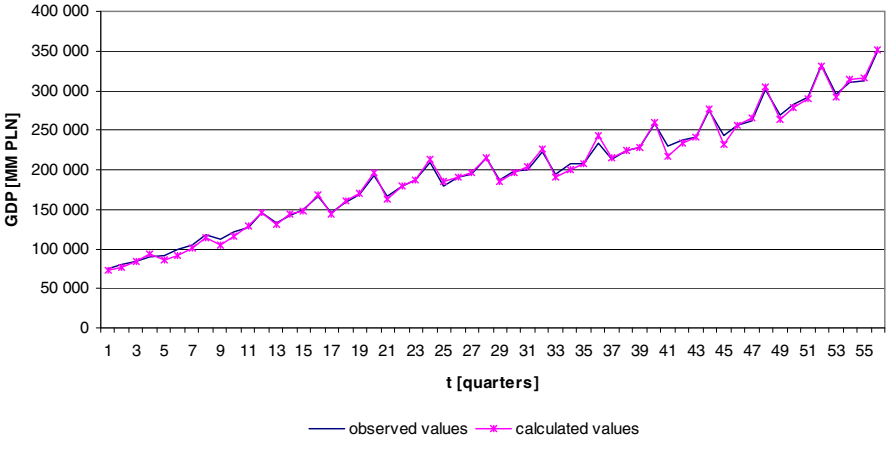


Fig. 5. GDP estimation by genetic programming

In order to evaluate the quality of the model produced by the genetic programming method it was compared with artificial neural networks (best of 5 models) created automatically by Statistica 8.0 system. The coefficient of determination for ANN was almost the same as for genetic programming having R^2=0.997.

4.2 Model for Commercial Loan Rates

A model for commercial loan interest rates for enterprises in Poland has been taken as the next example. Again after 10,000 generations performed by each agent the equation (2) has been chosen.

$$CR_t = PPI_{t-4} + W6_{t-1} + CPI_{t-3} + \frac{MLN_{t-1} - FTT_{t-3} - M2_{t-4}}{MTI_{t-1} + WSC_{t-3}} + \frac{(EDB_{t-2}UR_{t-4} + ORA_{t-1})NTR_{t-4}}{CPI_{t-4}GDP_{t-4}} + \frac{NFA_{t-2}WIG_{t-3}^{\ 2}}{(WS_{t-1} - MTI_{t-2})NDA_{t-1}UR_{t-3}} \quad (2)$$

This time the chosen model is more complex and has as many as 17 variables utilised and the set of functions consists of 5 functions F={+, -, *, /, square}. If more agents were be involved a simpler model could be achieved.

Observed and fitted values for the model have been shown in Fig. 6. Coefficient of determination was $R^2 = 0.994$.

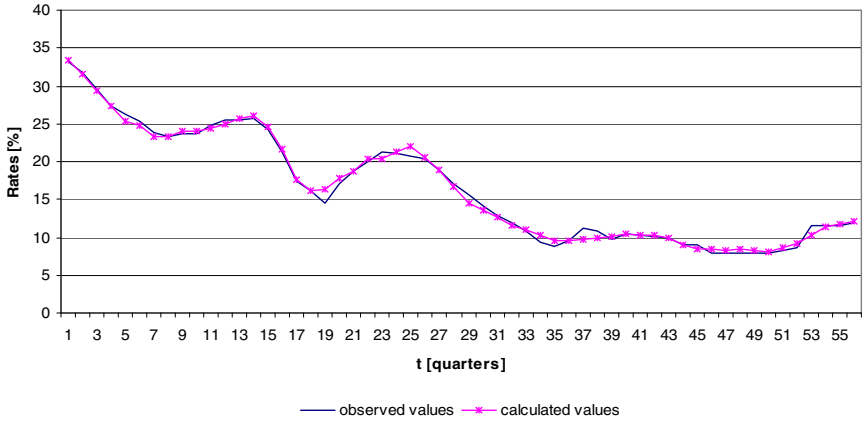


Fig. 6. Loan rates estimated by genetic programming

The model obtained by GP was also compared with artificial neural network created in Statistica 8.0 package and also in this case the coefficient of determination for ANN was very similar having $R^2 = 0.992$.

4.3 Model for Investments

The last test example concerned discovering a model for investments volume described by gross fixed capital formation (in current prices).

The model chosen out of 10 different models is described by the equation (3).

$$FCF_t = -4NFA_{t-4} - DR_{t-3}GTR_{t-3}NFA_{t-4} + NFR_{t-3} + PEE_{t-1} + \frac{SPB_{t-1}}{PPB_{t-4}} - 2SPB_{t-3} + 5SPB_{t-4} + \frac{UR_{t-4}}{IE_{t-3}} \quad (3)$$

This time the model chosen from the result set used only 9 variables and 4 simple functions F={+, -, *, /}. Observed and fitted values for this model have been shown in Fig. 7. Coefficient of determination was $R^2 = 0.989$.

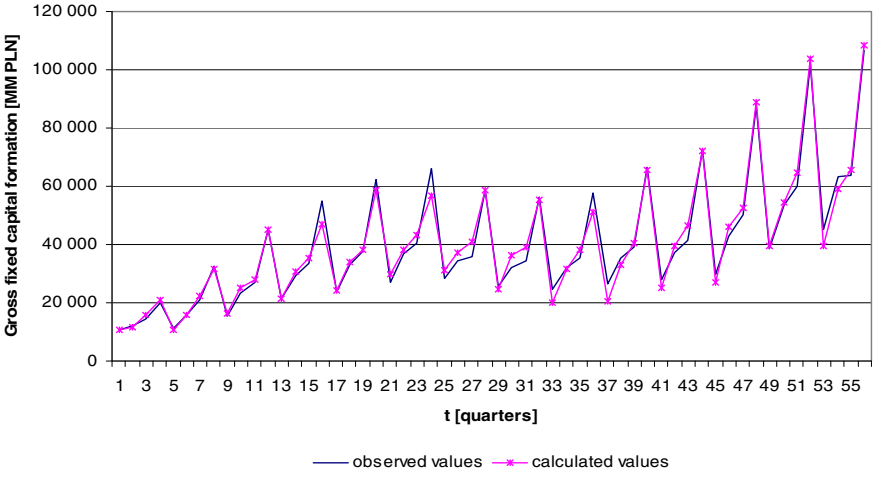


Fig. 7. Investments as gross fixed capital formation estimated by GP

Finally artificial neural network for FCF was created in Statistica 8.0 achieving this time better value for coefficient of determination $R^2 = 0.996$. The difference comparing with R^2 computed for the model generated by genetic programming is relatively small, but it may be further reduced by introducing more agents with different parameters.

5 Conclusions

The main purpose of the paper was to show that genetic programming can be successfully used as a tool for knowledge discovering about the macroeconomic environment of an enterprise. As this is a very computational demanding task the use of computational intelligence helps to achieve simpler and more reliable models. GEP method used in this paper has been implemented in Java language and was run in parallel for each agent. However more sophisticated multi-agent systems could be built using e.g. JADE framework in order to increase the number of agents involved in models discovering, and as a consequence, to improve the quality of the final model proposed for a decision-maker. An example of such approach can be found e.g. in [8].

Computational examples provided by the authors prove that the proposed method is as efficient as artificial networks regarding the estimation quality of macroeconomic indicators. The advantage of GP is that decision makers are provided with the set of concrete models, from which the most suitable one can be automatically or manually chosen to reflect certain economic rights and dependencies.

Moreover, genetic programming as a method for knowledge discovering can be applied also in other areas of the decision making process in a company, such as demand planning or production planning and then combine knowledge achieved from the company itself with the knowledge acquired from its environment.

Additional tests for the stability of the presented models should be done, facing e.g. world economy recession after 2007. Also the selection algorithm for the final model proposed to the decision-maker should be improved. Now this procedure takes into account only the number of variables and the number of functions used in a particular model generated by a single agent. This not always leads to the simplest and the most legible model.

References

1. Aminian, F., Suarez, E.D., Aminian, M., Walz, D.T.: Forecasting Economic Data with Neural Networks. Journal of Computational Economics 28(1), 71–88 (2006)
2. Arifovic, J.: Genetic Algorithms Learning and the Cobweb Model. Journal of Economic Dynamics and Control 18(1), 3–28 (1994)
3. Chen, S.-H., Yeh, C.-H.: Genetic Programming Learning and the Cobweb Model. In: Angeline, P., Kinnear Jr., K.E. (eds.) Advances in Genetic Programming, ch. 22, pp. 443–466. MIT Press, Cambridge (1996)
4. Cramer, N.L.: A representation for the Adaptive Generation of Simple Sequential Programs. In: Grefenstette, J.J. (ed.) Proceedings of an International Conference on Genetic Algorithms and the Applications. Carnegie Mellon University, Pittsburgh (1985)
5. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems (2001)
6. Freisleben, B., Ripper, K.: Economic forecasting using neural networks. In: Proceedings of IEEE International Conference on Neural Networks (1995)
7. Herbrich, R., Keilbach, M., Graepel, T., Bollmann–Sdorra, P., Obermayer, K.: Neural Networks in Economics: Background, Applications, and New Developments. Advances in Computational Economics 11, 169–196 (1993)
8. Jędrzejowicz, P., Ratajczak-Ropel, E.: Agent-Based Gene Expression Programming for Solving the RCPSP/max Problem. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 203–212. Springer, Heidelberg (2009)
9. Koza, J.R.: Genetic Programming: On The Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
10. Koza, J.R.: Genetic programming for economic modeling. Statistics and Computing 4(2), 187–197 (1994)
11. Koza, J.R., Poli, R.: A genetic programming tutorial. In: Burke, E. (ed.) Introductory Tutorials in Optimization, Search and Decision Support. Stanford University, Stanford (2003)
12. Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming. Lulu Enterprises (2008)

Parallel Appearance-Adaptive Models for Real-Time Object Tracking Using Particle Swarm Optimization

Boguslaw Rymut and Bogdan Kwolek

Rzeszów University of Technology
W. Pola 2, 35-959 Rzeszów, Poland
{brymut,bkwolek}@prz.edu.pl

Abstract. This paper demonstrates how appearance adaptive models can be employed for real-time object tracking using particle swarm optimization. The parallelization of the code is done using OpenMP directives and SSE instructions. We show the performance of the algorithm that was evaluated on multi-core CPUs. Experimental results demonstrate the performance of the algorithm in comparison to our GPU based implementation of the object tracker using appearance-adaptive models. The algorithm has been tested on real image sequences.

1 Introduction

One of the rising stars of Collective Intelligence is Particle Swarm Optimization (PSO), developed in 1995 by Kennedy and Eberhart [10]. PSO is a derivative-free optimum search algorithm based on the collective intelligence of a group of simple agents, which interact with each other and with their environment. Such local interactions result in the global behavior of the whole population. The individual entities are simple, knowing no more than their own current locations and fitness scores, their personal best locations, and the swarm's best location. In Particle Swarm Optimization, each potential solution to the problem is called particle and the word "swarm" comes from the irregular movements of the particles in the problem space. PSO was inspired by the social behavior of bird flocking and fish schooling and it has its roots in artificial life and social psychology, as well as in engineering and computer science.

PSO can be used to solve a wide spectrum of different optimization problems, including tasks that can be solved using Genetic Algorithms. Some example applications include function minimization and neural network training. In [13], a parallel Particle Swarm Optimization algorithm has been proposed. The algorithm is synchronous and it has good performance on problems where the fitness evaluations require the same amount of time. In [1], a PSO algorithm has been successfully applied to object tracking. Recently, in [12] it has been shown that a GPU implementation of a PSO based object tracker can exhibit a more than 40-fold speed-up over a CPU implementation.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 455–464, 2011.

Vision based object tracking is a mandatory step in many applications, such as surveillance, traffic monitoring, sport event analysis, mixed virtual reality and even in observational problems in the natural science. It is one of the major steps toward understanding video content. The goal of visual object tracking is to repeatedly localize an object of interest in successive frames. Most object trackers search for the target locally in new frames using a similarity measure between a reference object model and candidate targets. Thus, the task of object tracking can be considered as a numerical optimization problem, where a local optimization is used to track the local mode of the similarity measure in a parameter space of translation, rotation and scale. In [14], it was shown that in tasks consisting in tracking the face or the human a PSO tracker outperforms a particle filter based tracker in terms of accuracy. Particle filters [8], which are sequential Monte Carlo methods based on point mass representations of probability densities, can be employed to any state-space model and generalize the conventional Kalman filtering methods. Currently, they are widely used to track targets in image sequences. Their weakness is that they require a large number of particles for accurate estimation of state variables lying in a high dimensional space. In contrary, PSO has better capabilities to explore the search space as well as facility to balance the trade-off between exploration and exploitation.

Image sequences acquired in real scenarios pose specific challenges to tracking algorithms, due to, for example, low or variable lighting conditions, scale change and changes of the appearance. On-line appearance models [9] are one of the most successful approaches to object tracking. An on-line variant of the Expectation Maximization (EM) [5] algorithm is typically used to learn the parameters of the appearance models. It identifies stable structures and naturally combines such structures with transient image information in the $\mathcal{WSL}$ framework [9]. Adaptive appearance models were successfully applied in various applications, including challenging tasks like model based articulated object tracking [2] [11]. One of the drawbacks of the object trackers, which are built on adaptive appearance models is considerable computation time. This motivated us to develop the GPU implementation of a tracker built on the appearance-adaptive models and the particle swarm optimization [12].

In this work we show the processing times of our GPU object tracker, which was extended about the affine transformations. We demonstrate the computation times as well as speeds-up that have been obtained on GPU as well as on multi-core CPUs. The parallel computations on multi-core CPUs were achieved using Streaming SIMD Extensions (SSE) and Open Multi-Processing (OpenMP). Currently, OpenMP is widely utilized standard for parallelizing programs in a shared memory environment [3]. Experimental results show that our parallel algorithm exhibits about 4.5 fold speed-up over standard C/C++ implementation. Using 256 and 512 particles in PSO the tracking at a 4-core CPU can be done with 20 and about 10 frames per second, respectively. Using a 2-core CPU of a low-cost notebook we tracked objects with 6 fps. The algorithm has been tested on real image sequences.

2 Appearance-Adaptive Models for Object Tracking

Let $I_t(x)$ denote the brightness value at the location $x = [x_1, x_2]^T$ in an image $\mathcal{I}$ that was acquired at time t. Let $\mathcal{R}$ be a set of J image locations $\{x(j) \,|\, j = 1, 2, ..., J\}$ defining a template. $Y_t(\mathcal{R}) = \{I_t(x(j)) \,|\, j = 1, 2, ..., J\}$ is a vector of the brightness values at locations $x(j)$ in the template. We assume that the transformations of the template can be modeled by a parametric motion model $g(x; \omega_t)$, where x denotes an image location and ω_t is a motion parameter vector.

The image variations of planar objects that undergo orthographic projection can be described by a six-parameter affine motion models [6]:

$$g(x; \omega) = \begin{bmatrix} a & d \\ c & e \end{bmatrix} x + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = Ax + u, \tag{1}$$

where the motion parameter vector $\omega = (a, c, d, e, u_1, u_2)^T$. The affine transformation allows different stretching along rows and columns of an image and shearing. Applying an affine transformation on image patches is called warping. We assume that the warped current image is a representation of the reference object template. The goal of the tracking is to estimate the warping parameters $\hat{\omega}_t$ in each frame.

Our intensity-based appearance model consists of three components, namely, The $\mathcal{W}$-component accounting for the two-frame variation, the $\mathcal{S}$-component depicting the stable structure within all previous observations and $\mathcal{F}$ component, which takes the place of the original lost component $\mathcal{L}$ and represents a fixed template. The model $A_t = \{\mathcal{W}_t, \mathcal{S}_t, \mathcal{F}_t\}$ represents the appearances existing in all observations up to time $t - 1$. It is a mixture of Gaussians [9] with centers $\{\mu_{i,t} \,|\, i = w, s, f\}$, their corresponding variances $\{\sigma^2_{i,t} \,|\, i = w, s, f\}$ and mixing probabilities $\{m_{i,t} \,|\, i = w, s, f\}$.

The fitness score has been evaluated according to the following equation:

$$f(\omega_t) = \prod_{j=1}^{J} \sum_{i=w,s,f} \frac{m_{i,t}(j)}{\sqrt{2\pi\sigma^2_{i,t}(j)}} \exp\left[-\frac{1}{2}\left(\frac{\check{Y}_t(j) - \mu_{i,t}(j)}{\sigma_{i,t}(j)}\right)^2\right] \tag{2}$$

where $\check{Y}_t(\mathcal{R}) = Y_t(g(\mathcal{R}; \omega_t))$. In the fitness function we utilize a recursively updated appearance model, which depicts stable structures seen so far, initial object appearance as well as two-frame variations.

The update of the current appearance model A_t to A_{t+1} is done using the EM algorithm. For a template $\hat{Y}_t(\mathcal{R}) = Y_t(g(\mathcal{R}; \hat{\omega}_t))$, which has been obtained through applying the estimated $\hat{\omega}_t$, we evaluate the posterior contribution probabilities as follows:

$$o_{i,t}(j) = \frac{m_{i,t}(j)}{\sqrt{2\pi\sigma^2_{i,t}(j)}} \exp\left[-\frac{1}{2}\left(\frac{\hat{Y}_t(j) - \mu_{i,t}(j)}{\sigma_{i,t}(j)}\right)^2\right] \tag{3}$$

where $i = w, s, f$ and $j = 1, 2, ..., J$. The posterior contribution probabilities (with $\sum_i o_{i,t}(j) = 1$) are utilized in updating the mixing probabilities in the following manner:

$$m_{i,t+1}(j) = \gamma o_{i,t}(j) + (1-\gamma) m_{i,t}(j) \quad | \, i = w, s, f \tag{4}$$

where γ is accommodation factor. Then, the first and the second-moment images are determined as follows:

$$M_{1,t+1}(j) = (1-\gamma) M_{1,t}(j) + \gamma o_{s,t}(j) \hat{Y}_t(j) \tag{5a}$$

$$M_{2,t+1}(j) = (1-\gamma) M_{2,t}(j) + \gamma o_{s,t}(j) \hat{Y}_t^2(j) \tag{5b}$$

In the last step the mixture centers and the variances are calculated as follows:

$$\mu_{s,t+1}(j) = \frac{M_{1,t+1}(j)}{m_{s,t+1}(j)}, \;\; \sigma_{s,t+1}(j) = \sqrt{\frac{M_{2,t+1}(j)}{m_{s,t+1}(j)} - \mu_{s,t+1}^2(j)} \tag{6}$$

$$\mu_{w,t+1}(j) = \hat{Y}_t(j), \quad \sigma_{w,t+1}(j) = \sigma_{w,1}(j) \tag{7}$$

$$\mu_{f,t+1}(j) = \mu_{t,1}(j), \quad \sigma_{f,t+1}(j) = \sigma_{f,1}(j) \tag{8}$$

In order to initialize the model A_1 the initial moment images are set using the following formulas: $M_{1,1} = m_{s,1} Y_{t0}(\mathcal{R})$ and $M_{2,1} = m_{s,1}(\sigma_{s,1}^2 + Y_{t0}^2(\mathcal{R}))$.

3 Object Tracking Using PSO

PSO is a population based algorithm introduced in [10] that utilizes a set of particles representing potential solutions of the optimization task. Despite the simplicity of the individual particles, the swarm as a whole has a remarkable level of coherence and coordination. Each solution is represented as a series of coordinates in n-dimensional space. A number of particles are initialized randomly within the search space. Every particle flies in the solution space with a velocity adjusted dynamically according to its own experience and the experience of the whole swarm. Each particle has a very simple memory of its personal best solution so far, called *pbest*. The global best solution for each iteration is also determined and is termed *gbest*. On each iteration, every particle is moved a certain distance from its current location, influenced a random amount by the *pbest* and *gbest* values. The particles are evaluated according to a user defined fitness function $f()$. The velocity of each particle i is updated in accordance with the following equation:

$$v_i^{(j)} \leftarrow w v_i^{(j)} + c_1 r_1^{(j)} (pbest_i^{(j)} - \omega_i^{(j)}) + c_2 r_2^{(j)} (gbest_i - \omega_i^{(j)}) \tag{9}$$

where $v_i^{(j)}$ is the velocity in the $j-$th dimension of the $i-$th particle, c_1, c_2 denote the acceleration coefficients, $r_1^{(j)}$ and $r_2^{(j)}$ are uniquely generated random

numbers in the interval [0.0, 1.0], and w stands for an inertia weight. The inertia weight allows the balance of the exploration and exploitation abilities of the swarm as well as eliminates the need for velocity clamping.

The first part in (9) takes into account the previous velocity, which provides the necessary momentum for particles to fly across the search space. The second part is known as the cognitive component and represents the personal thinking of each particle. This component encourages the particles to fly toward their own best position *pbest* found so far. The third part is known as the social component and represents the collaborative effect of the particles in finding the global optimum. This component pulls the particles toward the best position(s) found so far by their neighbors. The inertia part keeps particles to explore new areas while the cognitive and social parts try to keep them exploiting around the visited points.

The new position of a particle is calculated in the following manner:

$$x_i^{(j)} \leftarrow x_i^{(j)} + v_i^{(j)} \tag{10}$$

The local best position of each particle is updated as follows:

$$pbest_i \leftarrow \begin{cases} \omega_i, & \text{if } f(\omega_i) > f(pbest_i) \\ pbest_i, & \text{otherwise} \end{cases} \tag{11}$$

and the global best position *gbest* is defined as:

$$gbest \leftarrow \arg\max_{pbest_i} \{f(pbest_i)\} \tag{12}$$

The value of velocity v_i should be restricted to the range $[-v_{max}, v_{max}]$ to prevent particles from moving out of the search range. In [10], Eberhart and Kennedy suggested that the PSO should perform better if v_{max} in each dimension is set equal to the dynamic range of that dimension. In some optimization problems the local best version of PSO, where particles are influenced by the best position within their neighborhood, as well as their own past experience can give better results. While such a configuration of the PSO is generally slower in convergence than algorithm with *gbest*, it typically results in much better solutions and explores a larger part of the problem space.

At the beginning of the optimization the PSO initializes randomly locations as well as the velocities of the particles. Then the algorithm selects *pbest* and *gbest* values. Afterwards, equations (9)-(12) are called until maximum iterations or minimum error criteria is attained. After that, given $\hat{\omega}_t = gbest$ we calculate $\hat{Y}_t$, and then update of the object model using formulas (3)-(8).

In contrast to traditional optimization problems with stationary optima, tracking objects in image sequences requires the algorithm to find the optimum not once, but in every successive image. There are various approaches to dealing with moving objects, such as decaying the score of the best location after every. In consequence, such an operation results in forcing the swarm to continually search for a better location. In particular, it prevents the swarm from completely converging to a single point, allowing the swarm agents to be appropriately spaced in order to quickly reacquire a target in the next image.

In the simplest solution the tracking can be realized as deterministic searching of window location whose content best matches a reference window content. PSO allows us to avoid such time consuming exhaustive searching for the best match. It provides an optimal or sub-optimal match without the complete knowledge of the searching space. In PSO based tracking, at the beginning of each frame in the initialization stage, an initial position is assigned to each particle

$$\omega_{i,t} \leftarrow \mathcal{N}(gbest, \Sigma) \tag{13}$$

given the location $gbest$ that has been estimated in the previous frame $t-1$. In the evaluation phase the fitness value of each particle is determined on the basis of (2). Every particle has an associated affine transformation, which is used to warp a video frame.

4 Experimental Results

The experiments were conducted on a desktop PC with 4 GB RAM, Intel Core i5, 2.8 GHz processor with NVIDIA GeForce 9800 GT graphics card. The graphics card has 14 stream multiprocessors with 1.5 GHz, each with 8 cores. It is equipped with 1024 MB RAM, 64 KB constant memory and 16 KB common memory for each multiprocessor. We implemented the algorithm in CUDA and compared the runtimes with its counterpart that was implemented in C/C++ and executed on the CPU. The CPU code was compiled with Intel C++ Compiler for Windows. Table 1 shows the running times of the tracking algorithm both on CPU and GPU as well as the speed-up. The communication delays for copying images from CPU to GPU and vice versa have not been taken into account. The most time-consuming operation of the algorithm is calculation of the fitness function (2). This operation amounts to 0.9 of the whole processing time.

Table 1. Tracking time [ms] and speed-up of GPU (NVIDIA GeForce 9800 GT) over CPU (Intel Core i5, 2.8 GHz) at a desktop PC

# particles	256	512	1024	2048	4096
CPU [ms]	250	405	949	1878	3737
GPU [ms]	61	69	72	90	165
CPU/GPU	4.1	5.7	13.2	20.9	22.7

OpenMP is a library (application program interface or API) that supports parallel programming on shared memory parallel computers. It consists of a set of directives (pragmas) and library routines that can be inserted into Fortran or C/C++ codes to enable use of more than one thread. It handles scheduling over available cores in a static fashion. OpenMP provides a fork-and-join execution model in which a program begins execution as a thread. The thread executes

sequentially until a parallelization directive for a structured block of code is found. If this takes place, such a thread creates a set of threads and becomes the master thread of the new group of threads. Each thread executes the same code redundantly until the end of the parallel section and the threads communicate by sharing variables. The exit point of a structured block is an implicit synchronization point for the master thread and the threads created for the block. After the synchronization the master thread continues with the computation and the other threads end. The advantage of OpenMP is that an existing code can be effortlessly parallelized by placing OpenMP directive instructions.

Table 2 contains tracking times that have been obtained using OpenMP. As we can see the efficiency of parallel computations is very high. The speed-up over the program executed without OpenMP support is about 3.3. The speed-up was achieved owing to data parallelism [7], which is also known as loop-level parallelism. It focuses on effective distributing the data across different parallel computing nodes. In the appearance-adaptive models the pixels are assumed to be independent and therefore they can be processed in parallel.

Table 2. Tracking time [ms] and speed-up of the GPU over the CPU (Intel Core i5, 2.8 GHz, OpenMP)

#particles	256	512	1024	2048	4096
CPU [ms]	73	145	284	573	1142
CPU/GPU	1.2	2.1	4.0	6.4	6.9

Table 3 shows computation times that were achieved using OpenMP and SSE instructions and registers. The functions like `exp`, `log`, `pow` were implemented using algorithms presented in [4] and SSE, SSE2 SSE3 and SSSE 3 (Supplemental Streaming SIMD Extension 3) instructions. All SSE instructions operate on 128-bit XMM registers. By the use of such registers the fitness score given by (2) can be computed very quickly. In our implementation we evaluate simultaneously all three Gaussians $\mathcal{N}(\check{Y}, \mu_i, \sigma_i)$, which are then simultaneously multiplied by the mixing probabilities. Such values are finally used in updating the fitness score. This process can be depicted in the following pseudo-code:

$$
\begin{aligned}
f_w &= m_w \times \mathcal{N}(\check{Y}, \mu_w, \sigma_w) \\
f_s &= m_s \times \mathcal{N}(\check{Y}, \mu_s, \sigma_s) \\
f_f &= m_f \times \mathcal{N}(\check{Y}, \mu_f, \sigma_f) \\
f &\leftarrow f + \texttt{log}(f_w + f_s + f_f)
\end{aligned}
$$

For each mixture j the mixing probabilities m_w, m_s, m_f are stored in 96 bits of single 128-bit XMM register. The corresponding mixture centers and the variances are stored in two XMM registers.

As we can see in Tab. 3 the speed-up is considerable. For 512 particles the algorithm is faster than the GPU algorithm. It is worth noting that for such a number of particles not all GPU resources were fully exploited.

Table 3. Tracking time [ms] and speed-up of the GPU over the CPU (Intel Core i5, 2.8 GHz, SSE and OpenMP)

#particles	256	512	1024	2048	4096
CPU [ms]	51	101	200	398	803
CPU/GPU	0.8	1.5	2.8	4.4	4.9

Table 4. Tracking time [ms] and speed-up of GPU (NVIDIA GeForce 9600M GS) over CPU (Intel Core 2 Duo, 2.2 GHz, OpenMP and SSE) at a notebook

#particles	256	512	1024	2048	4096
CPU [ms]	164	339	667	1327	2651
GPU [ms]	65	84	113	248	416
CPU/GPU	2.5	4.0	5.9	5.4	6.4

In Tab. 4 are presented the results that were obtained on a typical notebook with Intel Core 2 Duo 2.2 CPU and NVIDIA GeForce 9600M GS graphics card. As we can notice, the ratio of computation times for 4096 and 256 particles at the notebook is far larger comparison of the ratio on the desktop computer. The graphics card of the desktop has 14 multiprocessors, whereas the graphics card of the notebook has 4 multiprocessors and therefore the discussed ratio is smaller for the desktop computer.

The experimental results presented above indicate that OpenMP together with SSE instructions can lead to faster tracking algorithm on multi-core CPUs. More important, thanks to OpenMP support we achieved tracking in real-time using 265 particles, i.e. tracking with 13 fps, see Tab. 2. With OpenMP and SEE support we performed tracking using 512 particles at about 10 Hz, see also results in Tab. 3.

Figure 1 depicts some tracking results that were achieved using an active camera. The aim of the algorithm was to keep the face undergoing tracking at specific location in the image. In the tracking experiments the face moved in front of the wooden furniture, which has very similar color to skin color. It is worth to note that in such a scenario very popular skin-color based trackers are unable to track the target. The precision of tracking in the depicted image sequence was about 0.5 pix using PSO consisting of 256 particles. The precision was evaluated using the corners of the template as ground-truth, which has been determined manually. The size of the reference template is 32×32 pixels. The algorithm operates on images of size 640×480 pixels.

We conducted also experiments consisting in person following with a mobile robot. The camera was mounted on a mobile robot Pioneer 2DX. Sample experimental results are depicted in Fig. 2.

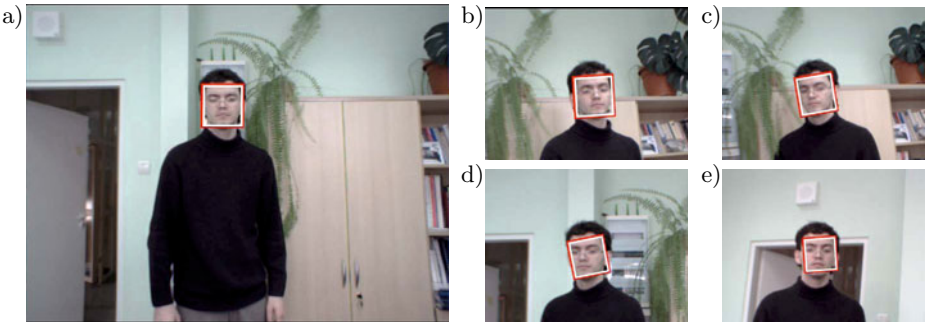


Fig. 1. Real-time tracking of the face using an active camera. Frame #0 a), #50 b), #100 c), #150 d) and #200 e).

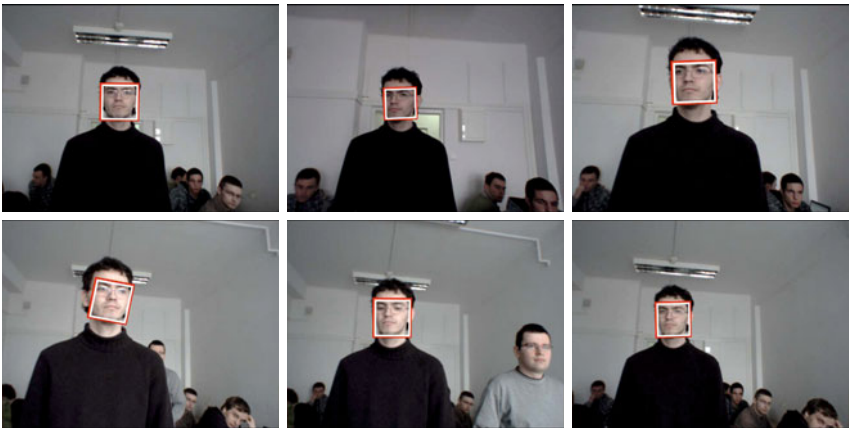


Fig. 2. Person following using a vision-guided mobile robot. Frames #55, 120, 165, 250, 275, 300 (left-to-right, top-to-bottom).

5 Conclusions

In this paper, we have shown how the particle swarm optimization tracker built on appearance-adaptive models can be accelerated significantly using OpenMP and SSE instructions. The results showed that our algorithm running on a 4-core CPU is about 4.5 times faster than an algorithm based on pure C/C++ code. As a result the tracking algorithm runs at frame-rates exceeding 20 frames per second. Future work will concentrate on particle swarm optimization based multiple object tracking with the use of the appearance-adaptive models. We intend to apply multiple swarms to make collective decisions for objects undergoing temporal occlusions.

Acknowledgement. This work has been partially supported by the National Science Centre (NCN) within the project N N516 483240.

References

1. Anton-Canalis, L., Hernandez-Tejera, M., Sanchez-Nielsen, E.: Particle swarms as video sequence inhabitants for object tracking in computer vision. In: Proc. of the Sixth Int. Conf. on Intelligent Systems Design and Applications, ISDA 2006, vol. 2, pp. 604–609. IEEE Computer Society, Washington, DC, USA (2006)
2. Balan, A.O., Black, M.J.: An adaptive appearance model approach for model-based articulated object tracking. In: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 758–765. IEEE Computer Society Press, Washington, DC, USA (2006)
3. Chapman, B., Jost, G., van der Pas, R., Kuck, D.: Using OpenMP: Portable Shared Memory Parallel Programming. MIT Press, Cambridge (2007)
4. Cody, W.J.: Software Manual for the Elementary Functions. Prentice-Hall series in computational mathematics. Prentice-Hall, Inc., Upper Saddle River (1980)
5. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. of the Royal Statistical Society. Series B 39(1), 1–38 (1977)
6. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Trans. on PAMI 20(10), 1025–1039 (1998)
7. Hillis, W.D., Steele Jr., G.L.: Data parallel algorithms. Commun. ACM 29, 1170–1183 (1986)
8. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. Int. J. of Computer Vision 29, 5–28 (2006)
9. Jepson, A.D., Fleet, D.J., El-Maraghi, T.: Robust on-line appearance models for visual tracking. IEEE Trans. on PAMI 25(10), 1296–1311 (2003)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE Int. Conf. on Neural Networks, pp. 1942–1948. IEEE Press, Piscataway (1995)
11. Kwolek, B.: 3D model-based tracking of the human body in monocular gray-level images. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 494–505. Springer, Heidelberg (2007)
12. Rymut, B., Kwolek, B.: GPU-supported object tracking using adaptive appearance models and particle swarm optimization. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6375, pp. 227–234. Springer, Heidelberg (2010)
13. Schutte, J.F., Reinbolt, J.A., Fregly, B.J., Haftka, R.T., George, A.D.: Parallel global optimization with the particle swarm algorithm. Int. J. for Numerical Methods in Engineering 61(13), 2296–2315 (2004)
14. Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M.: Sequential particle swarm optimization for visual tracking. In: IEEE Int. Conf. on CVPR, pp. 1–8 (2008)

Following the Leader – Particle Dynamics in Constricted PSO

Jacek Kabziński

Institute of Automatic Control, Technical University of Lodz,
Stefanowskiego 18/22 90-924 Lodz, Poland
jacek.kabzinski@p.lodz.pl

Abstract. We consider particle swarm optimization algorithm with a constriction coefficient and investigate particle dynamics without stagnation assumptions. We propose differential models of a particle following the swarm leader, while global best and personal best position are changing. We introduce three qualitative kinds of particles – a leader, a lazy follower and a sedulous follower with equations allowing quantitative investigation of parameter influence. This analysis constitutes an attempt to understand PSO dynamics and the choice of swarm parameters and inspires parameters adaptation.

Keywords: particle swarm optimization, convergence, stability analysis.

1 Introduction

Shortly after Eberhart and Kennedy [1] introduced particle swarm optimization (PSO) in their pioneer work, researchers' attention focused on several questions:

- How can we describe and analyze the dynamical behavior of a particle and the behavior of a whole swarm?
- How is this behavior influenced by the choice of parameters, for which parameters' values is it stable?
- How can we make use of this analysis to perform an informative choice or adaptation of parameters?

The standard PSO algorithm is described by difference equations with random coefficients and variables. Therefore it is tempting to analyze particle behavior using control theoretic approaches. The literature concerning PSO is very abundant and explosively growing, but the following two groups of references are particularly important to understand results (and simplifications) of research on particle dynamics and stability. (We are able to mention only the most representative publications of each group).

The first concept is based on deterministic trajectory analysis of a single, one-dimensional particle [2-5]. As a result of these studies, despite all simplifying assumptions, some important numerical aspects of swarm optimization were clarified,

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 465–474, 2011.

constriction factor was introduced [2] and several promising parameters' values were proposed.

The second approach accepts stochastic nature of PSO dynamics and concentrates on the stability analysis. Some researches use Lyapunov techniques to obtain robust stability in presence of randomness [6,7]. In most of the approaches stagnation is adopted as a simplification, so it is assumed that global best and local best positions are constant during the analysis [8]. An interesting attempt to the stochastic analysis of PSO, when the global best is considered as a stochastic process, is presented in [9] based on continuous time PSO model.

In this contribution we address the problem of particle's dynamics and stability. We study the most popular PSO version utilizing constriction factor together with cognitive and social (acceleration) parameters. The proposed analysis is based on a discrete-time model. We consider moving global best and personal best positions and we are especially interested in tracking abilities of a particle. We investigate the deterministic particle model as well as the stochastic one. We present equations for mean values dynamics only, although it is possible to derive higher moments' equations.

As a result we obtain a general equation of particle motion (section 3) and propose three qualitative categories of particles – the leader (section 4), the sedulous follower and the lazy follower (section 5) with equations allowing quantitative investigation of parameter influence. Conclusions regarding the parameter choice, the contribution of the particle to the swarm success and some features of optimization are presented in section 6.

2 Constricted PSO

We consider the most popular version of particle swarm optimization algorithm to solve minimization problems.

Potential solutions are represented by S numbered individuals or particles – M dimensional vectors $x_i \quad i = 1,2,\dots,S$, whose length is the number of degrees of freedom of the optimization problem.

A population of particles is initialized in the search space with random positions $x_i(0)$ and velocities $v_i(0)$.

During the t-th iteration particles are evaluated by the objective function $F(x)$ and the following values are memorized: $p_i(t)$ - the best position of the $i'th$ particle, $i = 1,2,\dots,S$ during iterations $1,2,\dots,t$ and $p_G(t)$ – the global best position found so far by particle informants. During this study it is not important whether we consider global or local swarm topology, so we may generally assume that for the i-th particle its neighborhood consists of the particles numbered $i-r, i-r+1, \cdots, i, \cdots i+r$.

Next, position update ('velocity') components are build for each particle $i = 1,2,\dots,S$:

$$v_i(t+1) = \chi[v_i(t) + c_1 R_{i1}{}^{\circ}(p_i(t) - x_i(t)) + c_2 R_{i2}{}^{\circ}(p_G(t) - x_i(t))] \qquad (1)$$

where:

χ - is the constriction coefficient,

c_1, c_2 - are cognitive and social (acceleration) parameters,

$v_i(t)$ - is the previous velocity of the $i'th$ particle,

R_{i1}, R_{i2} - are random M –dimensional vectors, whose components are uniformly distributed in [0,1] and ° denotes component-wise multiplication.

Then the position of each particle is updated:

$$x_i(t+1) = x_i(t) + v_i(t+1). \tag{2}$$

The iterations are repeated until the stopping criterion is fulfilled. The algorithm parameters are: number of particles S and radius of neighborhood r. In the standard PSO algorithm it is usually assumed that the search space of feasible solution is a M–dimensional hyperrectangle defined by the bounds x_{min}, x_{max} and the velocity components are also bounded by v_{min}, v_{max}.

Previous investigations of particle dynamics [2,4,5] motivated the choice of particular parameters' values

$$\chi = 0.729,\ \mathrm{c}_1 = \mathrm{c}_2 = 2.05 \tag{3}$$

assuring the compromise between particle convergence and exploration abilities.

3 One-Dimensional Particle Model

As each dimension of the particle movement is independent, in order to qualitatively observe the behavior of the particle, we will consider 1-dimensional particle represented by its position $x(t)$ and velocity $v(t)$:

$$v(t+1) = \chi\big[v(t) + c_1(t)\big(p(t) - x(t)\big) + c_2(t)(p_G(t) - x(t))\big] \tag{4}$$

$$x(t+1) = x(t) + v(t+1). \tag{5}$$

where $p(t)$ is personal best and $p_G(t)$ is global/neighborhood best position, $c_1(t),\ c_2(t)$ are independent realizations of random variables uniformly distributed in $[0, c_1]$ and $[0, c_2]$ respectively.

As we are interested in following the best particle, we will denote the distance from the actual position to $p_G(t)$:

$$y(t) = p_G(t) - x(t). \tag{6}$$

After introducing progressive difference operator

$$\Delta p_G(t) = p_G(t+1) - p_G(t) \tag{7}$$

we are able to describe particle dynamics by:

$$\begin{bmatrix} v(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} \chi & \chi\varphi(t) \\ -\chi & 1 - \chi\varphi(t) \end{bmatrix} \begin{bmatrix} v(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} -\chi c_1(t) & 0 \\ \chi c_1(t) & 1 \end{bmatrix} \begin{bmatrix} p_G(t) - p(t) \\ \Delta p_G(t) \end{bmatrix}, \tag{8}$$

where $\varphi(t) = c_1(t) + c_2(t)$.

For constant parameters $c_1(t) = c_1,\ c_2(t) = c_2$, the particle is described by linear discrete-time system with two state variables: $v(t), y(t)$ and two inputs: $p_G(t) - p(t)$ and $\Delta p_G(t)$. The system matrix

$$M(t) = \begin{bmatrix} \chi & \chi\varphi(t) \\ -\chi & 1 - \chi\varphi(t) \end{bmatrix} \tag{9}$$

was investigated intensively in previous publications [2,4]. We want to stress two features clearly visible from equation (8):

- differential action with respect to $p_G(t)$ is present in particle dynamics,
- although eigenvalues of $M(t)$ depend only on parameters χ and φ, the controllability of system (8) with respect to input $p_G(t) - p(t)$ depends on c_1 alone.

If $c_1(t),\ c_2(t)$ are independent realizations of random variables, equation (8) describes a chain of stochastic variables, where $v(t), y(t)$ depend on all previous $v(t-i), y(t-i)$ and $c_1(t-i),\ c_2(t-1)$, but $c_1(t),\ c_2(t), v(t)$ are independent, as well as $c_1(t),\ c_2(t), y(t)$.

Creation of the personal best position $p(t)$ is provided by the following algorithm:

$$p(t) = \begin{cases} x(t) & if\ \ F\big(x(t)\big) \le F(p(t-1)) \\ p(t-1)\ if\ \ F\big(x(t)\big) > F(p(t-1)) \end{cases}, \tag{10}$$

If at iteration t the particle approaches its best personal position $p(t) = x(t)$, algorithm (10) will work as a 'latch' blocking $p(t)$ for a period σ, till iteration $t + \sigma + 1$, when the personal best position will be updated again. We may say that there exists a sequence of 'improving' iterations $t_0 = 0, t_1, t_2, \ldots$, such that

$$p(t) = \begin{cases} x(t) & if\ \ t = t_k,\ \ k = 0,1,2,\ldots \\ x(t_k) & if\ \ t_k \le t < t_{k+1},\ \ k = 0,1,2,\ldots \end{cases} \tag{11}$$

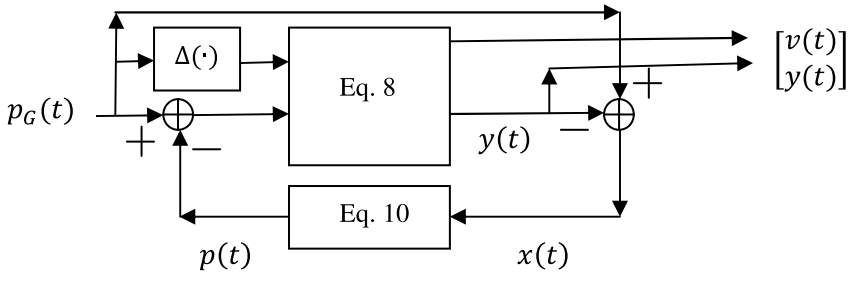


Fig. 1. Particle dynamics diagram

The period $\sigma = t_{k+1} - t_k - 1$ is of course variable, depending on activity of all particles, but useful conclusions may be derived investigating a case with constant σ. In this situation:

$$p(t) = \begin{cases} x(t) & if\ \ t = k(\sigma+1),\ \ k = 0,1,2,\ldots \\ x(k(\sigma+1)) & if\ \ k(\sigma+1) \le t < (k+1)(\sigma+1),\ \ k = 0,1,2,\ldots \end{cases}. \quad (12)$$

4 Being the Leader

Let us consider the behavior of the best particle. For this particle personal and global best positions are the same $p_G(t) = p(t)$, although its' actual position may be different. If the particle is not going to improve its' fitness: $\Delta p_G(t) = 0$, the movement of the particle is governed by

$$\begin{bmatrix} v(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} \chi & \chi\varphi(t) \\ -\chi & 1-\chi\varphi(t) \end{bmatrix} \begin{bmatrix} v(t) \\ y(t) \end{bmatrix}. \quad (13)$$

Calculation of expected values of $v(t), y(t)$ leads to a very similar equation

$$\begin{bmatrix} E[v(t+1)] \\ E[y(t+1)] \end{bmatrix} = \begin{bmatrix} \chi & \chi\bar{\varphi} \\ -\chi & 1-\chi\bar{\varphi} \end{bmatrix} \begin{bmatrix} E[v(t)] \\ E[y(t)] \end{bmatrix} \quad \bar{\varphi} = \frac{1}{2}(c_1 + c_2). \quad (14)$$

The eigenvalues $\mathrm{z}_{1,2}(\varphi, \chi)$ of matrix

$$M(\varphi, \chi) = \begin{bmatrix} \chi & \chi\varphi \\ -\chi & 1-\chi\varphi \end{bmatrix}. \quad (15)$$

were studied in many previous works [2,4]. Let us mention for the completeness, that the eigenvalues are inside the unit circle for the parameters bounded by inequalities

$$\chi > 0,\ \ \varphi > 0\ \ \chi < \begin{cases} 1 & \text{dla } \varphi < 2 \\ \frac{2}{\varphi-2} & \text{dla } \varphi > 2 \end{cases} \quad (16)$$

and will be complex with $\left|\mathrm{z}_{1,2}\right| = \sqrt{\chi}$ if

$$\frac{\varphi+1-2\sqrt{\varphi}}{(\varphi-1)^2} < \chi < \frac{\varphi+1+2\sqrt{\varphi}}{(\varphi-1)^2}. \quad (17)$$

Two particular cases are interesting: if $\varphi = \varphi_{max} = c_1 + c_2$ - maximal possible value for random variable $\varphi(t)$ and if $\varphi = \bar{\varphi} = 0.5(c_1 + c_2)$ - this value arrives in equation (14) describing expected value behavior. For parameters' values (3) the eigenvalues are:

$$\mathrm{z}_{1,2}(4.1,0.729) = -0.6298 \pm 0.5764i,\ \ \mathrm{z}_{1,2}(2.05,0.729) = 0.1173 \pm 0.8457i. \quad (18)$$

Figure 2 illustrates inequalities (16) and (17). Typical particle behavior and expected values (theoretical and empirical) were plotted in figure 3.

As we see, for parameters' values (3), the best particle will be attracted rather fast to its best position, governed by (13), or in the sense of expected values convergence by (14). Only during several first iterations the leader is able to explore the space around the best position.

If the leader is at its best position at iteration t, the next move will be described by

$$\begin{bmatrix} v_b(t+1) \\ y_b(t+1) \end{bmatrix} = \begin{bmatrix} \chi \\ -\chi \end{bmatrix} v_b(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta p_G(t),\ \Delta p_G(t) = v_b(t+1),\ y_b(t+1) = 0 \quad (19)$$

(where index b denotes this particular particle), so, if this situation (improvement of the personal best and remaining the global best) will repeat at next iterations, the velocity will be constricted by χ leading to the smaller and smaller steps. Therefore, also in this situation the leader gives up its exploration and almost freezes.

The well known conclusion from this analysis is that the persistent leader does not contribute to swarm effectiveness and that exploration ability of the swarm follows from the skill of cooperation and changing the leader during solving minimization problem.

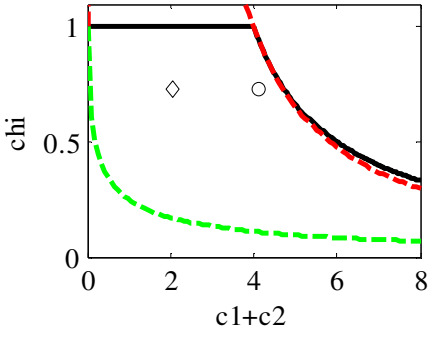


Fig. 2. Matrix $M(\varphi, \chi)$ eigenvalues distribution. Region of stability (*under solid lines*), region of complex eigenvalues (between dashed lines), parameters $(\varphi = 4.1, \chi = 0.729)$ (*circle*) and $(\varphi = 4.1,\ \chi = 0.729)$ (*diamond*).

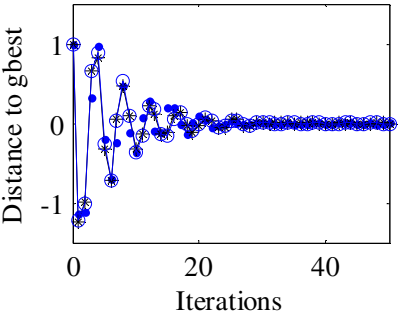


Fig. 3. Typical particle behavior (*dots*) under stochastic equation (13), empirical expected values (*stars*) calculated from 500 runs, expected values calculated from equation (14) (*circles*). All calculations done for parameters values (3) and initial position $v(0)=1$, $y(0)=1$.

5 Following the Leader

5.1 Sedulous Particle

Let us consider a particle arriving at its personal best position at iteration t : $p(t) = x(t), p_G(t) - p(t) = y(t)$. It follows from (8) that the next move of this particle will be described by:

$$\begin{bmatrix} v(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} \chi & \chi c_2(t) \\ -\chi & 1 - \chi c_2(t) \end{bmatrix} \begin{bmatrix} v(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta p_G(t). \quad (20)$$

If a particle is improving its' position every next iteration, it will be called a sedulous particle. Dynamics of the sedulous particle is described by (20), or in a sense of mean values by

$$\begin{bmatrix} E[v(t+1)] \\ E[y(t+1)] \end{bmatrix} = \begin{bmatrix} \chi & \chi\frac{c_2}{2} \\ -\chi & 1-\chi\frac{c_2}{2} \end{bmatrix} \begin{bmatrix} E[v(t)] \\ E[y(t)] \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} E[\Delta p_G(t)]. \quad (21)$$

Therefore again the matrix $M(\varphi,\chi)$ (15) decides on particle dynamics, but now $\varphi = c_2(t)$ (in 20) or $\varphi = 0.5c_2$ (in 21). It is easy to show, that if the difference of the global best position converges to a constant: $\lim_{t\to\infty} \Delta p_G(t) = \delta$, the sedulous particle speed and position mean values will converge to

$$\lim_{t\to\infty} \begin{bmatrix} E[v(t)] \\ E[y(t)] \end{bmatrix} = \begin{bmatrix} \delta \\ \frac{2\delta(1-\chi)}{\chi c_2} \end{bmatrix} \quad (22)$$

If the sedulous particle follows the global/local best particle, whose movement is governed by equation (19), the resulting dynamics will be

$$\begin{bmatrix} v(t+1) \\ y(t+1) \\ v_b(t+1) \end{bmatrix} = \begin{bmatrix} \chi & \chi c_2(t) & 0 \\ -\chi & 1-\chi c_2(t) & \chi \\ 0 & 0 & \chi \end{bmatrix} \begin{bmatrix} v(t) \\ y(t) \\ v_b(t) \end{bmatrix}, \quad (23)$$

or

$$\begin{bmatrix} E[v(t+1)] \\ E[y(t+1)] \\ v_b(t+1) \end{bmatrix} = \begin{bmatrix} \chi & \chi\frac{c_2}{2} & 0 \\ -\chi & 1-\chi\frac{c_2}{2} & \chi \\ 0 & 0 & \chi \end{bmatrix} \begin{bmatrix} E[v(t)] \\ E[y(t)] \\ v_b(t) \end{bmatrix}, \quad (24)$$

so again eigenvalues of matrix $M(\varphi,\chi)$ (15) decide how fast the sedulous particle position will converge to the best particle position.

If $\Delta p_G(t)$ is increasing or oscillating, the distance from the sedulous particle position to the best particle position $y(t)$ will increase or oscillate also.

Equations (20, 21) may be used also to derive dynamics of higher moments of random variables $y(t)$, $v(t)$. It is necessary to stress that, for example, standard deviation convergence does not result from expected value convergence.

Figure 4 illustrates some typical behaviors of a sedulous particle.

5.2 Lazy Particle

Let us consider a particle arriving at its personal best position at iteration *t*, and assume that the next improvement of personal misfit will happen at time $t+\sigma+1$, so

$$p(t) = x(t), p(t+1) = x(t),, p(t+\sigma) = x(t), p(t+\sigma+1) = x(t+\sigma+1t) \quad (25)$$

We will call this particle σ-lazy particle, so a sedulous particle is a 0-lazy particle. The following description of σ-lazy particle dynamics follows from equation (8):

$$\begin{bmatrix} v(t+\sigma+1) \\ y(t+\sigma+1) \end{bmatrix} = W_\sigma \begin{bmatrix} v(t) \\ y(t) \end{bmatrix} + N_\sigma\big(p_G(t), \dots, p_G(t+\sigma+1)\big) \quad (26)$$

where W_σ and N_σ are defined by the following recurrence ($M(t)$ is defined in (9)):

$$W_0 = \begin{bmatrix} \chi & \chi c_2(t) \\ -\chi & 1-\chi c_2(t) \end{bmatrix}, \qquad W_\sigma = M(t+\sigma)W_{\sigma-1} + \begin{bmatrix} 0 & -\chi c_1(t+\sigma) \\ 0 & \chi c_1(t+\sigma) \end{bmatrix}, \quad (27)$$

$$N_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta p_G(t), \quad N_\sigma = M(t+\sigma)N_{\sigma-1} + \begin{bmatrix} -\chi c_1(t+\sigma) & 0 \\ \chi c_1(t+\sigma) & 1 \end{bmatrix} \begin{bmatrix} \Delta_\sigma p_G(t) \\ \Delta p_G(t+\sigma) \end{bmatrix}, \quad (28)$$

$$\Delta_\sigma p_G(t) = p_G(t+\sigma) - p_G(t). \quad (29)$$

$$\begin{bmatrix} E[v(t+\sigma+1)] \\ E[y(t+\sigma+1)] \end{bmatrix} = EW_\sigma \begin{bmatrix} E[v(t)] \\ E[y(t)] \end{bmatrix} + EN_\sigma\big(p_G(t), \dots, p_G(t+\sigma+1)\big) \quad (30)$$

$$EW_0 = \begin{bmatrix} \chi & \chi\frac{c_2}{2} \\ -\chi & 1-\chi\frac{c_2}{2} \end{bmatrix}, \qquad EW_\sigma = M(\bar{\varphi},\chi)(EW_{\sigma-1}) + \begin{bmatrix} 0 & -\chi\frac{c_1}{2} \\ 0 & \chi\frac{c_1}{2} \end{bmatrix}, \quad (31)$$

$$EN_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} E[\Delta p_G(t)], \quad EN_\sigma = M(\bar{\varphi},\chi)EN_{\sigma-1} + \begin{bmatrix} -\chi\frac{c_1}{2} & 0 \\ \chi\frac{c_1}{2} & 1 \end{bmatrix} \begin{bmatrix} E[\Delta_\sigma p_G(t)] \\ E[\Delta p_G(t+\sigma)] \end{bmatrix}, \quad (32)$$

($M(\varphi,\chi)$ was defined by (15) and $\bar{\varphi} = 0.5(c_1 + c_2)$).

Matrix $M(\varphi,\chi)$ is important for σ-lazy particle dynamics, but now it acts indirectly, influencing (27) or (31). Particle dynamics depends on W_σ, or in sense of mean value, on EW_σ. If we investigate these matrices for constant cognitive and social parameters we find from (32) that

$$\lim_{\sigma\to\infty} EW_\sigma = \lim_{\sigma\to\infty} W_\sigma = \begin{bmatrix} 0 & 0 \\ 0 & \frac{c_2}{c_1+c_2} \end{bmatrix} \quad (33)$$

Therefore swarm parameters should be chosen to obtain desired eigenvalues of W_σ for moderate, expected values of σ, rather than for eigenvalues of W_0. For example eigenvalues of EW_σ for parameters (3) are collected in table 1.

Table 1. Eigenvalues of EW_σ for parameters (3), $m = \max\{abs[eig(EW_\sigma)]\}$.

σ	0	1	2	4	5	6
eig(EW_σ)	0.4909 $\pm$ 0.6992i	-0.5083 0.0262	-0.4133 0.3322	0.3865 0.6945	0.5299 $\pm$ 0.3071i	0.1034 $\pm$ 0.1896i
m	0.8543	0.5083	0.4133	0.6945	0.6125	0.2160

As we notice matrices EW_σ will demonstrate significantly different properties than EW_0 - for some σ eigenvalues are real and absolute value of egenvalues is decreasing quite fast with increasing σ. It means that if a particle remains σ-lazy for bigger σ it will stagnate very fast. In figure 5 we demonstrate typical behavior of a σ-lazy particle in case of constant global/local best position.

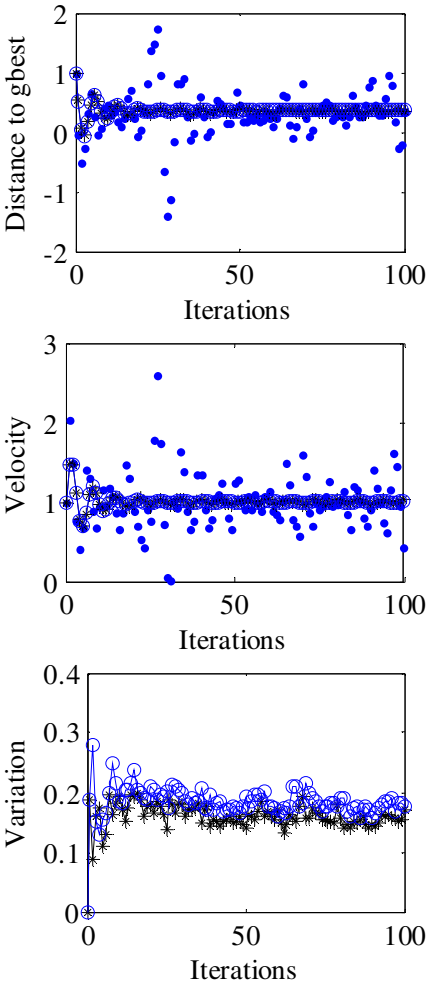


Fig. 4. Typical sedulous particle behavior (*dots*) under stochastic equation (20). Empirical expected values (*stars*) calculated from 1000 runs, expected values calculated from equation (21) (*circles*). Empirical variation of velocity (*stars*) and distance (*circles*). All calculations done for $\Delta p_G(t) = 1$, parameters values (3) , and initial position $v(0)=1$, $y(0)=1$.

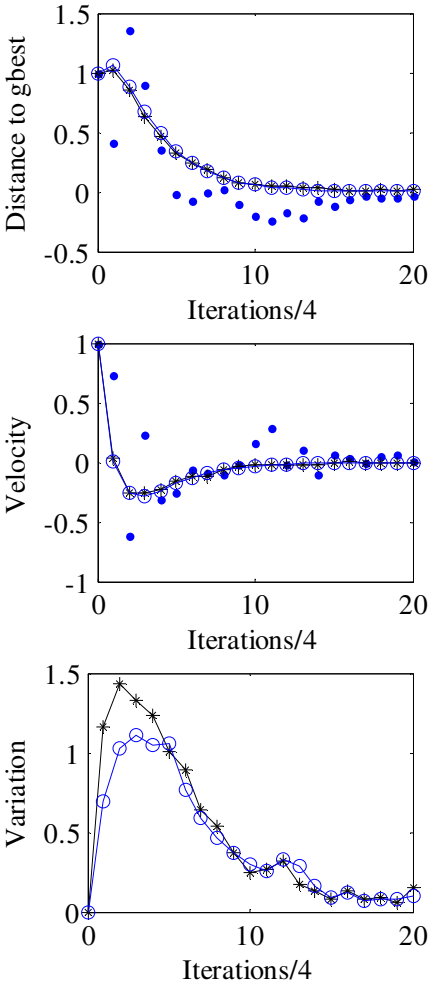


Fig. 5. Typical 3-lazy particle behavior (*dots*) under stochastic equation (26). Empirical expected values (*stars*) calculated from 1000 runs, expected values calculated from equation (30) (*circles*). Empirical variation of velocity (*stars*) and distance (*circles*). All calculations done for $\Delta p_G(t) = 0$, parameters values (3) , and initial position $v(0)=1$, $y(0)=1$.

6 Conclusions

We have investigated the particle dynamics in case of moving personal best and global best positions. The derived equations describe single particle dynamics and

expected values behavior in multi-run experiment. The analysis may be used to explain the role of the PSO parameters through algorithm execution.

In contrast to previous research we stress different influence of cognitive and social acceleration. We propose three qualitative labels for particles: the leader, the sedulous follower and the σ-lazy follower. These features may be easily monitored for each particle. Frequent change of the leader is crucial for swarm exploration and effectiveness of optimization. Therefore, if the prolonged leadership is detected, we may react with speed modification.

According to the presented analysis, it is also obvious, that if a particle 'is lazy' for a period σ longer than several iterations, it will not contribute to the swarm success, so we have to react and interrupt 'laziness' modifying the particle's speed or position.

It is important to stress that properties of a σ -lazy follower dynamics are different than a sedulous particle dynamics (for the same swarm parameters). A sedulous particle is able to combine an active exploration of the search-space with following the leader and the exploration process is more efficient if the leader is moving.

The above presented conclusions stimulate parameter choice and adaptation techniques that should be tested in numerical experiments.

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. 1995 IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
2. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evolutionary Computation 6(1), 58–73 (2002)
3. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85, 317–325 (2003)
4. Clerc M.: Particle Swarm Optimization, ISTE Ltd (2006)
5. Engelbrect, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons Ltd., Chichester (2005)
6. Kadirkamanathan, V., Selvarajah, K., Fleming, P.J.: Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Trans. Evolutionary Computation 10(3), 245–255 (2006)
7. Wakasa, Y., Tanaka, K., Nishimura, Y.: Control-Theoretic Analysis of Exploitation and Exploration of the PSO Algorithm. In: 2010 IEEE International Symposium on Computer-Aided Control System Design, Part of 2010 IEEE Multi-Conference on Systems and Control, Yokohama, Japan, September 8-10, pp. 1807–1812 (2010)
8. Clerc, M.: Stagnation analysis in particle swarm optimization or what happens when nothing happens, Dept. Comput. Sci., Univ. Essex, Colchester, U.K., Tech. Rep. CSM-460 (2006)
9. Fernandez-Martinez, J.L., Garcia-Gonzalo, E.: Stochastic Stability Analysis of the Linear Continuous and Discrete PSO Models. IEEE Trans. Evolutionary Computation (to appear, 2011), doi:10.1109/TEVC.2010.2053935

An Adaptive Discretization in the ACDT Algorithm for Continuous Attributes

Urszula Boryczka and Jan Kozak

Institute of Computer Science, University of Silesia, Będzińska 39,
41–200 Sosnowiec, Poland
Tel.:/Fax: (+48 32) 291 82 83
{urszula.boryczka,jan.kozak}@us.edu.pl

Abstract. Decision tree induction has been widely used to generate classifiers from training data through a process of recursively splitting the data space. In the case of training on continuous-valued data, the associated attributes must be discretized in advance or during the learning process. The commonly used method is to partition the attribute range into two or several intervals using single or a set of cut points. One inherent disadvantage in these methods is that the use of sharp cut points makes the induced decision trees sensitive to noise. To overcome this problem this paper presents an alternative method called adaptive discretization based on Ant Colony Decision Tree (ACDT) approach. Experimental results showed that, by using that methodology, better classification accuracy has been obtained in both training and testing data sets in majority of cases concerning the classical decision tree constructed by ants. It suggests that the robustness of decision trees could be improved by means of this approach.

Keywords: ACDT, Ant-Miner, Data Mining, Ant Colony Decision Tree, Ant Colony Optimization.

1 Introduction

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making. A decision tree induction is a very popular technique in machine learning. It has a lot of beneficial aspects in comparison to other machine learning methods like, e.g., computational complexity, high accuracy and readability of data. A decision tree can be used to classify an example by starting at a root of the tree and moving through it until a leaf node, which provides the classification of the instance. The decision tree induction is a typical inductive approach to learn knowledge on classification. The decision tree induction algorithms were firstly introduced in 60's and as a general scheme of recursive partitioning is nowadays a quite attractive approach and still worth further developments [5,12,14].

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 475–484, 2011.

There exist a plenty of different algorithms for decision tree induction and their implementations. All the approaches described here were implemented as variants of Ant Colony Optimization in data mining tasks. The most popular of them is the Ant-Miner algorithm. This approach firstly proposed by Parpinelli at al. applies an Ant Colony Optimization heuristic [6,7] to the classification task of data mining to discover an ordered list of classification rules [10,11]. The Ant-Miner algorithm was modified several times and deeply analyzed. Many of these modifications are described in [3]. In this article the population size of ants is discussed, new pheromone updating rules, exploration and exploitation rate and new pruning procedures and heuristic functions are tested. The heuristic function is a crucial point in this approach, so some extensions have also been proposed, mainly based on algorithms for construction decision trees such as CART [2]. In order to improve the processing time of the Ant-Miner algorithm the parallel Ant-Miner algorithm is proposed and analyzed [4]. The classical algorithm proposed by Parpinelli was not adapted to the continuous attributes, but some augmentations or extensions to different Ant-Miner versions of algorithms are applied for example the cAnt-Miner algorithm, cope with continuous attributes during the rule construction process [8,9,15]. The ACDT algorithm presented in this paper is an approach which is a different approach when compared it to the predecessor with respect to the decision tree construction. This approach is not suitable to discover list of rules. The ACDT algorithm, firstly proposed in 2010 [1] is a new technique to construct decision trees. The original version undergoing the twoing criterion or another splitting rule did not cope with the continuous attributes.

Our motivation is to show a new method of coping with the continuous attributes during the process of construction the decision tree. In the context of real-world classification problems described by nominal as well as continuous attributes it is important to overcome this problem because of appropriate interpretation these values. In order to improve our method we incorporate new splitting rule for better analyzing this real-world data sets and it seems to be one of proper ways.

In the present thesis we would like to discuss a series of important sections. The first section is devoted to an introduction to the subject of this paper. Section 2 describes Ant Colony Decision Tree approach, especially the splitting rules. Section 4 focuses on the presented new version of the algorithm cACDT. Section 5 presents the experimental study that has been conducted to evaluate the performance of cACDT, taking into consideration eleven data sets. The last section concludes obtained results and discusses the future evolution of the presented approach.

2 Ant Colony Decision Trees

In the context of decision tree construction or discovering classification rules in data mining, Ant Colony Optimization (ACO) approaches have been widely applied to different tasks. In essence, Ant Colony Decision Trees (ACDT) is a first approach for constructing decision trees and is competitive with the well known CART algorithm.

In ACDT each ant chooses the appropriate attribute for splitting in each node of the constructed decision tree according to the heuristic function and pheromone values (fig. 1). The heuristic function is based on the Twoing criterion, which helps ants divide the objects into two groups, connected with the analyzed attribute values. In this way, the attribute, which well separate the objects is treated as the best condition for the analyzed node. The best splitting is observed when we classified the same number of objects in the left and right subtrees with the maximum homogenity in the decision classes. Pheromone values represent the best way (connection) from the superior to the subordinate nodes – all possible combinations in the analyzed subtrees. For each node we calculate the following values according to the objects classified using the Twoing criterion of the superior node.

A decision tree is built in accordance with splitting rule that performs the multiple splitting of learning sample into smaller parts. Data in each node have to be divided into two parts with maximum homogeneity in the decision class.

Twoing criterion will search for two classes that will make up together more then 50% of the data. Twoing splitting rule will maximize the following change-of-impurity measure which implies the following maximization problem for nodes m_l, m_r:

$$\underset{a_j \leq a_j^R, j=1,\ldots,M}{\arg\max} \left(\frac{P_l P_r}{4} \left[\sum_{k=1}^{K} |p(k|m_l) - p(k|m_r)| \right]^2 \right), \tag{1}$$

where:

$p(k|m_l)$ – the conditional probability of the class k provided in node m_l,
P_l – the probability of transition objects into the left node m_l,
P_r – the probability of transition objects into the right node m_r,
K – number of decision class,
a_j – j–th variable,
a_j^R– the best splitting value of variable a_j.

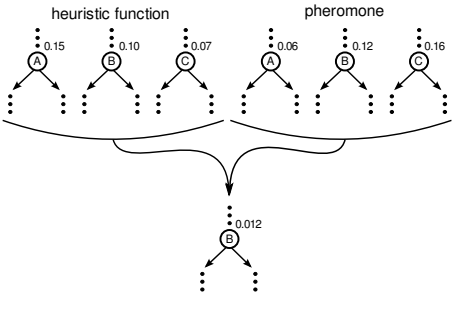


Fig. 1. Choice of splits in ACDT

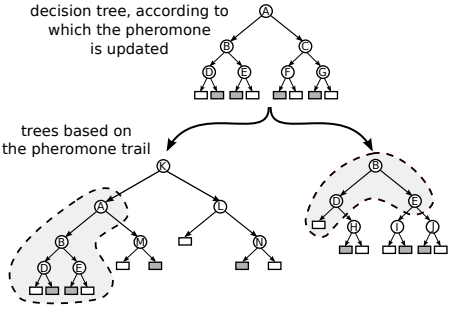


Fig. 2. Building the decision tree with pheromone

Although Twoing splitting rule allows us to build more balanced trees, this algorithm works slower than the Gini rule. For example, if the total number of classes is equal to K, then we will have 2^{K-1} possible splits.

The pseudo-code of the proposed algorithm is presented below. Lines 2–13 describe one iteration of this algorithm. At the beginning of its work, each ant builds one decision tree (lines 4–11). At the end of the loop, the best decision tree is chosen and then the pheromone is updated according to the splits performed during the process of construction the decision tree, iteratively. While constructing the tree, agents–ants are analyzing previous structures and some modifications are performed in the single node. This process is performed till the best decision tree is obtained. The process of building the decision tree is presented in Figure 2.

Algorithm 1. Pseudo-code of the ACDT algorithm

```
1  initialization_pheromone_trail(pheromone);
2  for number_of_iterations do
3      best_tree = NULL;
4      for number_of_ants do
5          new_tree = build_tree(pheromone);
6          pruning(new_tree);
7          assessment_of_the_quality_tree(new_tree);
8          if new_tree is_higher_quality_than best_tree then
9              best_tree = new_tree;
10         endIf
11     endFor
12     update_pheromone_trail(best_tree, pheromone);
13 endFor
14 result = best_constructed_tree;
```

The value of the heuristic function is determined according to the splitting rule employed in CART approach (see formula (1)), in dependence on the chosen criterion. Whereas the probability of choosing the appropriate test in the node is calculated according to a classical probability used in ACO:

$$p_{i,j} = \frac{\tau_{m,m_{L(i,j)}}(t)^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{i}^{a} \sum_{j}^{b_i} \tau_{m,m_{L(i,j)}}(t)^{\alpha} \cdot \eta_{i,j}^{\beta}} \tag{2}$$

where:

$\eta_{i,j}$ – a heuristic value for the test of the attribute i and value j,

$\tau_{m,m_{L(i,j)}}$ – an amount of pheromone currently available at time t on the connection between nodes m and m_L, (it concerns the attribute i and value j),

α, β – the relative importance with experimentally established values 1 i 3.

The initial value of the pheromone trail, similarly to the Ant–Miner approach is established in dependence on the number of attribute values. While

the pheromone updates are performed (3) by increasing the previous values on each pairs of nodes (parent–child):

$$\tau_{m,m_L}(t+1) = (1-\gamma) \cdot \tau_{m,m_L}(t) + Q(T) \tag{3}$$

where $Q(T)$ determines the evaluation function of decision tree (see formula (4)), and γ is a parameter representing the evaporation rate, equal to 0.1.The evaluation function for decision trees will be calculated according to the following formula:

$$Q(T) = \phi \cdot w(T) + \psi \cdot a(T,P) \tag{4}$$

where:
$w(T)$ – the size (number of nodes) of the decision tree T,
$a(T,P)$ – the accuracy of the classification object from a test set P by the tree T,
ϕ and ψ – constants determining the relative importance of $w(T)$ and $a(T,P)$.

3 ACDT Algorithm for Continuous Attributes

Most empirical learning systems are given a set of pre-classified cases, each described by a vector of attribute values, and construct from them a mapping from attribute values to classes. The attributes used to describe cases can be grouped into continuous attributes, which values are real numbers, and discrete attributes with unordered nominal values [14].

As mentioned in the introduction, the previous version of the Ant Colony Decision Tree approach does not cope with continuous attributes directly. It requires continuous attributes to be discretized in a preprocessing step. In classical version of the ACDT algorithm splitting the data due to the default equality (equivalence) test (5) or adjunction test (6) applying to the discrete attributes (fig. 3).

$$T_e(x) = \begin{cases} 1 & \text{, if } a_j(x) = v \\ 0 & \text{, otherwise} \end{cases} \tag{5}$$

$$T_a(x) = \begin{cases} 1 & \text{, if } a_j(x) \in V \\ 0 & \text{, otherwise} \end{cases}, \tag{6}$$

where:
T – a decision tree,
x – an object,
v – attribute value,
V – set of the attribute values,
a_j – a conditional attribute.

Many researchers have recently noted that the equality or adjunction rules in the decision tree construction are weaker in domains with a preponderance of continuous attributes than for learning tasks that have mainly discrete attributes. Discussing the effect of replacing continuous attributes by discrete attributes,

it can be observed that this approach causes limitation of the prediction new cases [8,9,13].

The proposed modification of the ACDT algorithm coping with continuous attributes incorporates an inequality test (7). For discrete attributes we still perform the default adjunction test in this new version of ACDT:

$$T(x) = \begin{cases} 1 & \text{, if } a_j(x) \leq c \\ 0 & \text{, otherwise } (a_j(x) > c) \end{cases}, \tag{7}$$

where c is a threshold value (cut-off point).
The effects of these changes influence on the precision and prediction of splitting the continuous values. The observations are as follows: the cutting points are performed according to the values occurring in the analyzed attributes in the testing set. It can be assessed empirically in series of experiments with a substantial number of learning tasks. For attribute a_j each cutting point is performed due to the concrete value derived from the testing set and finally is evaluated according to the chosen criterion (e.g. Twoing criterion (1)).

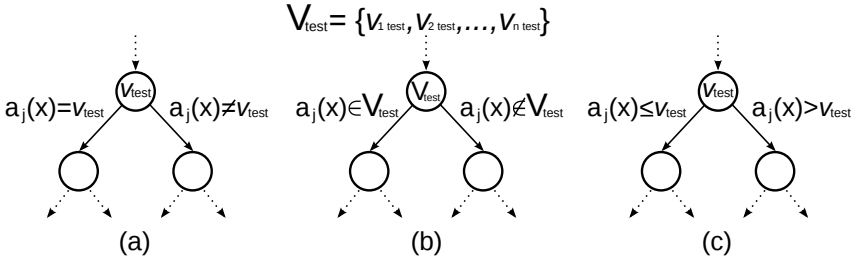


Fig. 3. Tests on the attributes of the ACDT and cACDT algorithm: (a) equality (equivalence) test, (b) adjunction test, (c) inequality test

4 Experiments

A variety of experiments were conducted to test the performance and behavior of the proposed algorithm. First we describe our experimental methodology and explain its motivation. Then we present and discuss our results. In this section we will consider an experimental study (see Table 2 and 3) performed for the following adjustments. We have performed 100 experiments for each data set. Each experiment included 25 generations with the population size of ant colony equal to 10.

For this purpose, a post–pruning phase was applied to the decision trees that were generated by ACDT to form the lookahead sample, and the size of the pruned decision trees will be considered. Previous comparative studies did not find a single pruning method that is generally the best and conclude that different pruning techniques behave similarly. Therefore, during the analysis we used the Twoing criterion and Error–Based Pruning, and examine post–pruning of the final trees. We have performed the experimental study for two approaches:

ACDT – with quality coefficient calculated accordingly to with the learning set, when the algorithm is evaluated based on the testing set. We also used the Twoing criterion. We evaluated the effectiveness of the approach and then updated the pheromone values for the best quality decision tree. The splitting was performed in each node accordingly to with the adjunction test.

cACDT (ACDT with continuous attributes) – with quality coefficient calculated accordingly to with the learning set, when the algorithm is evaluated based on the testing set. We also used the Twoing criterion. We evaluate the effectiveness of the approach and then update the pheromone values for the best quality decision tree. The splitting in each node is performed accordingly to the adjunction test, and for continuous attributes we apply inequality test.

4.1 Data Sets

Evaluation of the performance behavior of ACDT was performed using 11 public–domain data sets from the UCI (University of California at Irvine) data set repository available from at: `http://archive.ics.uci.edu/ml/`. Table 1 shows the main characteristics of the data sets, which are divided into two groups in a random way: training ($\frac{2}{3}$ of all objects) and testing ($\frac{1}{3}$ of all objects) sets, appropriately. In order to obtain reliable performance estimates train–and–test were carried out to produce each of the statistics presented in the tables (see below). The experiments were carried out on the Intel Celeron 2.27 GHz Computer with 3 GB RAM.

Table 1. Original parameters in data sets

Dataset	Number of instances	Number of attribute		Decision Class
		nominal	continuous	
australian	690	8	6	2
bands	512	19	20	2
bupa	345	0	6	2
crx	690	9	6	2
ecoli	336	0	7	8
glass	214	0	9	7
hepatitis	155	13	6	2
ionosphere	351	0	34	2
iris	150	0	4	3
spam	4601	0	57	2
yeast	1484	0	8	10

4.2 Results

Results of experiments prove that dynamic, adaptive discretization relies on the inequality test increase the classification accuracy of the constucted decision trees.

Table 2. Comparative study accuracy rate (standard deviations in parentheses)

Dataset	ACDT algorithm		cACDT algorithm		Ant-Miner algorithm [8]	cAnt-Miner algorithm [8]
australian	0.8650	(0.0013)	0.8603	(0.0083)	0.8552	0.8660
bands	0.6813	(0.0294)	**0.7370**	(0.0374)	—	—
bupa	0.5827	(0.0218)	0.5843	(0.0324)	—	—
crx	0.8565	(0.0000)	**0.8817**	(0.0112)	0.8532	0.8556
ecoli	0.4231	(0.0114)	**0.9505**	(0.0174)	—	—
glass	0.4940	(0.0289)	**0.8615**	(0.0269)	0.5148	0.6569
hepatitis	0.7107	(0.0283)	**0.7844**	(0.0280)	0.7461	0.8489
ionosphere	**0.8637**	(0.0136)	0.8495	(0.0213)	0.9068	0.9000
iris	0.7506	(0.0604)	**0.8984**	(0.0123)	—	—
spam	0.9178	(0.0040)	**0.9393**	(0.0087)	—	—
yeast	0.4092	(0.0138)	**0.4341**	(0.0167)	—	—

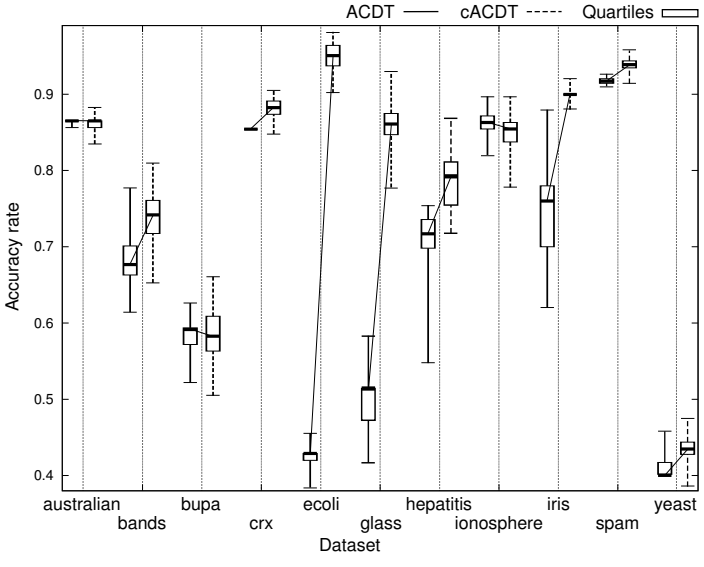


Fig. 4. Results (accuracy rate)

Results of these trials, summarized in tab. 2 show the accuracy of the classification for two approaches. The best results are presented in bold. This approach leads to improved accuracy on eight of the tasks (about 50%). Most of the improvements are significant - evidence of this situation is provided by fig. 4.

Please note, that for data sets with big number of decision classes (ecoli, glass and yeast), classical version of our approach (ACDT) obtains really low values of classification accuracy. The result confirms the clear trend which shows that the adaptive discretization degrades the performance of the presented approach

more as data set become larger, but can be beneficial for data sets with smaller clases. In general, it is finished after the only one split of data (the tree consists on 1 node), so it is unprofitably. The results of the performance of cAnt-Miner [8] highlights the promising effects of the proposed modification, leading to not very large trees and the structure of the tree is clarified. The above-mentioned results are depicted in Table 3. It is worth noting that it is natural.

The accuracy of cACDT approach is rivaled or surpassed the ACDT on the most of the datasets. The results show that the straightforward changes concerning the continuous attributes lead to an overall improvement in its performance on the five testable data sets. Only two data sets are significantly difficult during the analysis. The same dependence could be observed in the case of Ant-Miner and cAnt-Miner [8]. In this situation we highlight the fact of considerable improvement of the performance of our proposition. Statistical (The Wilcoxon Two Sample Test) evidence confirms that cACDT is better than ACDT algorithm.

Table 3. Comparative study number of nodes (standard deviations in parentheses)

Dataset	ACDT algorithm		cACDT algorithm	
australian	1.15	(0.87)	8.45	(6.54)
bands	10.44	(2.77)	18.27	(4.58)
bupa	4.56	(1.19)	10.53	(3.13)
crx	1.00	(0.00)	10.76	(2.50)
ecoli	3.64	(1.14)	7.04	(0.99)
glass	4.92	(1.23)	6.80	(1.24)
hepatitis	4.07	(1.17)	5.22	(1.89)
ionosphere	3.58	(1.11)	10.62	(2.60)
iris	8.87	(1.76)	3.20	(0.40)
spam	53.73	(12.43)	48.77	(7.82)
yeast	4.56	(5.47)	37.74	(11.40)

5 Conclusions

This study has focused on the extension to ACDT, named cACDT, which copes with continuous attributes during the decision trees construction. Our approach does not require a discretization method in a preprocessing step. The investigation of the variety of approaches concerning the splitting rule applied in ACDT and cACDT on several data sets has provided evidence for the following proposals. In test evaluation, the decrease of the number of levels the decision trees does not influence on the quality of this classification. Approaches that ignore the continuous attributes (ACDT) perform badly and lead to very inferior performance.

Our proposition cACDT has been compared against the predecessor of the ACDT with respect to predictive accuracy and the structure of the tree. Regarding the predictive accuracy Ant Colony Decision Tree with adaptive discretization - cACDT significantly outperformed the previous version. Please note, that

this modification is not complicated and rather intuitive. Therefore the results obtained by cACDT are promising and worth further analysis.

Really good performance of cACDT with simple modification leads to the next application concerning regresion. As future research direction, it would be interesting to extend this approach to allow the creation a specific analysis for decision attribute with continuous values.

References

1. Boryczka, U., Kozak, J.: Ant colony decision trees – a new method for constructing decision trees based on ant colony optimization. In: Pan, J.S., Chen, S.M., Nguyen, N. (eds.) ICCCI 2010. LNCS, vol. 6421, pp. 373–382. Springer, Heidelberg (2010)
2. Boryczka, U., Kozak, J.: A New Heuristic Function in Ant–Miner Approach. In: ICEIS 2009, Milan, Italy, pp. 33–38 (2009)
3. Boryczka, U., Kozak, J.: New Algorithms for Generation Decision Trees – Ant–Miner and Its Modifications, pp. 229–264. Springer, Berlin (2009)
4. Boryczka, U., Kozak, J., Skinderowicz, R.: Parellel Ant–Miner. Parellel implementation of an ACO techniques to discover classification rules with OpenMP. In: MENDEL 2009, pp. 197–205. University of Technology, Brno (2009)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman & Hall, New York (1984)
6. Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative learning approach to the Traveling Salesman Problem. IEEE Tr. Evol. Comp. 1, 53–66 (1997)
7. Dorigo, M., Birattari, M., Stützle, T., Libre, U., Bruxelles, D., Roosevelt, A.F.D.: Ant colony optimization – artificial ants as a computational intelligence technique. IEEE Comput. Intell. Mag. 1, 28–39 (2006)
8. Otero, F., Freitas, A., Johnson, C.: cAnt-Miner: An ant colony classification algorithm to cope with continuous attributes. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 48–59. Springer, Heidelberg (2008)
9. Otero, F.E.B., Freitas, A.A., Johnson, C.G.: Handling continuous attributes in ant colony classification algorithms. In: CIDM, pp. 225–231 (2009)
10. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: An ant colony algorithm for classification rule discovery. In: Abbas, H., Sarker, R., Newton, C. (eds.) Data Mining: a Heuristic Approach, Idea Group Publishing, London (2002)
11. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation, Special issue on Ant Colony Algorithms, 321–332 (2004)
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
13. Quinlan, J.R.: Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
14. Rokach, L., Maimon, O.: Data Mining With Decision Trees: Theory and Applications. World Scientific Publishing, Singapore (2008)
15. Schaefer, G.: Ant colony optimisation classification for gene expression data analysis. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 463–469. Springer, Heidelberg (2009)

Approximate Nash Equilibria in Bimatrix Games

Urszula Boryczka and Przemyslaw Juszczuk

Institute of Computer Science, University of Silesia,
ul.Bedzinska 39, Sosnowiec, Poland
{urszula.boryczka,przemyslaw.juszczuk}@us.edu.pl

Abstract. Nash equilibrium is one of the main concepts in the game theory. Recently it was shown, that problem of finding Nash equilibrium and an approximate Nash equilibrium is PPAD-complete. In this article we adapt Differential Evolution algorithm (DE) to the above problem. It may be classified as continuous problem, where two probability distributions over the set of pure strategies of both players should be found. Every deviation from the global optimum is interpreted as Nash approximation and called ϵ-Nash equilibrium. We show, that the Differential Evolution approach can be determined as iterative method, which in successive iterations is capable to obtain ϵ value close to the global optimum. The contribution of this paper is the experimental analysis of the proposed approach and indication of it's strong features. We try to demonstrate, that the proposed method is very good alternative for the existing mathematical analysis of the mentioned Nash equilibrium problem.

Keywords: Game theory, ϵ-Nash equilibrium, Differential evolution.

1 Introduction

Many social or economical situations have in common that there are a number of decision makers (also called players) with various objectives involed, who for some reason do not cooperate and the outcome depends only on the actions chosen of the different decision makers. This description of the problem corresponds to the n-player noncooperative games. The main problem in all game theory (not only normal form games, which we consider) is finding Nash equilibrium.

In this article we study 2-player non-zero sum games in normal form with normalized payoff matrix, where two payoff matrices are chosen randomly. Normal form games and Nash equilibria have important applications in many fields [17,7,15,13]. Determining whether Nash equilibria exist, and effectively computing them, are relevant problems that have attracted much research in computer science [5,12]. Finding simple, pure Nash equilibrium (in contrast to the mixed Nash equilibrium) is easy problem. Nash equilibrium may be described as a set of strategies for both players, that no player could achieve a higher payoff than actual. Also zero-sum games, which may be interpreted as a special case of non-zero sum games may be successfully solved by the pivot method and by the simplex method (both methods are part of the linear programming). The

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 485–494, 2011.

vast majority of the scientific work has been dedicated to the problem of finding mixed Nash equilibria. At this moment, it is clear, that computing mixed Nash equilibrium is PPAD-complete even for 2-player games [2]. The question of approximate Nash equilibrium emerged as the central remaining open problem in the area of equilibrium computation. Other difficult problem is computation of the Nash equilibrium with special properties, for example Pareto optimal Nash equilibrium, where the payoff of players should also be maximized. The above problem is in NP optimization class [6]. Famous John Nash theorem [14] guarantees that every normal form game has at least one equilibrium in mixed strategies.

Our motivation is to show a new evolutionary method of finding Nash equilibrium in non-zero sum games. Recently it was shown that there is no effective algorithm, that computes Nash equilibrium in polynomial time [1], so at this moment, Differential Evolution (DE) algorithm which may be classified as an approximation algorithm, seems to be one of appropriate ways.

Our article is organized as follows: first, we give some examples of other algorithms (including mathematical methods) for computing exact and aproximate Nash equlibria. In the next section we give a brief description of the problem. In the section four we describe DE algorithm and explain details of the solution. Finally we show some experiments and results. We summarize with short conclusions.

2 Related Works

In general, the Nash equilibrium is a strategy profile such that no deviating player could achieve a payoff higher than the one that the specific profile gives him. The main algorithm for computing Nash equilibria is the Lemke Howson (LH) algorithm [9], which is a pivoting algorithm similar to the simplex algorithm for linear programming. LH is used to compute exact, single Nash equilibrium. It is very often described as the state of the art algorithm, but in [21] it was shown that for some game classes it has exponential time of finding solution. In 1991 algorithm based on the support enumeration was introduced [4], similar, but more effective algorithm was described in [19]. Both algorithms favor games, where support of both players is very small. Algorithm capable to effectively compute equilibria with larger support is presented in [20]. Second class of algorithms are strict mathematical methods to compute pessimistic approximation of the Nash equlibrium. Such algorithms give $\epsilon = \frac{3}{4}$ [8], and $\epsilon = \frac{1}{2}$ [3]. As far as we know the best aproximation is obtained in [24] and it is equal to 0.3393. Only similar concept is used in program GAMBIT [11] and it is based on the quantum response equilibrium (QRE). Main difference between QRE and our approach is that we directly calculate approximate Nash equilibrium without any additional concept. Interesting complex work is presented in [18], where authors describe a few methods of searching of the Nash equilibrium. This approach is based on the computational intelligence methods. It is important, that authors focus on games with a small number of strategies, and are able to compute only exact

Nash equilibrium. Their approach is based on multistart, where algorithm with the same parameters is stared sequential.

3 Nash Equilibrium and Approximate Nash Equilibrium

A non-cooperative game in strategic form for two players is formally described as couple:

$$\Gamma(R, C) \tag{1}$$

and consists of:

- $\{X, Y\}$ - two elemental set of players, where X is called row player, and Y is called column player;
- finite set of pure strategies for every player: $X = (x_1, x_2, ..., x_m)$ for first player, and $Y = (y_1, y_2, ...y_n)$ for second player;
- the payoff matrix $C \in R^{M \times N}$ for the row player and $R \in R^{M \times N}$ for the column player.

Mixed strategy for both players may be defined as strategy profile (probalility distribution over the set of pure strategies). A pure strategy provides a complete definition of how a player will play a game. In particular, it determines the move a player will make for any situation he could face. A player's strategy set is the set of pure strategies available to that player. Important concept is strategy support, which may be defined as subset M_x: $\forall_i, x_i \in M_x, P(x_i) > 0$ and it reflects all pure strategies of the player with non-zero probability of the selection. Main problem of this article is indicated by two equations:

$$\forall_i, x_i \in M_x, x_i R y^* \leq x^{*T} R y^* \tag{2}$$

$$\forall_i, y_i \in N_y, x^{*T} C y_i \leq x^{*T} C y^*, \tag{3}$$

where:

x_i - the i-th pure strategy of the X player;
$x_i R y^*$ - the payoff for the Y using his mixed strategy y^* against the i-th pure strategy of the player X;
M_x,N_y - the support for the X and Y players;
x^{*T} - the transposed vector specifying the mixed strategy of the X.

A good solution is defined as Nash equilibrium if both above conditions are met. Assume that all utilities have been normalized to be between 0 and 1 (this is a common assumption, since scaling the utilities of a player by any positive factor, and applying any additive constant, results in an equivalent game). A set of mixed strategies is called an ϵ-approximate Nash equilibrium, where $\epsilon > 0$, if for each player all strategies have expected payoff that is at most ϵ more than the expected payoff of the given strategy. The ϵ equilibrium may be defined as follows:

$$\forall_i, s_i \in M_x, s_i R y^* \leq x^{*T} R y^* + \epsilon, \tag{4}$$

$$\forall_i, t_i \in N_y, x^{*T} C t_i \leq x^{*T} C y^* + \epsilon. \tag{5}$$

The ϵ-Nash equilibrium is a strategy profile such that no deviating player could achieve a payoff higher than the one that the specific profile gives him, plus ϵ. A stronger notion of approximate Nash equilibria is the ϵ-well-supported Nash equilibria, in which every pure strategy of the player has non-zero probability of chosing (this issue is beyond the scope of the article). An exact method of computing ϵ-Nash equilibrium will be described in the next section.

4 Differential Evolution

Differential Evolution is a population-based algorithm used for the optimization of multi-modal functions developed by Storn and Price [22]. This optimization method may be used for example to numerical optimization, neural network train [10], filter design [23] and other. We could tell at a glance that it may be similar to the typical evolutionary algorithm, but the differential evolution has very strong mutation schema, which directs search process to the global optimum. Mutation is the major genetic operator. It is not a trivial process and it also provides the algorithm convergence. Moreover, the mutation is performed before the crossover process. The pseudocode of the general DE algorithm is presented in algorithm 1.

Algorithm 1. Basic DE algorithm

1 Create the initial population of genotypes $P_0 = \{\boldsymbol{X}_{1,0}, \boldsymbol{X}_{2,0}, ..., \boldsymbol{X}_{n,0}\}$;
2 Set the generation number $g = 0$;
3 **while** *stop criterion is not met* **do**
4 Compute the fitness function for every genotype in the population $\{f(\boldsymbol{X}_{1,g}), f(\boldsymbol{X}_{2,g}), ..., f(\boldsymbol{X}_{n,g})\}$;
5 Create the population of trial genotypes V_g based on P_g;
6 Make crossover of genotypes from the population P_g and V_g to create population U_g;
7 Choose the genotypes with the highest fitness function from population U_g and P_g for the next population;
8 $generation = generation + 1$, go to step 4;

Before starting the optimization process, there is a need to construct the population of genotypes. It is necessary to initialize the population, providing a good distribution of individuals in exploration space. Single individual is represented by the linear vector:

$$\boldsymbol{X} = \{P(x_1), P(x_2), ..., P(x_n), P(y_1), P(y_2), ..., P(y_m)\},$$

where $P(x_n)$ is the probability of chosing n strategy of player X. Similar, $P(y_m)$ is the probability of chosing m strategy of second player. We assume, that number of strategies of both players is equal, so $n = m$. At figure 1 we could see the basic algorithm schema. After the crossover (point 5), the new child individual is compared with the shield vector.

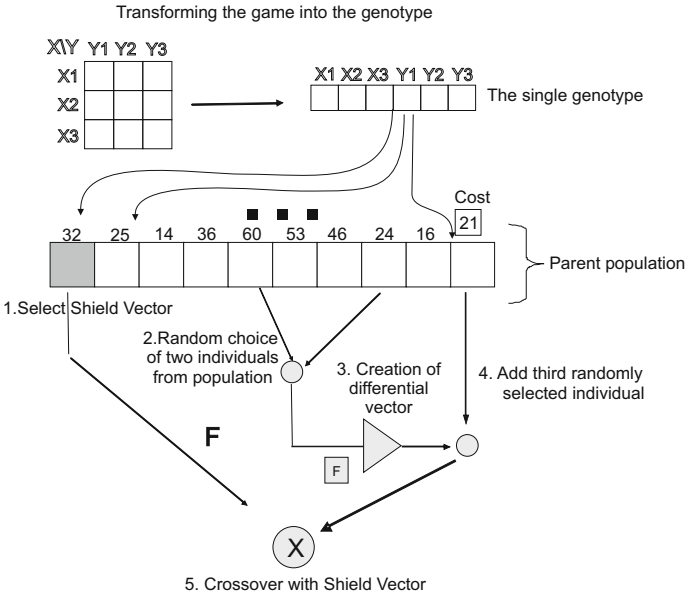


Fig. 1. DE schema

The mutation used in the DE is a much more complex process than schema used in traditional genetic algorithm but, thanks to the use of a mechanism known as differential vectors, it is possible to perform more directed search:

$$\forall_j \; U_{i,j} = P_{r_1,j} + F \cdot (P_{r_2,j} - P_{r_3,j})$$

where P is the population of parents. The parameter F specifies the strength of impact of the difference vector (between the two genotypes from the population). Numbers r_1, r_2 and r_3 determine index of the individual. The crossover process consists in the creation of a new individual (trial vector). Some of the elements of offspring vector come from individual and the others from the trial vector. This process can be expressed as follows:

$$\forall_i, \forall_j, U_{i,j} = \begin{cases} V_{i,j} \text{ when } RAND[0,1) < CR, \\ P_{i,j} \text{ in other case,} \end{cases}$$

where i is the index of the individual, j is the gene number, $V_{i,j}$ is gene of the i-th individual of the trial vector and CR is the crossover parameter. After the crossover process the offspring (U_i) is compared with its parent (P_i). Next, the better one of these individuals is added to the new population. The last step of the algorithm is the increment of the generation counter t. The best individual from the last generation is the result of the DE algorithm. A selection

schema used in this algorithm guarantee that individuals promoted to the next generation are not worse than individuals in actual generation.

$$\forall_i, \boldsymbol{X^*}_i = \begin{cases} \boldsymbol{U}_i \text{ when } f(\boldsymbol{U}_i) \leq f(\boldsymbol{X}_i), \\ \boldsymbol{X}_i \text{ in other case,} \end{cases}$$

where X^* is the individual in the next generation, $f()$ is the fitness function. The core of the proposed solution is the fitness function. The fitness function is calculated as below:

$$fitness = \sum_{i=1}^{n} |x^{*T} Ry* - x^{*T} Ry_i| + \sum_{i=1}^{n} |x^{*T} Cy* - x_i Ry^*| + n^2 \cdot F_{prob} \quad (6)$$

where:

$x_i Ry^*$ - the payoff for the Y using his mixed strategy y^* against the i-th pure strategy of the player X;
$P(x_i)$ - probability of chosing i-th pure strategy of X player;
F_{prob} - probability function.

Above probability function is the sum of all probabilities of both players. This value must be equal to 1, and every deviation from this value results in increased fitness value.

$$F_{prob} = |\sum_{i=1}^{n} P(x_i) - 1| + |\sum_{i=1}^{n} P(y_i) - 1| \quad (7)$$

The fitness function is computed for each individual from population U. The genotype with the lower fitness function value is transferred to the next population (fitness function equal to 0 is identified as global optimum).

5 Experiments and Results

The aim of this experiments is to show, that Differential Evolution approach is capable to obtain satisfactory results for the problem of computing approximate Nash equilibria. The above problem is strictly continuous, so DE seems to perfectly fit into this field. For all experiments described in this section we assume the following settings:

- the size of population $N = 50$;
- the crossover parameter $CR = 0.5$;
- the mutation parameter $F = 0.7$;
- the maximum number of iterations is equal to 10000 ;
- for every game algorithm was run 30 times.

Above parameters were experimentally set to the optimal values. For our experiments we used a 12 different random games with normalized payoffs generated with GAMUT [16]. It is standard test suite used in the game theory and it allows to generate different classes of games. We have classified these games to small (10 strategies per player), medium (15 strategies), and large (20 strategies)

Table 1. Percentage distance from the global optimum - ϵ-equilibria approximation

Game dimmension	Minimum ϵ	Maximum ϵ	Average ϵ	Median	Standard deviation
10 x 10 1	16.9%	45.2%	28.5%	27.0%	6.2%
10 x 10 2	23.5%	51.1%	33.9%	30.8%	7.8%
10 x 10 3	13.6%	57.9%	27.0%	23.0%	11.9%
10 x 10 4	13.6%	42.2%	25.6%	25.3%	7.7%
15 x 15 1	14.2%	40.0%	26.8%	26.6%	6.3%
15 x 15 2	16.2%	52.1%	28.1%	26.0%	7.9%
15 x 15 3	13.1%	50.6%	28.0%	25.8%	10.2%
15 x 15 4	14.3%	40.8%	24.6%	22.9%	6.5%
20 x 20 1	13.3%	46.4%	25.3%	23.8%	8.1%
20 x 20 2	14.0%	40.4%	25.0%	25.2%	6.4%
20 x 20 3	16.5%	40.4%	25.4%	24.8%	6.1%
20 x 2x 4	16.5%	40.2%	25.1%	24.9%	5.4%

Table 2. Number of the runs, in which algoritm was capable to obtain given threshold

Game dimmension	$\epsilon < 0.3393$	$\epsilon < 0.4$	$\epsilon < 0.45$	$\epsilon < 0.5$
10 x 10 1	83%	93%	96%	100%
10 x 10 2	53%	80%	90%	93%
10 x 10 3	73%	86%	90%	90%
10 x 10 4	86%	93%	100%	100%
15 x 15 1	90%	96%	100%	100%
15 x 15 2	76%	96%	96%	96%
15 x 15 3	76%	86%	90%	96%
15 x 15 4	90%	96%	100%	100%
20 x 20 1	86%	86%	96%	100%
20 x 20 2	90%	96%	100%	100%
20 x 20 3	86%	96%	100%	100%
20 x 20 4	93%	96%	100%	100%

instances. ϵ represents distance from the global optimum. In the table 1 we can see values from 12 different games: minimum, maximum, average, median and standard deviation.

In the section 2 we have mentioned a few algorithms, which are capable to compute the worst case approximate Nash equilibrium, where ϵ is equal to 0.3393 and 0.5. Two other considered values are 0.4 and 0.45. In the table 2 we can see percent of the games, which solution is better than assumed ϵ. Finally, in the figure 2 we can see the population convergence to the optimum (three different games: 10, 15 and 20 strategies per player). In conclusions we used real values of the ϵ, so 0.2 corresponds to the 20% distance from the optimum and so on.

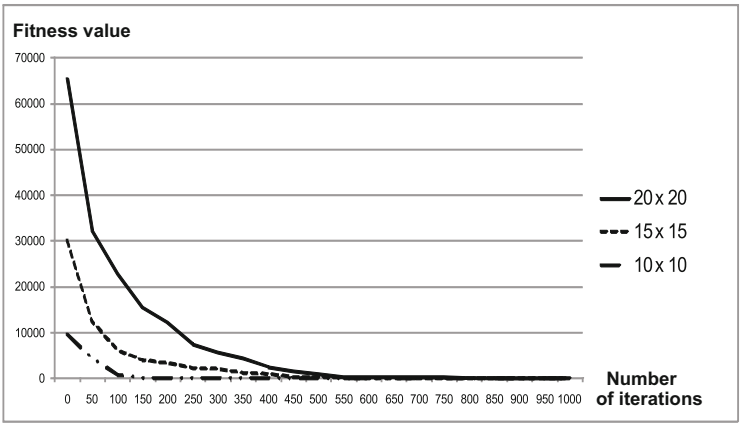


Fig. 2. The population convergence - three different games

6 Conclusions and Future Work

After the analysis of all test sets, the following conclusions may be reached: Differential Evolution was capable to solve every game (table 1), and average ϵ value was better than presented in the literature the worst case $\epsilon = 0.3393$. The worst case scenario gave weak results ($\epsilon > 0.4$), but in the other hand, the best results are very promising - almost every game has minimal ϵ value lower than 0.2. As we can see in the table 1 the majority of the results are better than $\epsilon = 0.3$.

The worst case of ϵ given by our algorithm is still worse than 0.3393 value (in single algorithm runs), but as we can see in the table 2, only a small number of algorithm runs gave $\epsilon > 0.3393$. It is worth of mention, that the above results hold for every type of the game - even, for the largest considered in this article games.

Our last results concern the population convergence to the global optimum. On the figure 2 we show three independent convergence charts (for the 10 x 10, 15 x 15 and 20 x 20 games). As we can see, the greatest progress comes in the first $500 - 600$ iterations. Next, the solution is deeply explored and progress is a far less significant. This property may be used in the future work - only to test candidate solution, and then eventually thoroughly explore it.

This research shows that Differential Evolution may be successfully used for the Nash equilibrium search problem. Unlike mathematical methods, the proposed approach is capable to give the set of solutions in the single algorithm run. Also, special features (like Pareto optimality) can easily be added. It is possible to set pure strategies of the single player, which must be used in Nash equilibrium and so on. Almost every problem described by I. Gilboa in [6] can be adapted to the proposed approach. Unfortunately, proposed method has some disadvantages. Mainly, at this moment we are not sure, how efficiently solve scalability problem.

References

1. Chen, X., Deng, X., Teng, S.-H.: Computing Nash Equilibria: Approximation and Smoothed Complexity. In: 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 603–612 (2006).
2. Chen, X., Deng, X.: Settling the complexity of two-player Nash equilibrium. In: 47th Symposium Foundations of Computer Science, pp. 261–271 (2006)
3. Daskalakis, C., Mehta, A., Papadimitriou, C.: A note on approximate Nash equilibria. Journal Theoretical Computer Science 410, 1581–1588 (2009)
4. Dickhaut, J., Kaplan, T.: A program for finding Nash equilibria, Working papers, University of Minnesota, Department of Economics (1991)
5. Etessami, K., Yannakakis, M.: On the Complexity of Nash Equilibria and Other Fixed Points (Extended Abstract). In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, pp. 113–123 (2007)
6. Gilboa, I.: Nash and correlated equilibria: Some complexity considerations. Games and Economic Behavior 1, 80–93 (1989)
7. Hammerstein, P., Selten, R.: Handbook of game theory - Chapter Game theory and evolutionary biology. University of Bonn (1994)
8. Kontogiannis, S.C., Panagopoulou, P.N., Spirakis, P.G.: Polynomial algorithms for approximating Nash equilibria of bimatrix games. Journal Theoretical Computer Science 410, 1599–1606 (2009)
9. Lemke, C.E., Howson, J.T.: Equilibrium Points of Bimatrix Games. Society for Industrial and Applied Mathematics 12, 413–423 (1964)
10. Magoulas, G.D., Plagianakos, V.P., Vrahatis, M.N.: Neural Network-Based Colonoscopic Diagnosis using On-Line Learning and Differential Evolution. Applied Soft Computing 4, 369–379 (2004)
11. McKelvey, R.D., McLennan, A.M., Turocy, T.L.: Gambit: Software Tools for Game Theory, Version 0.2010.09.01 (2010), http://www.gambit-project.org
12. McLennan, A.: The Expected Number of Nash Equilibria of a Normal Form Game. Econometrica 73, 141–174 (2005)
13. Murugavel, A.K., Ranganathan, N.: A Game-Theoretic Approach for Binding in Behavioral Synthesis. In: 16th International Conference on VLSI Design, pp. 452–458 (2003)
14. Nash, J.F.: Non-cooperative games. Annals of Mathematics 54(2), 286–295 (1951)
15. Neel, J., Mackenzie, A., Menon, R., Dasilva, L., Hicks, J., Reed, G.J.R.: Using game theory to analyze wireless ad hoc networks. IEEE Communications Surveys & Tutorials 7(4), 46–56 (2005)
16. Nudelman, E., Wortman, J., Shoham, Y., Leyton-Brown, K.: Run the GAMUT: A Comprehensive Approach to Evaluating Game-Theoretic Algorithms. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 2 (2004)
17. Ordeshook, P.: Game theory and political theory. Cambridge University Press, Cambridge (1986)
18. Pavlidis, N., Parsopoulos, K., Vrahatis, M.: Computing Nash Equilibria Through Computational Intelligence Methods. Journal of Computational and Applied Mathematics, 113–136 (2005)
19. Porter, R., Nudelman, E., Shoham, Y.: Simple Search Methods for Finding a Nash Equilibrium. Games and Economic Behavior 63, 642–662 (2008)
20. Sandholm, T., Gilpin, A., Conitzer, V.: Mixed-integer programming methods for finding Nash equilibria. In: Proceedings of the 20th National Conference on Artificial Intelligence, vol. 2, pp. 495–501 (2005)

21. Savani, R., von Stengel, B.: Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 258–267 (2004)
22. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
23. Storn, R.: Differential evolution design of an iir-Filter. In: IEEE International Conference on Evolutionary Computation ICEC 1996, pp. 268–273 (1996)
24. Tsaknakis, H., Spirakis, P.G.: An Optimization Approach for Approximate Nash Equilibria. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 42–56. Springer, Heidelberg (2007)

Co-operative, Parallel Simulated Annealing for the VRPTW

Rafał Skinderowicz

Institute of Computer Science, Silesia University, Sosnowiec, Poland
rafal.skinderowicz@us.edu.pl

Abstract. The parallel simulated annealing algorithm solving the vehicle routing problem with time windows (VRPTW) is presented. The algorithm consists of a number of processes which co-operate by synchronously exchanging data about their best solutions found so far. The exchange of information is made according to a pre-set communication scheme. The objective of the work is to investigate how the utilization of the data received by a process can affect the accuracy of its solutions to the VRPTW. Performance of the presented parallel algorithm is also considered.

Keywords: parallel simulated annealing, vehicle routing problem with time windows, parallel processes co-operation.

1 Introduction

The vehicle routing problem with time windows (VRPTW) consists in finding a route schedule serving a set of customers with a fleet of vehicles. Each customer has a specific demand and has to be served within a predefined time window. The total demand of any route must not exceed the vehicle capacity. The goal is to minimize the costs of operation, which involves reducing the number of vehicles needed and total distance travelled. The VRPTW is a NP-hard to solve exactly, thus heuristic algorithms are used to produce solutions in practice.

This work presents a parallel version of the well-known simulated annealing (SA) heuristic, which is used to solve the VRPTW. The algorithm consists of a number of components (processes) which co-operate periodically by exchanging their best solutions found so far. The information exchanged by the parallel processes can be used to focus the search on potentially attractive areas of the problem solution space. The main objective of the work is to explore how the co-operation can affect the quality of the solutions to the VRPTW.

The results of the work extend the previous efforts concerning the application of parallel simulated annealing to solve the VRPTW [4]. Parallel simulated annealing with asynchronous and synchronous inter-process communication to solve the VRPTW is described by Arbelaitz, Rodriguez and Zamakola, but the work does not discuss the impact of the balance between exploitation and exploration of the solution space on the quality of solutions [1]. Lee and Lee provide a detailed study of single and multiple Markov chains parallel simulated annealing

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 495–504, 2011.

algorithms [8]. The comprehensive study of parallelization of simulated annealing is presented by Azencott et al. [2].

Section 2 formulates the problem which is solved. Section 3 describes a sequential annealing algorithm, section 4 presents a proposed parallel version of the simulated annealing algorithm. Section 5 contains the discussion of the experimental results. Section 6 concludes the work.

2 Formulation of the Problem

Let $G = (V, A)$ be an undirected graph, where $V = \{v_0, v_1, \ldots, v_n\}$ and $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$. The vertex v_0 is the depot, the remaining vertices $v_i, i \neq 0$ are customers with a demand q_i of goods which has to be delivered to it. The time window $[e_i, f_i]$ and service duration s_i are associated with each customer v_i, where e_i and f_i are the earliest and latest time for a vehicle arrival, respectively. The vehicle can arrive to customer v_i before its earliest time e_i, but then it has to wait until the service can begin. The total demand of the customers serviced by a vehicle must not exceed its capacity Q. All tasks have to be completed within the time window of the depot $[e_0, f_0]$. The VRPTW consist in finding a set of vertex-disjoint routes which start and end at the depot v_0 such that each customer is served within its time window by exactly one vehicle. Two objectives are considered: minimizing the total number of routes (equal to the number of vehicles) and minimizing the total length of routes (equal to the distance travelled). The first criteria is usually considered more important, as in the work.

3 Sequential Simulated Annealing

Let S be a set of valid VRPTW solutions and $C : S \to R$ be a cost function that is to be minimized. Let $N(X)$, $N(X) \subset S$ be a set of neighbors of X for each $X \in S$. For the VRPTW the members of $N(X)$ are constructed by moving one or more customers among routes of X. The way in which these sets are created has a substantial influence on the quality of results obtained by a simulated annealing algorithm, so the sets $N(X)$ should be built through deep modifications of X.

In the sequential simulated annealing algorithm a sequence (or chain) of solutions (X_i), $X \in S$, is constructed as follows. An initial solution X_0 is computed using some heuristic. Given the current solution X_i, a potential next solution Y_i is chosen from $N(X)$ and its cost $C(Y_i)$ is calculated. Subsequently a solution X_{i+1} is set as follows:

$$X_{i+1} = \begin{cases} Y_i, & \text{if } C(Y_i) < C(X_i), \\ Y_i, & \text{with probability } p_i \text{ if } C(Y_i) \geq C(X_i), \\ X_i, & \text{otherwise .} \end{cases} \tag{1}$$

Probability p_i is defined as follows:

$$p_i = \exp(-\frac{C(Y_i) - C(X_i)}{T_i}), \tag{2}$$

where T_i is the positive number called the temperature. The sequence (T_i), $i = 0, 1, \ldots$, such that $T_i \geq T_{i+1}$ and $\lim_{i\to\infty} T_i = 0$, is called the cooling schedule, and a sequence of annealing steps within which the temperature remains constant is called the cooling stage. If the candidate solution Y_i has a lower cost than X_i, it is always accepted. In the other case of $C(Y_i) > C(X_i)$ it may become the next solution with the probability the greater, the higher the temperature T_i. Such uphill moves allow the search process to escape from local minima. It was proved that the convergence is guaranteed by the logarithmic cooling schedules of the form: $T_i \geq \frac{R}{\log(i+1)}$ for some constant R which depends on the cost function landscape [6]. Unfortunately, the process of annealing with the logarithmic schedule converges very slowly. For this reason an attempt to accelerate the SA by means of parallelism is very important.

Our algorithm comprises two phases f, $f \in \{1, 2\}$. The goal of phase 1 is to minimize the number of routes of the VRPTW solution, whereas phase 2 minimizes the total length of the routes starting from the solution from an earlier phase. In phase 1 the cost of solution X_i is computed as: $C_1(X_i) = c_1 N + c_2 D + c_3(r_1 - \overline{r})$, and in phase 2 as: $C_2(X_i) = c_1 N + c_2 D$, where N is the number of routes of solution X_i, D – the total travel distance of the routes, r_1 – the number of customers in a route which is tried to be shortened and eventually eliminated from the current solution, $\overline{r}$ – the average number of customers in all routes, c_1, c_2, c_3 – some constants. The temperature drops from an initial value $T_{0,f}$ according to equation $T_{k+1} = \beta_f T_k$, for $k = 0, 1, \ldots$ and some constants β_f ($\beta_f < 1$). The adopted cooling schedule does not guarantee the algorithm to converge to the optimal solution, but causes a significant reduction in the computation time. The computations are stopped after the specified number of cooling stages, a_1 and a_2, are completed in phase 1 and 2, respectively.

4 Parallel Simulated Annealing Algorithm

The main obstacle to parallelization of the simulated annealing algorithm is its inherently sequential nature. The sequence (X_i) of produced solutions can be described formally by non-homogeneous Markov chains. In these chains the probability of moving from one state to another depends on the preceding step and on the temperature of annealing. The candidate solutions that are rejected according to (1) are independent of each other and can be computed in parallel, but finding in parallel more than one acceptable solution leads to conflict. A single Markov chain (SMC) parallel simulated annealing (PSA), in which only one global solution in each iteration is accepted, preserves the sequential nature of SA, but is restricting from the viewpoint of performance [8]. The opposite approach is to allow each parallel process to follow its own search path and it is called the multiple Markov chain (MMC) PSA. Unfortunately, the MMC PSA

algorithm does not preserve the sequential nature of SA, which may adversely affect the quality of solutions [5]. In this work the latter approach is used, but parallel processes periodically exchange information about the best solutions found to date in order to improve the quality of final solutions.

The parallel simulated annealing algorithm comprises p components which are executed as processes $P_0, P_1, \ldots, P_{p-1}$ (Fig. 1). A process generates its own chain of solutions divided into two phases (lines 6–17). A phase consists of a_f cooling stages, during which the temperature remains constant (lines 9–15), and a cooling stage consists of L annealing steps.

```
1  L := E/p; {length of a cooling stage}
2  for P_j, j = 0, 1, ..., p − 1 do in parallel
3      Create the initial_solution using some heuristics;
4      current_solution_j := initial_solution;
5      best_solution_j := initial_solution;
6      for f := 1 to 2 do {execute phase 1 and 2}
7          {beginning of phase f}
8          T := T_{0,f}; {initial temperature of annealing}
9          repeat {a cooling stage}
10             for i := 1 to L do
11                 annealing_step_f(current_solution_j, best_solution_j);
12             end
13             Call co_operation procedure every ω annealing steps counting from
               the beginning of the phase;
14             T := β_f T;
15         until a_f stages were executed;
16         {end of phase f}
17     end
18 end
19 {Produce best_solution_{p−1} as the solution to the VRPTW}
```

Fig. 1. Parallel simulated annealing algorithm; constant $E = 10^5$ annealing steps

The processes co-operate with each other every ω annealing steps passing their best solutions found so far (line 13 in Fig. 1) according to the pre-set *communication scheme*. Fig. 2 presents a diagram of data flow between processes according to the scheme we called the *cascade scheme* (CS). Process P_i, $i > 0$, receives $best\_solution_{i-1}$ from process P_{i-1}, and compares it with its current best solution. Next the solution with lower cost is transferred to process P_{i+1}, $i < p-1$. Thus the best solution found by processes $P_0, \ldots, P_i$ is passed to process P_{i+1}, $i < p-1$. It is obvious that after the communication was performed, the process P_{p-1} has the best solution among those found by all processes.

A slightly modified version of the CS scheme is also presented. The main difference is that the processes in the algorithm are divided into sets I and D, such that $I \cap D = \emptyset$, $I \cup D = \{P_0, P_1, \ldots, P_{p-1}\}$. For simplicity, $I = \{P_0, P_1, \ldots, P_{N-1}\}$, where $N \in \{1, 2, \ldots, p\}$. If a process P_i from the set D receives a better solution

$$X_{\mathrm{s}} \rightarrow \left\{ \begin{array}{ccccccc} X_0^{(0)} & \rightarrow & X_0^{(\omega)} & \rightarrow & X_0^{(2\omega)} & \rightarrow \ldots \rightarrow & X_0^{(u_m\omega)} \\ & & \downarrow & & \downarrow & & \downarrow \\ X_1^{(0)} & \rightarrow & X_1^{(\omega)} & \rightarrow & X_1^{(2\omega)} & \rightarrow \ldots \rightarrow & X_1^{(u_m\omega)} \\ & & \downarrow & & \downarrow & & \downarrow \\ \vdots & & \vdots & & \vdots & & \vdots \\ & & \downarrow & & \downarrow & & \downarrow \\ X_{p-1}^{(0)} & \rightarrow & X_{p-1}^{(\omega)} & \rightarrow & X_{p-1}^{(2\omega)} & \rightarrow \ldots \rightarrow & X_{p-1}^{(u_m\omega)} \end{array} \right. \rightarrow X_{\mathrm{b}}$$

Fig. 2. Scheme of co-operation of processes (X_{s} – initial solution, X_{b} – final solution)

than its own best found so far, it is used as a *starting point* for the next annealing steps. Thus receiving a better solution causes a jump to the area of the solution space where the obtained solution was found. This greedy move is aimed at increasing the exploitation of attractive areas of the search space. Processes from the set I work *independently* of each other, i.e. received solutions do not affect their search paths. Increasing the number of independently operating processes allows greater exploration of the solution space. This co-operation scheme we name *cascade scheme with a division of roles* (RS).

Both presented co-operation schemes are also very effective in terms of the interprocess communication, which is very important to obtain a good speedup, especially on the parallel computer with distributed memory. The total number of messages sent between the processes in the *co_operation* procedure (line 13 in Fig. 1) is equal to $2(p-1)$, because there are $p-1$ receive and $p-1$ send operations performed.

5 Results of Experiments

In order to evaluate the proposed algorithm with interprocess co-operation schemes presented, an implementation in C language was made. Computations were performed on a set of tests chosen from the well-known Solomon repository [9]. The following parameters were fixed: $c_1 = 40000$, $c_2 = 1$, $c_3 = 50$, $a_1 = 40$, $a_2 = 100$, $E = 10^5$, $\beta_1 = 0.95$, $\beta_2 = 0.98$.

The main objective of the experiments was to investigate how the number $N = |I|$, $N \leq p$, of processes working in an independent manner influences the accuracy of solutions to the VRPTW. The algorithm was executed on selected tests from the Solomon repository with an increasing number of processes, $p =$ 10, 20, 32 and 40. Later in the work the assumption is taken that each process is executed by a separate processor (or a processor core in case of a multi-core processor), therefore the term *processor* will refer to a process executed by that processor.

With regard to the number of independently working processes there were several values examined, namely $N = \lfloor rp \rfloor$, where $r = 0.2$, 0.4, 0.6, 0.8 and 1.0. For example, if $r = 0.6$ and the number of processes $p = 10, 20, 32$ and 40, then the number of independent processes equals to $N = 6, 12, 19$ and 24,

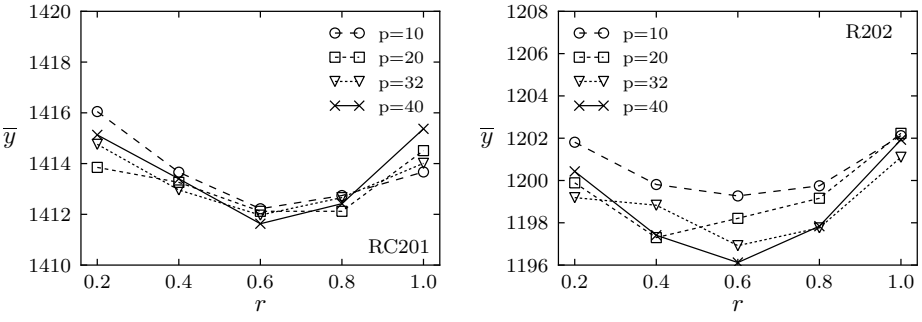


Fig. 3. Avg. length $\overline{y}$ vs. the number of independently operating processes $N = \lfloor rp \rfloor$ for tests R202 and RC201

respectively. For each test and for each combination of the parameters' values the algorithm was run 100 times. The experiments showed that quality of the solutions obtained depends on the ratio between independently and dependently working processors. The results of experiments for tests R202, RC201, RC203, RC108 are illustrated in Fig. 3 and Fig. 4. It can be seen that for the first three cases the quality of solutions increases when 60% ($r = 0.6$) or 80% ($r = 0.8$) of the total number of processes work in an independent manner. This trend is more evident the more parallel processes are being used. The quality of solutions drops when the processes do not co-operate at all ($r = 1.0$), which is equivalent to the maximum degree of exploration of the search space. On the other hand, when the majority of the processes greedily accepts received solutions ($r = 0.2$), the quality of solutions also decreases.

For the RC108 case (Fig. 4), the change in the number of independently working processes does not affect the quality of solutions to the VRPTW so much. It is likely that the co-operation of processes has a positive effect only if the majority of high-quality solutions are close to each other in the solution space, fortunately, this is often true for the VRPTW [3]. If the good-quality

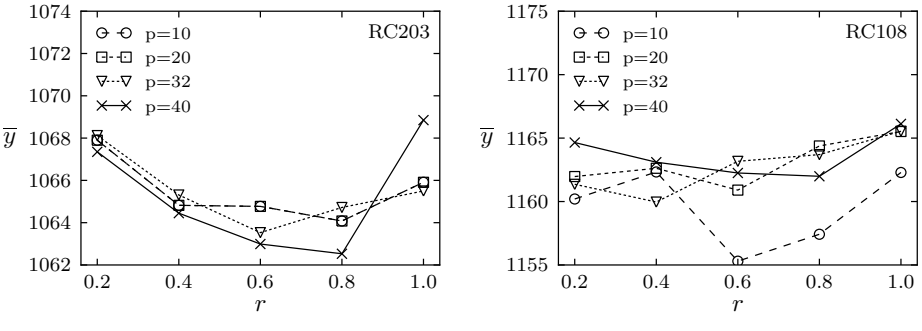


Fig. 4. Avg. length $\overline{y}$ vs. the number of independently operating processes $N = \lfloor rp \rfloor$ for tests RC203 and RC108

solutions are scattered all over the solution space, the co-operation of processors narrows the search unnecessarily. In fact, the co-operation presented can not improve exploration of the solution space, because any acceptance of the solution obtained from another processor reduces the cumulative poll of unique solutions.

In Tab. 1 detailed results for the selected tests are presented, the number of independent processors was set to 60% ($r = 0.6$, hence RS0.6). As can be seen, except for test RC108, the average total length of tours in solutions is larger for the serial version of the algorithm than for the parallel version. For the RC108 case the one-processor version of the algorithm achieves slightly better quality than the parallel version, which confirms earlier observations that the co-operation of processors is related to the number and locations of the local minima in the solution space. The length of tours in an average solution found is about 2% worse than the best known and the best solutions found are equal, in most cases, to the best known solutions.

Table 1. Results of experiments for tests R202, RC108, RC201, RC206 obtained by the RS algorithm with the number of independent processes set to $N = \lfloor 0.6p \rfloor$ (p – number of processors, $\overline{y}$ – average total length of routes in a solution, s_y – standard deviation of average length, y_b – best known solution, y_m – best solution over all runs, $\overline{t}$ – average execution time in seconds, s_t – standard deviation of execution time, S – speedup). Only solutions with the number of routes equal to the best known were considered.

test	p	$\overline{y}$	s_y	y_b	y_m	$\overline{t}$	s_t	S
R202	1	1215.26	23.58	1191.70	1192.00	122.2	1.0	1.00
	10	1199.27	10.47	1191.70	1191.70	12.3	0.5	9.91
	20	1198.21	9.82	1191.70	1191.70	6.3	0.5	19.40
	32	1196.92	7.77	1191.70	1191.70	3.9	0.3	31.08
	40	1196.12	6.15	1191.70	1191.70	3.2	0.4	38.20
RC108	1	1160.22	11.57	1139.82	1139.82	141.3	0.9	1.00
	10	1160.35	16.34	1139.82	1139.82	13.8	0.4	10.27
	20	1163.92	16.72	1139.82	1140.42	7.1	0.3	19.90
	32	1166.00	17.91	1139.82	1140.42	4.4	0.5	32.11
	40	1165.54	19.19	1139.82	1139.82	3.5	0.5	39.98
RC201	1	1418.25	8.50	1406.91	1406.94	11.7	1.2	1.00
	10	1412.89	6.18	1406.91	1406.94	11.5	0.5	9.72
	20	1411.56	5.07	1406.91	1406.94	6.0	0.2	18.73
	32	1411.19	5.70	1406.91	1406.94	3.7	0.5	30.47
	40	1412.07	4.67	1406.91	1406.94	3.0	0.0	37.24
RC206	1	1164.11	14.01	1146.32	1146.32	156.6	1.8	1.00
	10	1154.21	7.68	1146.32	1146.32	16.1	0.4	9.74
	20	1153.38	6.75	1146.32	1146.32	8.1	0.3	19.25
	32	1156.26	9.60	1146.32	1146.32	5.1	0.3	30.90
	40	1157.45	9.37	1146.32	1146.32	4.1	0.3	37.88

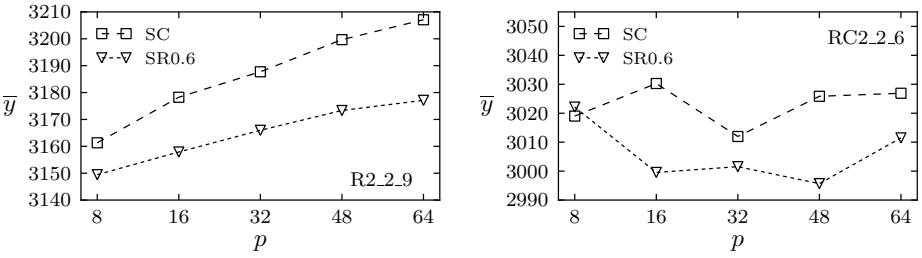


Fig. 5. A comparison of the avg. length $\overline{y}$ vs. the number of processes p for the algorithms SC and RS0.6 obtained for the tests R2_2_9 and RC2_2_6 (200 customers)

Based on the observations made, some preliminary experiments were taken in order to evaluate the performance of the algorithms on a set of tests with the larger number of customers. Six tests C2_2_3, R1_2_8, R2_2_9, RC1_2_4, RC2_2_1, RC2_2_6 with 200 customers from the set proposed by Gehring and Homberger were selected [7]. For each test the algorithm with CS scheme and the algorithm with RS scheme (with $r = 0.6$) were executed. As can be seen from Fig. 5, the second algorithm (RS0.6) allows to obtain better average solution values (in terms of the total routes length) than the algorithm with the CS scheme, which confirms our earlier observations that it is important to maintain the proper ratio between the exploration of the solution space and its greedy exploitation. In Tab. 2 the best solution values (collected over 100 executions for each test) are presented. As can be seen the best results obtained by our algorithms are in four cases very close ($\sim 1\%$) and in two cases (C2_2_3, RC2_2_6) of a better quality than the best known solutions.

5.1 Performance of the MPI Implementation

The proposed algorithm was implemented in the C programming language with MVAPICH1 v. 0.9.9 MPI library. All above-mentioned computations were performed on a cluster of 336 nodes which consisted of 2 Intel Xeon Quad Core 2.33 GHz processors, each with 12MB level 3 cache. Nodes were connected with the Infiniband DDR fat-tree full-cbb network (throughput 20 Gbps, delay 5 μs). The computer was executing a Debian GNU/Linux operating system.

As can be seen from fig. 6, the algorithm achieves very good speedups for up to 40 processors. Most of the work in the algorithm is evenly divided among processors, i.e. each processor performs exactly the same number of iterations of the simulated annealing portion of the algorithm. Only the main processor performs additional computations associated with the initialization part of the algorithm, which consists of loading the input data, collecting and saving the results.

In Tab. 1 detailed run times (in seconds) and speedups are presented. Surprisingly, for the RC108 test case superlinear speedups were observed for 10 and 32 processors. This is probably due to an increased amount of cache memory resulting from the use of multiple processors.

Table 2. The best results obtained by CS and RS0.6 algorithms for the selected tests from Gehring-Homberger repository (v – number of routes, y_b – best known solution, y_{CS} and $y_{RS0.6}$ – best solution values obtained for the algorithms presented)

test	v	y_b	y_{CS}	diff. [%]	$y_{RS0.6}$	diff. [%]
C2_2_3	6	1775,11	1775,08	-0,002	1776,96	0,104
R1_2_8	18	2952,65	2974,19	0,730	2974,19	0,730
R2_2_9	4	3092,53	3108,32	0,511	3108,32	0,511
RC1_2_4	18	2852,62	2880,8	0,988	2880,8	0,988
RC2_2_1	6	3099,53	3100,07	0,017	3106,08	0,211
RC2_2_6	4	2920,07	2881,71	-1,314	2878,16	**-1,435**

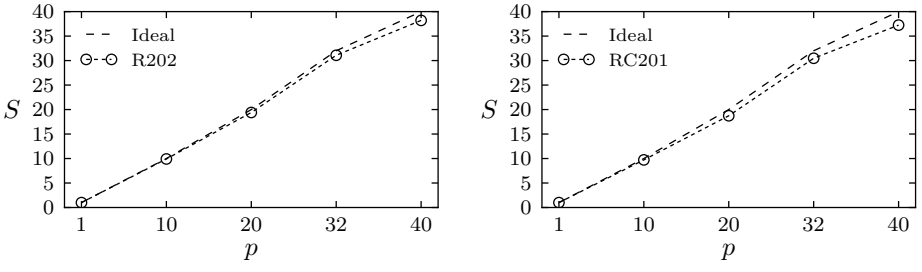


Fig. 6. Avg. speedup $\overline{S}$ for tests R202 and RC201. The ideal speedup is also marked as a reference.

6 Conclusions

The two parallel simulated annealing algorithms with processors co-operation to solve VRPTW were presented. The experimental results obtained for the selected tests confirm that it is possible to increase the chance of a single processor to find solutions of better quality through the use of data about the best solutions found by other processors. However, it is important to maintain the proper ratio between the exploration of the solution space and the greedy exploitation of accumulated knowledge (i.e. best solutions found so far). Both algorithms were able to find solutions of quality very close and in two cases better than the best known. It is worth noting that the proposed co-operation schemes have no significant negative effect on the algorithm performance. The MPI implementation of the algorithm achieves very good speedups up to 40 processors.

Acknowledgments. We thank the following computing centers where the computations of our project were carried out: Academic Computer Centre in Gdansk TASK, Academic Computer Centre CYFRONET AGH, Kraków (computing grant 027/2004), Poznań Supercomputing and Networking Center, Interdisciplinary

Centre for Mathematical and Computational Modelling, Warsaw University (computing grant G27-9), Wrocław Centre for Networking and Supercomputing (computing grant 30).

References

1. Arbelaitz, O., Rodríguez, C., Zamakola, I.: Low cost parallel solutions for the vrptw optimization problem. In: ICPP Workshops, pp. 176–181. IEEE Computer Society, Los Alamitos (2001)
2. Azencott, R. (ed.): Simulated annealing. Parallelization techniques. J. Wiley, NY (1992)
3. Czech, Z.J.: Statistical measures of a fitness landscape for the vehicle routing problem. In: IPDPS, pp. 1–8. IEEE, Los Alamitos (2008)
4. Czech, Z.J., Mikanik, W., Skinderowicz, R.: Implementing a parallel simulated annealing algorithm. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2009. LNCS, vol. 6067, pp. 146–155. Springer, Heidelberg (2010)
5. Durand, M.D., White, S.R.: Trading accuracy for speed in parallel simulated annealing with simultaneous moves. Parallel Comput. 26(1), 135–150 (2000)
6. Hajek, B.: Cooling schedules for optimal annealing. Math. Oper. Res. 13(2), 311–329 (1988)
7. Homberger, J., Gehring, H., et al.: Two evolutionary metaheuristics for the vehicle routing problem with time windows. Infor-Information Systems and Operational Research 37(3), 297–318 (1999)
8. Lee, S.-Y., Lee, K.G.: Synchronous and asynchronous parallel simulated annealing with multiple markov chains. IEEE Trans. Parallel Distrib. Syst. 7(10), 993–1008 (1996)
9. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35(2), 254–265 (1987), http://w.cba.neu.edu/~msolomon/problems.htm

The Parallel Ant Vehicle Navigation System with CUDA Technology

Wojciech Bura and Mariusz Boryczka

Institute of Computer Science, University of Silesia
Wojciech.Bura@asseco.pl, mariusz.boryczka@us.edu.pl

Abstract. The paper describes the parallel ant-based vehicle navigation algorithm adapted to run on GPUs. Experiments performed on real-world data show, that this new algorithm is more efficient and gives as good results as its predecessors. The algorithm's performance scales while growing up the number of GPUs' multiprocessors so it seems to be well designed for parallelization.

Keywords: ant system, vehicle navigation, GPU.

1 Introduction

In recent years we witness the rapid growth in popularity and development of car navigation systems. This is primarily due to the increasing availability of portable electronic devices equipped with GPS module, which may be used for this purpose. Route selection is one of the most important problems for car navigation system. In this problem, there are many possible routes for selection between a given pair of origin and destination. In addition there are many driver preferences in relation to these routes.

There are proposed several systems for optimum route selection for car navigation systems, e.g. genetic algorithms [1,10] or fuzzy logic [14,20]. One of them is AVN — an ant-based algorithm approach to vehicle navigation [16,17,18]. This algorithm is based on the ant algorithm [6,8]. It finds the optimal (or nearly optimal) routes between two points on the map, using user preferences for distance, traffic load, road width, risk of collision, quality and number of intersections. During calculations, the time of departure is also taken into account due to the fact that the weight of the individual segment of the route may have assigned different values at different time of day or night. It is important that we search for an algorithm which is able to propose not only the best (optimal) route, but a set of good solutions. Thus, it is possible to choose the alternative route when the unforeseeable situation (traffic jam, temporarily closed road etc.) occurs.

The improved AVN — NAVN was presented in [4] and its parallel version (PAVN) was proposed in [5]. Results of computational experiments show that this algorithm is much more efficient and capable of use for the real-word data.

From the beginning of the researches on the ant algorithms scientists tried to parallelize them to achieve better results in quality and computation time.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part II, LNCS 6923, pp. 505–514, 2011.

Most parallel implementations of ant algorithms are just parallelization of their standard form [2,3,7,15,19]. They differ only in whether the computations for the new pheromone matrix are done locally in all colonies or centrally by a master processor. Some authors also consider multi-colony algorithms [3,11,12,13], and different approaches for parallel ant algorithms are described in [9]. Unfortunately, the obtained speedup values for different problems (especially TSP) and using 8-15 processors ranges from value less then 1 (which means parallel execution is worse than serial execution) up to about 3. Recently the promising results gave the application of the CUDA approach, where the ant algorithm was specifically adjusted to this architecture, and speedup of running time of algorithm for TSP implemented on GPU was about 20 [21]. This inspired us to use this idea for the PAVN which produces better (but no enough) results than the sequential NAVN algorithm, and high performance of this algorithm is very important because of its potential use in the on-line mode. It is significant that the latest mobile devices are equipped with a GPU capable of general calculations (for example PowerVR SGX Series5XT in iPhone and iPad). This work presents OCLAVN — the OpenCL (CUDA) version of the PAVN algorithm.

The work is organized as follows. Section 2 shortly describes ant systems. Section 3 presents an original, sequential AVN algorithm and its improved version (NAVN), more efficient and capable of use for the real-word data. Section 4 mentions the basic parallel version of NAVN — PAVN, which runs in shared memory model. In section 5 the OpenCL version of PAVN algorithm is introduced (OCLAVN) and some algorithmic and programming issues that had to be solved during implementation are described. Section 6 presents the results of computational experiments on real-word data. Section 7 summarizes the work.

2 Ant Algorithms

Ant algorithms take inspiration from the behavior of real ant colonies to solve combinatorial optimization problems. They are based on a colony of artificial ants, that is, simple computational agents that work cooperatively and communicate through artificial pheromone trails [6]. The artificial ant in turn is a simple, computational agent that tries to build feasible solutions to the problem tackled exploiting the available pheromone trails and heuristic information. It has some characteristic properties. It searches minimum cost feasible solutions for the problem being solved. It has a memory storing information about the path followed until that moment. It starts in the initial state and moves towards feasible states, building its associated solution incrementally. The movement is made by applying a transition rule, which is a function of the locally available pheromone trails and heuristic values, the ant's private memory, and the problem constraints. When, during the construction procedure, an ant moves, it can update the pheromone trail associated to the edge. The construction procedure ends when any termination condition is satisfied, usually when an objective state is reached. Once the solution has been built, the ant can retrace the traveled path and update the pheromone trails on the visited edges/components.

3 Sequential Ant Algorithms for Car Navigation Systems (AVN and NAVN)

The AVN algorithm is designed for computing the optimal route between two points on the map. It is based on the ant system introduced by Dorigo, Maniezzo and Colorni [6]. AVN provides preferences desired by the user. The set of user parameters consists of coefficients controlling the importance level of distance, width, number of intersections, traffic, risk and quality of the proposed route.

First, initial setup of algorithm parameters is performed. Then ants are located at the starting point. Each ant is active until it reaches the destination point and is not blocked at the intersection. An ant is blocked when there is no way to choose to continue the travel. In the next step, the probability of each possible next direct route is calculated based on the cost function calculated for each active ant. The probability of the move from node i to node j for ant k is calculated as:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \prod_{l \in parameters} \xi_{ij_l}^{-2\alpha_l}}{\sum_{h \notin tabu_k} \tau_{ih}^\alpha \prod_{l \in parameters} \xi_{ih_l}^{-2\alpha_l}} & \text{if } j \in \text{tabu}_k \\ 0 & \text{otherwise} \end{cases}$$

where τ_{ij} is the value of pheromone trail on edge from i to j, α is a coefficient that controls importance level of τ_{ij}, tabu_k is a set of unavailable nodes, already visited by the ant k, ξ_{ij_l} are values of cost functions for parameter l for edge ij, α_l is a coefficient controlling the importance level of parameter l, and cost functions for all parameters for edge ij are [16]: $\xi_{ij_{distance}}$, $\xi_{ij_{width}}$, $\xi_{ij_{traffic}}$, $\xi_{ij_{risk}}$ and $\xi_{ij_{quality}}$.

Based on the calculated probability ant k selects a route to go. To do this, a random value q in range $\langle 0, 1 \rangle$ is compared with parameter $Q \in \langle 0, 1 \rangle$, to choose an exploitation or an exploration:

$$j = \begin{cases} \arg\max_{h \in J_i^k} \{p_{ih}^k\} & \text{if } q \leq Q \text{ (exploitation)} \\ S & \text{otherwise (exploration),} \end{cases}$$

In next steps the node selected by the ant is added to its tabu list and if ant k arrives the destination or is blocked at the certain node, it is deleted from the active ant list. For each ant that reached the destination the complete cost ψ of the whole route is calculated and for all calculated costs ψ the average cost χ is worked out. If $\psi_k < \chi$ then ant is added to AWA (Award Winner Ants) list, otherwise it is added to PLA (Punish Loser Ants) list. In addition, for an ant (if any) with cost ψ lower then the cost of the global best solution then new best solution is remembered.

At the end of every loop updating of pheromone trail takes place. The pheromone trail on the edge ij for ant k is updated as follows:

$$\tau_{ij}(t) = \begin{cases} \tau_{ij}(t-1) + \frac{av}{\psi_{ij}} & \text{if } k \in \text{AWA} \\ \tau_{ij}(t-1) \cdot pv & \text{if } k \in \text{PLA} \end{cases}$$

where av is the awarding coefficient, $av > 1$, and pv is the punishment coefficient, $0 < pv < 1$. Then the global evaporation of the pheromone trail is performed as $\tau_{ij}(t) = \rho \cdot \tau_{ij}(t-1)$, where ρ — evaporation coefficient, $0 < \rho < 1$. After the assumed number of repetitions of the algorithm the global best solution is returned as the result.

To improve the performance of the AVN algorithm, as a result of our research a new ant algorithm has been defined (NAVN) [4]. In NAVN algorithm, killing the blocked ants is replaced by their returns from dead ends. Pheromone trail is updated both locally (when an ant is traveling on the map) and globally (after every loop) on the paths from the best solution. Additionally, while moving back a blocked ant reduces pheromone trail on the edge which led it to the node without a way out. This considerably reduces the probability of selecting this edge by other ants.

The results of experiments on real-word data showed, that the NAVN algorithm can find good (although not optimal) solutions for large real map, much better than those achieved by the original AVN algorithm. It was also interesting, that the solution produced by the NAVN algorithm was regarded by many people (driving between the mentioned points) as the best route.

4 The Parallel NAVN — PAVN

The usefulness of the NAVN algorithm has encouraged us to improve its performance by its parallelization [5]. Main features of PAVN are:

- the PAVN algorithm exploits many threads,
- the part of the NAVN algorithm performed in parallel is the same as in the sequential version (walking through the graph) and is performed by a simple thread,
- one thread may support any number of ants,
- threads use the common memory both for the problem description and the pheromone,
- the synchronization of threads is held after each cycle.

Below the pseudo-code of the PAVN algorithm is presented.

Procedure `PAVN`

```
begin
    PrepareNormalization;
    Initialize;
    foreach loop do
        foreach worker do
            LocateAnts;
            foreach thread do
                RunWorker;
        while anyWorkerRunning do
            Wait;
        ValueAnts;
        AwardBestSolution;
        PunishLoserAnts;
        Modify q0;
    SelectBestOptimizedDirection;
end
```

Procedure `RunWorker`

```
begin
    foreach iteration do
        foreach ant do
            if antIsActive then
                ConstructProbability;
                if antHasNoMove then
                    MoveBack;
                else
                    SelectRoute;
                    UpdateTabuList;
end
```

The results obtained by the PAVN algorithm in comparison with the sequential version of the algorithm (NAVN) were:

- the PAVN algorithm with the same number of cycles produced a little bit better results and the performance time was shorter (about 50%) for PAVN,
- increasing the number of cycles led to further improvement of the results (with the performance time similar to the sequential version),
- increasing the number of threads (the number of processor cores were constant) as well as the number of ants made both the computation time and the quality of results worse.

5 OpenCL Ant Vehicle Navigation Algorithm)

Our further improvement of the PAVN algorithm was performed exploiting the computational potential of modern GPUs — devices equipped with the CUDA architecture (Compute Unified Device Architecture) from nVIDIA.

5.1 The CUDA Architecture

CUDA is an universal architecture, developed by nVIDIA, for parallel computing on graphics processing units (GPUs) and on dedicated accelerators (eg. Tesla). This makes it possible to use GPUs for general computing performed so far only on CPUs (Fig. 1). In contrast to the CPU, GPU's architecture requires the allocation of tasks to multiple threads which are executed relatively slowly, but globally offering a solution faster than traditional, general purpose, sequential processors. Such a GPU devices are ideally suited for computations that can be run in parallel.

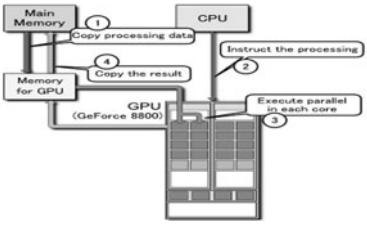


Fig. 1. Processing flow on CUDA

In order to be able to run our software on hardware from other manufacturers in the future, we decided to use an OpenCL programming interface. OpenCL (Open Computing Language) is a low-level API for heterogeneous computers, which may include the use of the CUDA architecture. Exploiting OpenCL, developers can create programs that run on the GPU device using programming language based on the C99 standard.

5.2 Adaptation of the NAVN Algorithm for CUDA and OpenCL

To be able to run the NAVN algorithm on GPU it was necessary to introduce various modifications to take into account the specificities of the target hardware architecture (CUDA) and the OpenCL programming interface as well. This resulted in the new version of the PAVN algorithm — OCLAVN

Optimization of Data Transfer between Host and Device. Computations in CUDA architecture are performed on heterogeneous environment. The user application is divided into two parts: a host system code, and a GPU code, and both have to communicate each other. In OpenCL, data must be transferred from the host to the device, and vice verse. These transfers can decrease overall performance so they should be minimized. Data should be kept on the device as long as possible. The OpenCL version of PAVN meets these considerations. Almost all operations are performed by various kernels on GPU with data kept in device's memory.

Global Memory Usage Minimization. The OpenCL specification describes three kinds of memory: global, local and private. They differ in term of scope, capacity and access time. Global memory has highest capacity and is accessible by every thread, but is significantly the slowest. Local memory is shared across threads in the same group, and private memory is private to a thread and is not visible to another threads. In many places we had to change the logic of PAVN to minimize the need of access to global memory. For example, ants' status data and solutions build by the ants are all located in local memory to improve performance.

Eliminating the Need for Synchronization. During calculations there are moments when two (or more) concurrent threads need to update the same global memory location. To solve these kind of issues the various methods of synchronization may be used. For example, atomic update operations or semaphore scheme can be used to avid the conflicts. OpenCL supports integer atomic operations on global memory. Therefore, it is possible to implement semaphores and use them to avoid simultaneous access to the same memory points by various threads. Unfortunately, atomic operations involving global memory slow down the execution of entire kernel, so we have decided to change the PAVN algorithm so that there is no synchronization. There are two major sources of memory conflicts in PAVN: on-line pheromone trail update in procedure *SelectRoute* and whole procedure *PunishNonActiveAnts*. They have been excluded from the final OCLAVN algorithm version, and the pheromone trail is now updated in procedures *AntMoveBack* and *AwardBestSolution*.

Code Optimizations for the SIMD Architecture. For best performance threads should be running in groups of at least 32 elements (warp size), with total number of threads of thousands. Branches in the program code do not impact performance significantly, providing the way that each of 32 threads takes the same execution path (the Single Instruction Multiple Data execution model). In various places we have changed the PAVN algorithm to achieve as much as possible the same execution path of all threads in the same group.

Other Considerations. OCLAVN uses single-precision floats because they provide the best performance on CUDA GPUs. For the same reason native runtime math operations are used. Functions using *native_functionName* map directly to the hardware level. They are faster but provide somewhat lower accuracy, yet enough good for our application.

The OCLAVN algorithm may be described as follows (procedure *RunAnts* is invoked for kernel calculations):

```
Procedure OCLAVN
begin
    PrepareNormalization;
    Initialize;
    Init MemoryOnDevices;
    foreach device do
        Write SearchMap;
    foreach loop do
        foreach device do
            InvokeKernel RunAnts for
            antCount * warpSize
            threads in antCount
            groups;
        foreach device do
            Read AntSolutions;
        foreach device do
            ValueAnts;
        Modify q0;
        foreach device do
            Write ChangedParams;
        if BestSolutionIsChanged then
            foreach device do
                Write BestSolution;
        foreach device do
            InvokeKernel
            AwardBestSolution for
            bestSolutionStepCount
            threads;
    FreeMemoryOnDevices;
    SelectBestOptimizedDirection;
end
```

```
Procedure RunAnts
begin
    threadId = GetLocalThreadId;
    antNum = GetGroupId;
    if threadId == 0 then
        CopyGlobalParamsToLocalMemory;
        InitializeLocalAnt;
    Barrier;
    InitializeTabuList;
    foreach iteration do
        if antIsActive then
            ConstructProbability;
            if threadId == 0 then
                if antHasNoMove then
                    MoveBack;
                else
                    SelectRoute;
                    UpdateTabuList;
            Barrier;
    CopyLocalAntToGlobalMemory;
end
```

In the OCLAVN algorithm's pseudo-code special comment should be provided for keywords **Write**, **Read** and **InvokeKernel**: **Write** realizes memory transfer from host to global memory on device, **Read** — memory transfer from global memory on device to host memory, and **InvokeKernel** runs given procedure on device by given number of threads (work items) divided into groups (work groups).

6 Computational Experiments

Data for the experiments were collected from the system Open Street Map (OSM) in the area of the city of Katowice. These are the actual data entered into the system through the efforts of numbers of volunteers. The map used in experiments consists of 31044 nodes and 68934 edges. The average number of edges (incoming and outgoing) per node is 2.2. The detailed data for the experiments were (other parameters of the OCLAVN are presented in Tab. 1): OSM start node id: 383783583 (suburbs of Katowice), OSM end node id: 384912139 (downtown of Katowice), departure time: 17:30, and speed: 40 km/h.

Table 1. OCLAVN algorithm's parameters used during computations

Algorithm	α	β	Pv	Bv	ρ	Q	φ	τ_0	m (no. of ants)	$N_{\max}$
NAVN	1.0	2.0	0.9	0.1	0.2	0.9	0.995	1/44500	30	50
OCLAVN	0.1	1.0	0.9	0.1	0.1	0.9	0.995	1/44500	30	50

Table 2. Parameters of selected nVIDIA GPUs

GPU	Multiprocessors	CUDA Cores	Compute Capability
GF 8400 GS	1	8	1.1
GF 9500 GT	4	32	1.1
Quadro NVS 320M	4	32	1.1
GF GTX 295	2x30	2x240	1.3

Table 3. Selected results of experiments

CPU	GPU	Algorithm	Cost	Time (ms)
Intel Core2 Quad 2,4 GHz	GF 8400 GS	OCLNAVN	15702	39842
Intel Core2 Quad 2,4 GHz	GF 9500 GT	OCLNAVN	15152	18618
Intel Core2 Duo 2,5 GHz	Quadro NVS 320M	OCLNAVN	14939	15270
Intel Core I5 2,67 GHz	GF GTX 295	OCLNAVN	15538	**5232**
Intel Core I5 2,67 GHz	-	NAVN	15651	6424

In our experiments only the preferences of the distance were used, other values were constant. This is due to the limited scope of information available in the OSM system. Tab. 1 shows parameter's values, common to both algorithms used during experiments. Changes in the logic of the original NAVN algorithm forced the usage of different parameters' values for sequential NAVN and parallel OCLAVN. These values have been discovered during preliminary runs of algorithms. Experiments were carried out on Microsoft Windows XP and, in case of GF GTX 295, on Windows 7. All the applications were compiled by Visual C++ 2008. OpenCL version of the algorithm was build with libraries from NVIDIA GPU Computing Toolkit 3.2. Sequential NAVN also uses single-precision floats, so that the experimental results were more reliable.

OpenCL version of the algorithm was run on various CUDA capable GPUs form nVIDIA, which are described in Tab. 2. The experiments allowed us to observe how hardware capabilities influence the results of the experiments. In case of GF GTX 295 only one device available on GPU was used.

Tab. 3 presents selected results of experiments on different platforms. The results shown are average values from 10 runs. Quality of each result (in term of lower cost value) is very similar. The OCLAVN run on GF GTX 295 has the best time performance, better than results obtained by sequential NAVN (in fact, the performance on the CPU was only for the comparison of the quality of results, because features of the CPUs and GPUs for the sequential computations

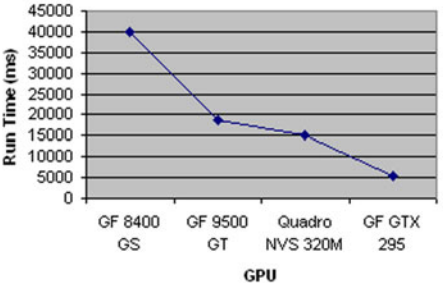


Fig. 2. OCLAVN algorithm runtime on different GPUs

are incomparable). Fig. 2 presents the execution time of OCLAVN on different GPUs and it shows that the algorithm's performance scales with growing number of multiprocessors. It is not a linear speedup, because the base AVN is not a completely data parallel algorithm. The search space is located in global memory on GPU, so that the performance limitation of that kind of memory was the most important challenge for our algorithm's tuning attempts.

7 Conclusions

In the course of the work we managed to propose OpenCL version of the NAVN algorithm. We have shown its usefulness during the tests on real-world data. The run-time of the OCLAVN algorithm was shorter then for the sequential version of NAVN. Very important result of experiments is the observation, that the OCLAVN algorithm is well designed for the use of GPUs, so the execution time scales while increasing number of GPUs' multiprocessors. Influence of the global memory performance limitations on overall algorithm performance can be decreased in the latest and future releases of GPUs. The promising example is new GPU architecture from nVIDIA called Fermi. One of the many improvements of the new architecture is significant speedup of global memory access.

References

1. Alhalabi, S.M., Al-Qatawneh, S.M., Samawi, V.W.: Developing a route navigation system using genetic algorithm. In: Proc. of 3rd International Conference on Information and Communication Technologies: From Theory to Applications, Damascus, pp. 1–6 (2008)
2. Bolondi, M., Bondaza, M.: Parallelizzazione di un algoritmo per la risoluzione del problema del comesso viaggiatore. Master's thesis, Politecnico di Milano (1993)
3. Bullnheimer, B., Kotsis, G., Strauss, C.: Parallelization strategies for the ant system. In: De Leone, R., et al. (eds.) High Performance Algorithms and Software in Nonlinear Optimization, Applied Optimization, pp. 87–100. Kluwer, Dordrecht (1998)
4. Bura, W., Boryczka, M.: Ant colony system in ambulance navigation. Journal of Medical Informatics & Technologies 15, 115–124 (2010)

5. Bura, W., Boryczka, M.: Parallel version of the NAVN system. In: Systemy Wspomagania Decyzji. Uniwersytet Śląski w Katowicach (2010) (in polish)
6. Cordon, O., Herrera, F., Stutzle, T.: A review on the ant colony optimization mtaheuristic: Basis, models and new trends. Mathware & Soft Computing 9, 1–36 (2002)
7. Dorigo, M.: Parallel ant system: An experimental study. Politechnico di Milano (1993) (unpublished)
8. Dorigo, 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. Janson, S., Merkle, D., Middendorf, M.: Parallel ant algorithms. In: Alba, E. (ed.) Parallel Metaheuristics. Wiley Book Series on Parallel and Distributed Computing, pp. 171–201. Wiley, Chichester (2005)
10. Kambayashi, Y., Tsujimura, Y., Yamachi, H., Yamamoto, H.: Integrating uncomfortable intersection-turns to subjectively optimal route selection using genetic algorithm. In: Proceedings of the 5th IEEE International Conference on Computational Cybernetics (ICCC 2007), Tunisia, pp. 203–208 (2007)
11. Krüger, F., Merkle, D., Middendorf, M.: Studies on a parallel ant system for the bsp model (1998) (unpublished)
12. Michels, R., Middendorf, M.: An ant system for the shortest common supersequence problem. In: Glover, F., Corne, D., Dorigo, M. (eds.) New Ideas in Optimization, pp. 51–61. McGraw-Hill, New York (1999)
13. Middendorf, M., Reischle, F., Schmeck, H.: Multi colony ant algorithms. Journal of Heuristics 8(3), 305–320 (2002)
14. Pang, G., Takahashi, K., Yokota, T., Takenaga, H.: Adaptive route selection for dynamic route guidance system based on fuzzy-neural approaches. IEEE Trans. on Vehicular Technology 48, 2028–2041 (1999)
15. Piriyakumar, D.A.L., Levi, P.: A new approach to exploiting parallelism in ant colony optimization. In: Proceedings of the International Symposium on Micromechatronics and Human Science (MHS), pp. 237–243 (2002)
16. Salehinejad, H., Farrahi-Moghaddam, F.: An ant based algorithm approach to vehicle navigation. In: Proceedings of the First Joint Congress on Fuzzy and Intelligent Systems (2007)
17. Salehinejad, H., Pouladi, F., Talebi, S.: A new route selection system: Multiparameter ant algorithm based vehicle navigation approach. In: CIMCA 2008, IAWTIC 2008 and ISE 2008 (2008)
18. Salehinejad, H., Talebi, S.: A new ant algorithm based vehicle navigation system: A wireless networking approach. In: Proceedings of the International Symposium on Telecommunications (2008)
19. Souto, R.P., de Campos Velho, H.F., Stephany, S.: Reconstruction of chlorophyll concentration profile in offshore ocean water using a parallel ant colony code. In: Proc. 16th European Conference on Artificial Intelligence (ECAI), Hybrid Metaheuristics (HM) (2004)
20. Teodorovic, D., Kikuchi, S.: Transportation route choice model using fuzzy interface technique. In: Proc. 1st International Symposium on Uncertainty Modeling and Analysis: Fuzzy Reasoning, Probabilistic Models, and Risk Management, Univ. Maryland, College Park, pp. 140–145 (1990)
21. You, Y-S.: Parallel ant system for traveling salesman problem on GPUs (2010) (unpublished)

Author Index