Ngoc Thanh Nguyen
Kiem Hoang
Piotr Jędrzejowicz (Eds.)

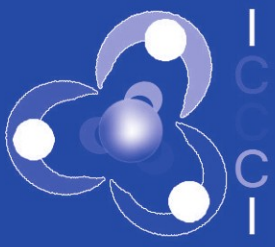
LNAI 7654

Computational Collective Intelligence

Technologies and Applications

4th International Conference, ICCCI 2012
Ho Chi Minh City, Vietnam, November 2012
Proceedings, Part II

2 Part II



Springer

Lecture Notes in Artificial Intelligence 7654

Subseries of Lecture Notes in Computer Science

Ngoc Thanh Nguyen Kiem Hoang
Piotr Jędrzejowicz (Eds.)

Computational Collective Intelligence

Technologies and Applications

4th International Conference, ICCCI 2012
Ho Chi Minh City, Vietnam, November 28-30, 2012
Proceedings, Part II

Springer

Volume Editors

Ngoc Thanh Nguyen
Wroclaw University of Technology
Institute of Informatics (I-32)
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
E-mail: ngoc-thanh.nguyen@pwr.edu.pl

Kiem Hoang
University of Information Technology
National Vietnam University VNU-HCM
Ho Chi Minh City, Vietnam
E-mail: kiem108@gmail.com

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

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-34706-1 e-ISBN 978-3-642-34707-8
DOI 10.1007/978-3-642-34707-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012950991

CR Subject Classification (1998): I.2.1, I.2.3-4, I.2.6-11, H.2.7-8, H.2.4, H.3.3-5, H.4.1-2, H.5.3, K.4.3-4, I.5.1-4, I.4.9-10, G.1.6, H.5.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

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

Printed on acid-free paper

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

Preface

This volume contains Part II of the proceedings of the 4th International Conference on Computational Collective Intelligence (ICCCI 2012) held in Ho Chi Minh City, Vietnam, November 28–30, 2012. The conference was organized by Wrocław University of Technology (Poland) in cooperation with the University of Information Technology (Vietnam National University VNU-HCM, Vietnam). The conference was run under the patronage of the Committee of Informatics, Polish Academy of Sciences, and the IEEE SMC Technical Committee on Computational Collective Intelligence.

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, the second International Conference on Computational Collective Intelligence (ICCCI 2010) held in Kaohsiung, Taiwan, and the third International Conference on Computational Collective Intelligence (ICCCI 2011) held in Gdynia, Poland, this conference 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 making group decisions or processing knowledge 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 2012 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 10 thematic streams and 10 special sessions.

The main track streams, covering the methodology and applications of CCI, included: Knowledge Integration, Data Mining for Collective Processing, Fuzzy, Modal and Collective Systems, Nature-Inspired Systems, Language Processing

Systems, Social Networks and Semantic Web, Agent and Multi-agent Systems, Classification and Clustering Methods, Multi-dimensional Data Processing, Web Systems, Intelligent Decision Making, Methods for Scheduling, Image and Video Processing.

The special sessions, covering some specific topics of particular interest, included: Collective Intelligence in Web Systems, Computational Intelligence for Business Collaboration, Advanced Data Mining Techniques and Applications, Industrial Applications of Computational Collective Intelligence, Cooperative Problem Solving, Computational Swarm Intelligence, Collective Intelligence with Semantic Technology, Smart Solutions in Computational Collective Intelligence, Semantic Methods for Knowledge Discovery and Communication, Mobile Intelligent Sensors and Systems Technology in Radial Assistive Living, and Modelling and Optimization Techniques for Business Intelligence.

We received 397 submissions from 33 countries. Each paper was reviewed by two to four members of the International Program Committee and international reviewer board. Only 113 best papers were selected for oral presentation and publication in the two volumes of the *Lecture Notes in Artificial Intelligence* series.

We would like to express our sincere thanks to the Honorary Chairs, Phan Thanh Binh, President of National University VNU-HCM (Vietnam), Tadeusz Więckowski, Rector of Wroclaw University of Technology (Poland), and Pierre Lévy, University of Ottawa (Canada), for their support.

We also would like to express our thanks to the keynote speakers, Philip Chen, President of IEEE SMC, University of Texas (USA), Witold Pedrycz, University of Alberta (Canada), Longbing Cao, University of Technology Sydney (Australia), and Adam Grzech, Wroclaw University of Technology (Poland), for their world-class plenary speeches.

Special thanks go to the Organizing Chairs (Anh Duc Duong and Radosław Katarzyniak) 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 papers to ensure the high quality of accepted papers. We thank the Publicity Chairs, Special Sessions Chairs, and the members of the Local Organizing Committee.

We thank the sponsors, the National Foundation for Science and Technology Development (Nafosted, Vietnam), Inha University (Korea), and Hue University (Vietnam).

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

It is our pleasure to announce that the conferences of ICCCI series are 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 and intend that ICCCI 2012 significantly contributes to the fulfillment of the academic excellence and leads to even greater successes of ICCCI events in the future.

November 2012

Ngoc Thanh Nguyen
Kiem Hoang
Piotr Jędrzejowicz

Organization

Honorary Chairs

Phan Thanh Binh	President of National University VNU-HCM, Vietnam
Tadeusz Więckowski	Rector of Wrocław University of Technology, Poland
Pierre Lévy	University of Ottawa, Canada

General Chairs

Ngoc Thanh Nguyen	Wrocław University of Technology, Poland
Kiem Hoang	University of Information Technology, VNU-HCM, Vietnam

Steering Committee

Ngoc Thanh Nguyen (Chair)	Wrocław University of Technology, Poland
Piotr Jędrzejowicz (Co-chair)	Gdynia Maritime University, Poland
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 Kowalczyk	Swinburne University of Technology, Australia
Ryszard Tadeusiewicz	AGH University of Science and Technology, Poland
Toyoaki Nishida	Kyoto University, Japan

Program Chairs

Dimitar Filev	IEEE SMC, USA
Piotr Jędrzejowicz	Gdynia Maritime University, Poland
Kazumi Nakamatsu	University of Hyogo, Japan
Edward Szczerbicki	University of Newcastle, Australia

Organizing Chairs

Anh Duc Duong	University of Information Technology, VNU-HCM, Vietnam
Radosław Katarzyniak	Wrocław University of Technology, Poland

Liaison Chairs

Quang A Dang	National Foundation for Science and Technology Development (NAFOSTED), Vietnam
Geun-Sik Jo	Inha University, Korea
Manh Thanh Le	Hue University, Vietnam

Local Organizing Co-chairs

Vinh Phuoc Tran	University of Information Technology, VNU-HCM, Vietnam
Phuc Do	University of Information Technology, VNU-HCM, Vietnam

Special Session Chairs

Amine Chohra	Paris-East University, France
Bogdan Trawinski	Wrocław University of Technology, Poland

Publicity Chairs

Dariusz Barbucha	Gdynia Maritime University, Poland
Cao Thi Kim Tuyen	University of Information Technology, VNU-HCM, Vietnam

Doctoral Track Chairs

Hong Hai Dam Quang	University of Information Technology, VNU-HCM, Vietnam
Tokuro Matsuo	Yamagata University, Japan

Keynote Speakers

Philip Chen, President of IEEE SMC, University of Texas, USA
Speech Title: *System Modeling: From Transparent Linguistic Interface in Fuzzy System to Kernel-Based Modeling*

Witold Pedrycz, University of Alberta, Canada
Speech Title: *Models of Collaborative Knowledge Management: A Perspective of Granular Computing*

Longbing Cao, University of Technology Sydney, Australia
Speech Title: *Modelling, Analysis and Learning of Ubiquitous Intelligence*

Adam Grzech, Wroclaw University of Technology, Poland
Speech Title: *Specifications and Deployment of SOA-based Applications within a Configurable Framework Provided as a Service*

Special Sessions

WebSys 2012: Collective Intelligence in Web Systems – Web Systems Analysis
Organizers: *Kazimierz Choroś and Mohamed Hassoun*

CIBC 2012: Computational Intelligence for Business Collaboration
Organizers: *Jason J. Jung and Huu-Hanh Hoang*

ADMTA 2012 on Advanced Data Mining Techniques and Applications
Organizers: *Bay Vo, Tzung-Pei Hong, and Le Hoai Bac*

IACCI 2012 on Industrial Applications of Computational Collective Intelligence
Organizers: *Van Tien Do*

CPS 2012: Special Session on Cooperative Problem Solving
Organizers: *Piotr Jedrzejowicz and Dariusz Barbucha*

CSI 2012: Computational Swarm Intelligence
Organizers: *Urszula Boryczka*

CIST 2012: Collective Intelligence with Semantic Technology
Organizers: *Geun Sik Jo and Trong Hai Duong*

SmartS 2012: Smart Solutions in Computational Collective Intelligence
Organizers: *Ondrej Krejcar and Peter Brida*

MissTRAL 2012: Mobile Intelligent Sensors and Systems Technology in Radial Assistive Living
Organizers: *Marek Penhaker, Martin Černý, and Martin Augustynek*

SMKDC 2012: Semantic Methods for Knowledge Discovery and Communication
Organizers: *Tzu-Fu Chiu, Chao-Fu Hong, and Radosław Katarzyniak*

MOTBI 2012: Modelling and Optimization Techniques for Business Intelligence
Organizers: *Le Thi Hoai An and Pham Dinh Tao*

International Program Committee

Jair Minoro Abe	Paulista University, Brazil
Cesar Andres	Universidad Complutense de Madrid, Spain
Costin Badica	University of Craiova, Romania
Dariusz Barbucha	Gdynia Maritime University, Poland
Maria Bielikova	Slovak University of Technology in Bratislava, Slovakia
Urszula Boryczka	Silesian University, Poland
Tru Cao	Vietnam National University HCM, Vietnam
Frantisek Capkovic	Slovak Academy of Sciences, Slovakia
Dariusz Ceglarek Poznan	School of Banking, Poland
Krzysztof Cetnarowicz	AGH University of Science and Technology, Poland

Shyi-Ming Chen	National Taichung University of Education, Taiwan
Tzu-Fu Chiu	Aletheia University, Taiwan
Amine Chohra	Paris-East University, France
Kazimierz Choros	Wroclaw University of Technology, Poland
Phan Cong-Vinh	NTT University, Vietnam
Irek Czarnowski	Gdynia Maritime University, Poland
Fabiano Dalpiaz	University of Trento, Italy
Paul Davidsson	Malmo University, Sweden
Mauro Gaspari	University of Bologna, Italy
Adam Grzech	Wroclaw University of Technology, Poland
Anamika Gupta	University of Delhi, India
Hoang Huu Hanh	Hue University, Vietnam
Chao-Fu Hong	Aletheia University, Taiwan
Tzung-Pei Hong	National University of Kaohsiung, Taiwan
Fong Mong Horng	National Kaohsiung University of Applied Sciences, Taiwan
Dosam Hwang	Yeungnam University, South Korea
Joanna Jedrzejowicz	Gdansk University, Poland
Gordan Jezic	University of Zagreb, Croatia
Joanna Jozefowska	Poznan University of Technology, Poland
Jason J. Jung	Yeungnam University, South Korea
Radosław Katarzyniak	Wrocław University of Technology, Poland
Chong Gun Kim	Yeungnam University, South Korea
Ondrej Krejcar	University of Hradec Kralove, Czech Republic
Piotr Kulczycki	Cracow University of Technology, Poland
Kazuhiro Kuwabara	Ritsumeikan University, Japan
Raymond Y.K.	Lau City University of Hong Kong, Hong Kong
Florin Leon	UTI, Romania
Hoai An Le-Thi	University of Lorraine, France
Xiafeng Li	Texas A&M University, USA
Andrei Lihu	Politehnica University of Timisoara, Romania
Adam Meissner	Poznan University of Technology, Poland
Jacek Mercik	Wroclaw University of Technology, Poland
Grzegorz J. Nalepa	AGH University of Science and Technology, Poland
Filippo Neri	University of Malta, Malta
Dinh Thuan Nguyen	Vietnam National University HCM, Vietnam
Linh Anh Nguyen	University of Warsaw, Poland
Thanh Thuy Nguyen	University of Engineering and Technology, Vietnam
Alberto Nunez	Universidad Complutense de Madrid, Spain
Manuel Núnez	Universidad Complutense de Madrid, Spain
Chung-Ming Ou	Kainan University, Taiwan
Ewa Ratajczak-Ropel	Gdynia Maritime University, Poland

Zbygniew Ras	UNC Charlotte, USA
Leszek Rutkowski	Czestochowa University of Technology, Poland
Ali Selamat	Universiti Teknologi Malaysia, Malaysia
Tadeusz Szuba	AGH University of Science and Technology, Poland
Yasufumi Takama	Tokyo Metropolitan University, Japan
Hoang Chi Thanh	Ha Noi University of Science, Vietnam
Michel Toulouse	Oklahoma State University, USA
Bogdan Trawinski	Wroclaw University of Technology, Poland
Jan Treur	Vrije University, The Netherlands
Iza Wierzbowska	Gdynia Maritime University, Poland
Drago Zagar	University of Osijek, Croatia
Danuta Zakrzewska	Lodz University of Technology, Poland
Constantin-Bala Zamfirescu	University of Sibiu, Romania

International Reviewer Board

Gely Alain
Duong Tuan Anh
Martin Augustynek
Branko Babuiak
Miroslav Behan
Raymond Bisdorff
Alexandre Blansch
Grzegorz Bocewicz
Mariusz Boryczka
Leszek Borzemski
Peter Brida
Conan-Guez Brieu
Krzysztof Brzostowski
Marcin Budka
Vladimir Bures
Martin Cerny
Ram Chakka
Chien-Chung Chan
Yue-San Chang
Ching-Fan Chen
Peng-Wen Chen
Chun-Hao Chen
Wei-Chen Cheng
Igor Chikalov
Nam Hoai Do
Phuc Do
Tien Van Do
Jarosław Drapała
Trong Hai Duong
Jerome Euzenat
Michael Feld
Robert Frischer
Marek Gajovsky
Michal Gála
N.P. Gopalan
Quang-Thuy Ha
Anne Hakansson
Tutut Herawan
Nguyen Thanh Hien
Huynh Xuan Hiep
Van Thien Hoang
Jiri Horak
Fang-Cheng
Hui-Huang Hsu
Rado Hudak
Proth Jean-Marie
Piotr Jedrzejowicz
Sang-Gil Kang
Vladimir Kasik
Sri Kolla
David Korpas
Tomas Kozel
Adrianna Kozierkiewicz-Hetmanska
Dariusz Krol
Edyta Kucharska
Marek Kukucka

Guo-Cheng Lan
Bac Le
Chun-Wei Lin
Wen-Yang Lin
Arne Lokketangen
Jakub Lokoc
Luca Longo
Wojciech Lorkiewicz
Xiuqin Ma
Zdenek Machacek
Juraj Machaj
Jaroslav Majernik
Marcin Maleszka
Nguyen Duc Manh
Mariusz Mazurkiewicz
Bernadetta Mianowska
Peter Mikulecky
Viorel Milea
Yang Mingchuan
Le Hoai Minh
Katarzyna Musial
Do Thanh Nghi
Long Thanh Ngo
Vu Thanh Nguyen
Thanh Binh Nguyen
Hayato Ohwada
Young-Tack Park
Rafael Parpinelli
David Pelta
Marek Penhaker
Marcin Mirosław Pietranik
Grzegorz Popek
Ibrahima Sakho
Andrzej Sieminski
Aleksander Skakovski
Rafał Skinderowicz
Janusz Sobecki
Nguyen Hung Son
Ja-Hwung Su
Zbigniew Telec
Le Hoang Thai
Le Nhat Thang
Huynh Thi Thanh Binh
Nguyen Duc Thuan
Nguyen Quang Thuan
Cuong Chieu To
Trong Hieu Tran
Hong Linh Truong
Christopher Turner
Bay Vo
Leuo-hong Wang
Tai-Ping Wang
Leon S.L. Wang
Yu-Lung Wu
Niu Yishuai
Mahdi Zargayouna
Krzysztof Zatwarnicki
Aleksander Zgrzywa
Beata Marta Zielosko
Jean-Daniel Zucker

Table of Contents – Part II

Multi-dimensional Data Processing

Web Systems

Intelligent Decision Making

Methods for Scheduling

Image and Video Processing

Collective Intelligence in Web Systems – Web Systems Analysis

Advanced Data Mining Techniques and Applications

Cooperative Problem Solving

Computational Swarm Intelligence

Semantic Methods for Knowledge Discovery and Communication

Table of Contents – Part I

Knowledge Integration

Data Mining for Collective Processing

Fuzzy, Modal and Collective Systems

Nature Inspired Systems

Language Processing Systems

Social Networks and Semantic Web

Agent and Multi-agent Systems

Classification and Clustering Methods

Modelling and Optimization Techniques for Business Intelligence

Generic Operations in the Structured Space of the Music

Tomasz Sitarek[1] and Wladyslaw Homenda[2,3]

[1] Ph.D. Studies, Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland
[3] Faculty of Mathematics and Computer Science, University of Bialystok
ul. Sosnowa 64, 15-887 Bialystok, Poland

Abstract. In this paper we study the problem of performing operations in structured spaces of data. This problem is one of the few most important aspects of intelligent knowledge processing. For complex spaces of data performing operations requires deep analysis, which usually employs description of the operation, its syntactic structuring and semantic analysis. In the study we focus our attention on employing operation in processing music data. The case study carries out transposition accomplished in printed music notation and in Braille music notation. It is shown that semantic analysis is necessary to transpose in Braille music notation and makes transposition clearer and easier in printed music notation.

Keywords: syntactic structuring, semantic mapping, data understanding, music representation.

1 Introduction

Accomplishing operations on structured spaces of data is required in intelligent knowledge processing and in intelligent man-machine communication. Considering operations on structured spaces of data we distinguish inner and outer operations. Former ones take input and produce output in frames of the same format of data representation, while latter ones act between different formats. General discussion is immersed in the case of processing music information. Accomplishing outer operations in spaces of music information was studied in [2]. In this paper we study inner operations in spaces of music information. We consider two cases of music data representation: printed music notation and Braille music notation. The discussion shows that intelligent information exchange in man-machine communication requires structuring the space of information being exchanged. Methodology of syntactic structuring of printed music notation and Braille music notation was discussed in some earlier papers, e.g. [2] and [3]. In this paper we continue this discussion in context of transposition, which is fundamental music operation. We justify that automatic transposition of Braille

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 1–10, 2012.

music notation requires semantic analysis in addition to syntactic description. Semantic analysis, though not necessary, simplifies and makes easier automatic transposition of printed music notation.

The paper is structured as follows. In section 2 we present theoretical bases for music information precessing, especially ideas of syntax, lexicon, semantics and inner and outer operations. Section 3 contains general description of transposition operation, illustrated with examples from printed music notation. Strict view on transposition in the structured space of music and usage of semantics in this operation is discussed in section 4. Finally, conclusions and directions of future research are presented in section 5.

2 Syntactic and Semantic Mappings

Any intelligent processing of constructions of languages of natural communications requires uncovering structures of raw data. There are different ways leading to structuring. Our interest is focused on employing syntactic and semantic structuring of music information with special emphasis put on Braille music notation and printed music notations. It is obvious that raw data without any structuring is useless in intelligent communication. Otherwise the processing covers some characters from alphabet without any meaning. The aim is to create generic method for integrate syntactic and semantic structuring of music information. This structuring allows for optimized processing of music information described in different languages including Braille music notation and printed music notation. For people with good eyesight we bind Braille music notation with printed music notation in this study. The method is an extension of a likewise study in [3].

2.1 Syntax

Syntactic structuring of music information is the first stage of the analysis process. We will utilize context-free grammars for syntactic structuring of Braille music notation and printed music notation. We refer to and will continue discussion of syntactic structuring outlined in [3].

Let us recall that we use formal grammars, which are systems $G = (V, T, P, S)$ where: V is a finite set of nonterminal symbols (nonterminals), T is a finite set of terminal symbols (terminals), P is a finite set of productions and S is the initial symbol of grammar, $S \in V$. In general productions can be seen as a finite binary relation $P \subset (V \cup T)^+ \times (V \cup T)^*$). A grammar G is context-free one (CFG) $\iff$ $(\forall p)(p \in P \Rightarrow p \in V \times (V \cup T)^*)$.

Since there is no evidence that Braille music notation is a context-free language, we do not attempt to construct a context-free grammar generating the language of Braille music notation. Instead we use context-free grammars covering the language of Braille music notation. Such grammars will generate all constructions of Braille music notation and some others, which are not valid Braille music constructions. This approach cannot be used in generating Braille

music scores or parts of scores or in checking their correctness. However, since we employ context-free grammars for processing scores, which are assumed to be correct, the approach is proven. A discussion on construction of context-free grammars covering printed and Braille music notations is outlined in [3]

2.2 Lexicon

Lexicon is the space of language constructions, each of them supplemented with possible derivation trees, also known as parsing trees. Lexicon includes relations between items of this space. Such a tree satisfies the following rules:

- it is a subtree of the derivation tree of the whole score,
- it is the minimal tree generating the given language construction,
- the minimal tree can be extended by a part of the path from the root of this tree toward the root of the score derivation tree

Due to the last condition, usually there are many trees for a given language construction. We do not recall the meaning of parsing tree in a context-free grammar, refer to e.g. [4] for definition of it.

Different trees supplementing a given language construction describe different context of the language construction. For instance, if we consider a sequence of consecutive notes, the minimal derivation tree for these notes matches all such sequences in the whole score, if more than one is present. If the minimal tree is extended to the root of the derivation tree of the whole score, then it represents only this given sequence of notes. The concept of the lexicon can be applied, for instance, for better understanding and better performing of structural operations, e.g. *find* operation.

2.3 The World: The Space of Hearing Sensation

Languages allows to describe a real world of things, sensations, thoughts, ideas etc. Braille music notation describes the space of hearing sensations, which can be outlined as the space $B \times D \times P$ of triples (b, d, p). Each triple defines the performed sound, where b is beginning time, d is duration and p is pitch of this sound. In general, objects of the real world may be outlined with much reacher set of features, but this simple triples are sufficient for our discussion, c.f. [3].

Above mentioned approach is very generic, refers to physical essence of a sound and has not any links to a particular type of music description. This structure can be used for any music notation, especially it can be used for Braille music notation. This definition of the space of hearing sensation is also very useful in case of other structural operations, e.g. *conversion* between different types of music description, c.f. [3].

The purpose of using the world of real objects is to tie meaning to syntactic structures of music descriptions. In this study we apply this attempt to lexicon elements of the Braille or printed music notations. This attempt allows us to cast different descriptions of music information and different formats representing

music information onto the space of hearing sensation. In this way, it is possible to construct collaborating methods, which operate on these different descriptions and formats.

The idea of collaborating syntactic and semantic methods has found the practical application to processing of music information. It has been involved in a real processing of Braille music accomplished in frames of the Braille Score project, c.f. [1].

2.4 Semantics

As mentioned in the previous section, descriptions of music notation expressed in different languages and representation of music notation in different formats are cast on the world of hearing sensations. Such casts are called semantics of descriptions and representations of music information. Formally, let B is the lexicon of Braille music notation and H is the space of hearing sensation. Semantics S is a relation:

$$S \subset B \times H$$

The space of language constructions is immersed in the space of sounds. The immersion gives values of real world to language constructions. The immersion defines meaning of language constructions, i.e. defines semantics.

2.5 Outer and Inner Operations in the Space of Music Information

Each operation has an input and output data, which are the spaces of information it operates on. In the case of music data processing these spaces are lexicons connected with respective music syntax. As it was stated, the syntax and lexicons are given by grammars. Any of the music types, such as printed music, Braille music, MIDI can have more than one grammar according to features to be emphasized.

Outer operations are those, whose input and output data come from different formats or descriptions of processed data. Of course, such different formats and descriptions can not be defined by the same grammar. The example of outer operation done in the space of music information is conversion from printed music notation to Braille music notation.

Inner operations are performed in frames of the same format or description of processed data. Their input and output data can be defined by the same grammar. The examples of the inner operations are all editing operations such as: adding, deleting, copying, pasting or finding signs. The same operation are inner ones in the music space of information. In the further part of this paper we discuss transposition, which is also inner operation.

Semantics Necessity for Outer Operation. Outer operations requires semantic mappings, because such types of activity change the format of the information. There is a need to know the meaning of the notation to write that information in other format. The example of the outer operation is conversion

from one format to another, e.g. from printed music to Braille music. It is obvious that semantic knowledge is crucial because of variety of manners of language structures creating. It is a consequent of the grammar productions.

Semantics Usefulness for Inner Operation. Some inner operations require semantic information and others – not. The first one are called hard operations, the others – easy, c.f. [2]. Roughly speaking, if an operation can be performed on its own language structures (e.g. Braille cells), it can be seen as an easy one. Typically, such an operation can be done on lexicon's element(s) directly involved in operation's definition.

A hard operation needs information acquired from elements of the lexicon, which are not used in direct description of the operation. A hard operation is rather difficult to be accomplished in the lexicon, but it is easy to be done involving the space of sounds.

Semantics Usefulness. In practical applications the semantics brings additional data to the notation of specified language. In the real case semantic bounding is not a situation like trade-off between lexicon data and semantic data.

In usage, additional semantic data does not worsen operation applicability. Usually such type of data makes operation easier.

The cause originates from observation that the space of hearing sensation is much more familiar to people than lexicon space. The notation, language or grammar is connected with the way how the hearing sensation is written and may differ according to the country or publisher. It is useful to perform such operations in the space of sounds which is a common space for all formats.

3 Transposing Operation

Transposing is an operation aimed to change (rise or decrease) the score pitches by settled number of halftones. Such activity implies changes in key signature of the score. The important thing is to keep intervals between consecutive notes.

This type of operation is performed, for instance, in case of incompatibility of the singer's voice and other sounds played in the score. In such the case transposition fits score sound scale to singer's voice.

Transposition is clear and simple in the space of music sounds since it moves all sounds up or down for given number of halftones. Therefore, all interval between sounds are preserved. However, due to irregularity of the scale, simple shift of notes at the stave, i.e. shift for given number of spaces/lines, will break intervals between sounds. It is necessary to moderate pitches of some notes using chromatic symbols (sharps, flats, naturals).

3.1 Transposing in the Space of Sounds

Thanks to the space of sounds definition the transposition operation is simple. This space consists of triples (b, d, p), where b is beginning time, d is duration and p is pitch of this sound.

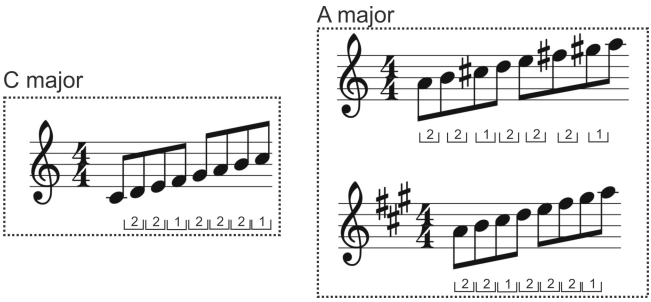


Fig. 1. Transposition from C major to A major, without chromatic symbols initially

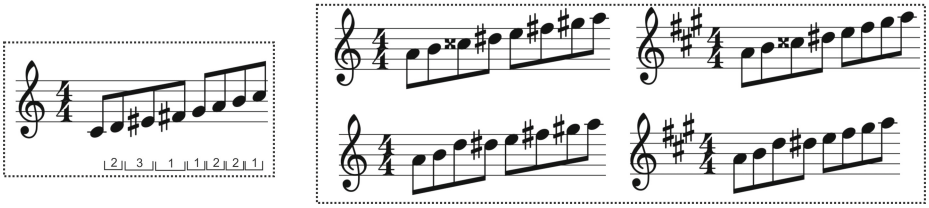


Fig. 2. Transposition from C major to A major, with 2 sharps initially

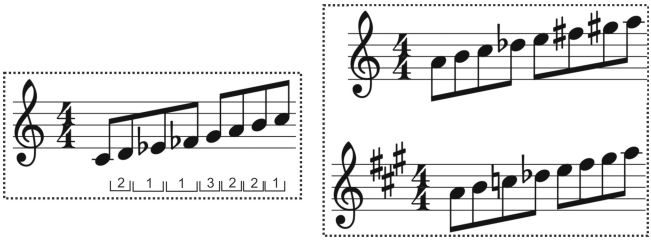


Fig. 3. Transposition from C major to A major, with 2 flats initially

Transposition operation is aimed to change pitches only. It does not affect beginning times and durations. If the pitch is denoted by numbers indicating halftones from the lowest tone in the space, the transposition by x halftones requires addition x to the pitch for all elements of the space of sounds.

3.2 Transposing Printed Music Notation – Example

Figure 1 presents transposition of the score in C major into A major key signature. To transpose it to A major scale there is a need to:

– for each note: note written as C write as A, written as D write as B, written as E write as C etc.,

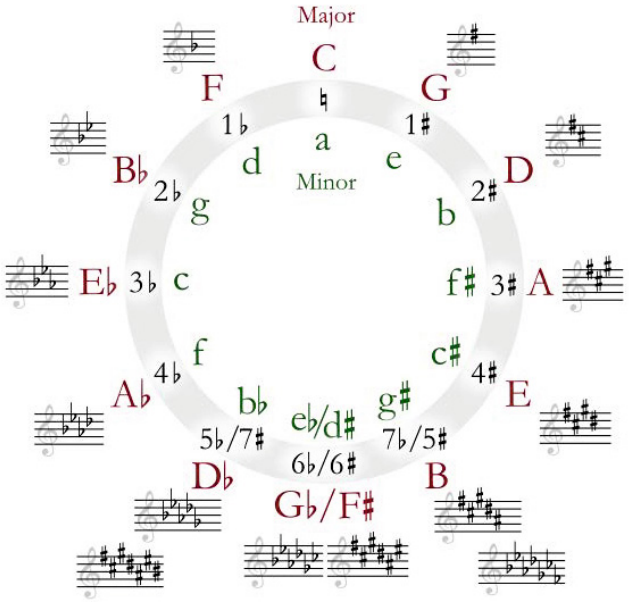


Fig. 4. Circle of fifths from http://en.wikipedia.org/wiki/File:Circle_of_fifths_deluxe_4.svg

- the above change is done shifting all notes at the stave for 5 lines/spaces (right upper part of this Figure),
- use chromatic symbols to moderate notes in order to preserve intervals (right upper part of this Figure),
- carry inserted chromatic symbols to the clef, these chromatic symbols creates key signature (lower part of right frame of Figures 1 and 3, and right part of right frame of Figure 2),

Sharps written in front of notes in the measure (Figure 1, right frame, upper staff) can be moved to the score key signature (Figure 1, right frame, lower staff). In all three staves in the Figure 1 respective intervals between consecutive notes are the same (2-2-1-2-2-2-1). The transposition increases each pitch by 9 halftones.

The chromatic symbols' managing issue is illustrated in the Figure 2 and 3.

In the Figure 2 in the left frame is shown pretransposed score with 2 sharps. First sharp is added to the third note which is going to be touched by chromatics managing, in opposition to the second sharp (added to the fourth note). This implies a few results. All of them express the solution of the transposition. The upper scores in the right frame has double sharp tied to the third note as a result of chromatics management. Double sharps of the lower scores in the right frame have been substituted by note's pitch rising (from C to D). Again, all scores preserve consecutive notes intervals.

Figure 3 presents transposition with flats signs. Similarly to the example from Figure 2, flats are placed near the third and fourth note (affected and unaffected by chromatics management). Solutions of this operation are presented in the right part of the illustration. Respective intervals are preserved.

Above algorithm is connected with circle of fifths (Figure 4). This artifact makes transposition process easier. All that has to be done is to set how many halftones to increase or decrease, select appropriate key signature and move notes with regards to chromatics management.

4 Transposing Operation with Semantics Usage

Transposition is an inner operation. It does not require semantics in case of printed music notation given by grammar described in [3]. All that has to be done is staff shift, key signature change and appropriate chromatics management. There is even no need to know octaves of notes to perform transposition. Anyway, semantics better explains the meaning of transposition and helps in its accomplishment.

Braille music (given by grammar described in [2]) case is more complicated and it uses semantics. This requirement involves with octave signs. Braille music marks some notes with octave, but with regard to interval between consecutive notes (c.f. [5]). The rule is as follows (two consecutive notes A and B, and we investigate if there is a need of octave sign before note B, $|A - B|$ – interval between A and B, $o(X)$ – octave of the note X):

1. $|A - B|$ is second or third $\Rightarrow$ no octave sign (even if $o(A) \neq o(B)$)
2. $|A - B|$ is fourth or fifth $\Rightarrow$ (octave sign $\Leftrightarrow o(A) \neq o(B)$)
3. $|A - B|$ is sixth or more $\Rightarrow$ octave sign (even if $o(A) = o(B)$)

Because of the subrule 2. some notes may loose octave sign, and some other notes may require octave sign after transposition. The necessity of the octave sign addition implies the need of knowledge of this note's octave and this implies semantics.

4.1 Performing Transposition in the Space of Sounds

As was stated above, transposition is inner operation that requires semantics in case of Braille music. Semantics is not necessary for transposition in the space of printed music, but is desired.

To make transposition a generic operation there is a need to perform it in the space of sounds. This means that semantics is going to be used as well as semantic mappings. Moreover that method makes operation to be easier in understanding.

The crucial elements in case of transposition in the space of sounds are semantic mappings. The semantic mapping and backward semantic mapping allows to analyze the notation as elements of the space of sounds.

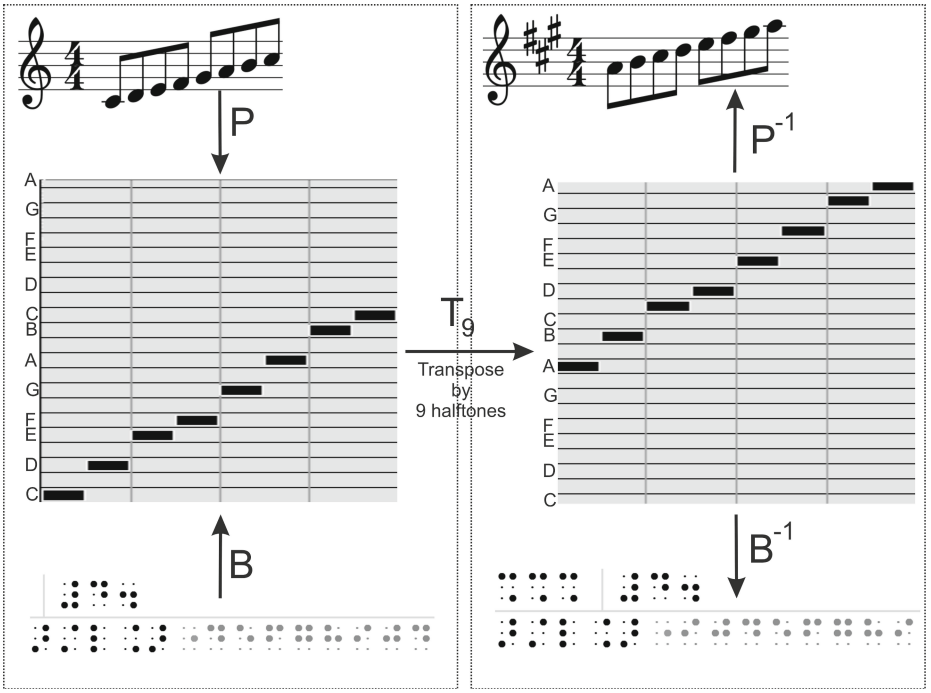


Fig. 5. Transposition in the space of sounds

Let assume that P is semantic mapping for printed music, P^{-1} is backward semantic mapping for printed music, B is semantic mapping for Braille music, B^{-1} is backward semantic mapping for Braille music. Lets be S the space of sounds. Then the transposition operation T in the space of sounds is defined as:

$$T : S \times \mathbb{N} \to S$$

and

$$T_x : S \to S$$

where $T_x(Y) = T(Y, x)$ transposes notation Y by x halftones.

The transposition by x halftones performed in the space of sounds and defined for the whole score is given by composition:

1. $B^{-1} \circ T_x \circ B$ – for Braille music
2. $P^{-1} \circ T_x \circ P$ – for printed music

Figure 5 illustrates transposition process in the space of sounds. In the left part of the image are presented printed (upper) and Braille (lower) notations. Each of them is transformed to the space of sounds by using semantic mappings (arrows described by P and B).

The space of sounds is then transposed by 9 halftones (arrow marked with T_9). The transposed space of sounds is transformed back to the space of printed and Braille music (P^{-1} and B^{-1} respectively).

5 Conclusions

In the paper we provide methods of performing operations in structured spaces of data. The discussion is firmly grounded in the space of music information, as we believe that automatic accomplishment of structured spaces of data is strongly domain dependent. Our studies show that general approaches to structured data processing would be firmly immersed in given domain. Expectation that it is possible to create a kind of *general solver*, independent on domain knowledge, is in practice far from current level of technology and research development. Therefore, we illustrate methods of syntactic structuring of descriptions of data and data itself, semantic analysis of them - which lead to a sort of automatic data understanding. The illustration is based on spaces of music information. We discuss transposition, which is one of the most important music operations, performed on two formats of music description: printed music notation and Braille music. The discussion would be easily adapted to other operations, e.g. extracting voice lines, selecting, searching. The general attempts to performing structured data operations would be applied to other domains as well. The most interesting application domain is natural language processing, since natural languages can be used to describe structures of data in different domains including spaces of music information.

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

Tomasz Sitarek contribution is supported by the Foundation for Polish Science under International PhD Projects in Intelligent Computing. Project financed from The European Union within the Innovative Economy Operational Programme (2007-2013) and European Regional Development Fund.

References

1. Grant no N R02 0019 06/2009, Breaking accessibility barriers in information society. Braille Score - a computer music processing for blind people, Institute for System Research, Polish Academy of Sciences, report, Warsaw (2011)
2. Homenda, W., Sitarek, T.: Performing Operations on Structured Information Space of Braille Music. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part IV. LNCS (LNAI), vol. 6884, pp. 232–241. Springer, Heidelberg (2011)
3. Homenda, W., Sitarek, T.: Notes on Automatic Music Conversions. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS (LNAI), vol. 6804, pp. 533–542. Springer, Heidelberg (2011)
4. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley Publishing Company (1979, 2001)
5. Krolick, B.: How to Read Braille Music, 2nd edn. Opus Technologies (1998)

Collective Cubing Platform towards Definition and Analysis of Warehouse Cubes

Duong Thi Anh Hoang, Ngoc Sy Ngo, and Binh Thanh Nguyen

Information Technology Center, Hue University
Hue, Vietnam
{htaduong,nsngoc,ntbinh}@hueuni.edu.vn

Abstract. Multidimensional data analysis, as supported by OLAP (online analytical processing), requires the computation of many aggregate functions over a large volume of historically collected data. Meanwhile, a recent trend in data communities has been the presence of dynamic, interdisciplinary data communities in which users can find and select data from a wide range of data providers. Using this approach, we have designed the Cubing service platform, which allows rapid retrieval of warehouse cubes in a way that would be familiar to any online shopper. In such an open marketplace, cubing services play a role as a metadata layer that maps cube definitions to the underlying schema and defines how the published cubes will be queried. The proposed platform couples an efficient cube selection mechanism with semantic reasoning capabilities, capable of processing large data sources, which expressed in a variety of formalisms, into a collection of warehouse datasets that expose the native metadata in a uniform manner. Thus, the platform is easily extensible and robust to updates of both data and metadata in the warehouse datasets.

Keywords: Data cube, OLAP, Open data, Linked data.

1 Introduction

As a collection of data from multiple sources, integrated into a common repository and extended by summary information, data warehousing (DWH) workloads usually consist of a class of queries typically interleaved with group-by and aggregation OLAP operators [1]. In OLAP, data cubes are used to support data analysis, in which the data is thought of as a multidimensional array with various measures of interest. The pressures of e-business productivity potential are pushing a data publishing architecture allowing managers to view, create and collaborate on Web reports and data analysis, providing additional functionalities including saving and sharing reports, navigating a network of warehouse cubes, and reaching back through to source data while doing analysis on web applications [2].

There arise a few data cubing architecture variations which make Web reporting and data analysis deployment scenarios an integral part of workgroup

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 11–20, 2012.

collaboration [2], i.e. cubes can be dynamically generated, using parameters specific to the user when the request is submitted. However, user requirements and constraints frequently change over time and ever-larger volumes of data and data modification, which may create a new level of complexity [3], maintaining multiple copies of the same data across multiple cubes for different kinds of business requirements. In order to achieve a truly open data place, we need standardized and robust descriptions of data cubing services [3], making easier the processes of configuration, implementation and administration of data cubes in heterogeneous environments with various deployment support.

Within the scope of this paper, we present the conceptual approach supporting a flexible exploration of large and complex search spaces of warehouse data cubes. In other words, individual data sources are placed under multiple classification hierarchies and can therefore be viewed by users in a multitude of ways. The cubing platform is used to explore sample data and enables the data users to quickly identify the most appropriate set of cube definitions in the warehouse so that they optimize two costs: the query processing cost and cube maintenance cost [4]. Moreover, on-demand data cube definitions can be generated dynamically by querying data and presentation parameters. Consequently, the platform can simplify publishing and sharing of data as well as increase the openness and accessibility of data spaces. To mark up data and create useful annotations that support navigation and discovery, given the set of user queries, the semantic indexing is aimed to support the selection of a set of materialized cubes from the entire population of data providers to minimize the query cost. The cube model, which extends object-oriented technology to data warehouses [5], is defined based on the generalization and inter-relationships among different cubes, thus improving the performance of query integrity and reducing data duplication as well as preventing the loss of data semantics in data warehouse.

The rest of the paper is organized as follows. Section 2 describes related background of this work. In Section 3, we give an overview of our conceptual cubing service platform along with enhanced data cube publishing foundation. A model for multidimensional cube selection and querying is described in Section 4, based on core ideas of semantic technologies. Finally, section 5 will conclude with a summary and an outlook on further research.

2 Related Works

The decision support is provided by OLAP tools, which present their users with a multi-dimensional perspective of the data in the warehouse and facilitate the report designs involving aggregations along the various dimensions of the data sets. Efficient computation of data cubes has been one of the focusing points in research since the introduction of data warehousing, OLAP, and data cube [1]. Previous studies can be classified into the various categories, e.g. efficient computation of data cubes with simple or complex measures [6, 7], selective materialization of data views [8, 9]. Most major BI vendors offer a specialized multidimensional storage engine that exposes aggregated data in business terms,

such as Business Objects, SQL Server Analysis Services, Oracle Hyperion and OLAP, and IBM Cognos, etc. provide structures by which the data from the star schema gets mapped into an aggregated cube [2]. Unfortunately, current in-house BI solutions, with complexity and a high cost of ownership, have been failed to deliver on the promise of a analysis solution that allows data cubes to respond to changing conditions. As time passes, the larger the data volume, the longer it takes to select, filter and aggregate the data cubes that is needed to generate a query response.

On the other hand, facilitated by data-as-a-service approach, new kinds of data markets have emerged, offering centralized points for publishing and sharing data [10–12]. These services make it easy to find data from a range of secondary data sources, then consume or acquire the data in a usable - and often unified - format. Several of these services are trying to create marketplaces for data, envisioning that data providers can offer their data sets for sale to data seekers. These services are aiming to be "Amazon for Data", while others in the category might be more accurate described as "Ebay for Data" or even "Wikipedia for data". Meanwhile, there is a growing list of statistical data providers such as the UN, the World Bank, Eurostat and others [13], already holding more than 100 million time series on a variety of topics, which support users to search, visualize, compare and download data from these providers.

Considering the potential approach of the developments of cubing services in business intelligence environment, the work to be developed in this research is aimed to achieve an efficient approach for storing, indexing and querying multidimensional data cubes. Thus, the approach is designed to could shield users from a vast amount of repetitive and tedious work to develop similar parts of a DWH application, which are tied to their own extraction and information delivery tools and a well-defined and understood data structure.

3 Cubing Platform Architecture

The cubing service platform provides a web-based information marketplace that is built on linked data principles, i.e., all datasets are referred to each other across different data publishers and domains, thereby adding value to both datasets. Moreover, the cubing platform supports a cube definition, submits it to the cube selection, and responds with the related data cubes. The parameters for the cube may be any combination of implicitly collected data as the user interacts with the cube services.

The architecture of the server consists of the main parts, i.e. the back-end and the web application. While the back-end services include data acquisition from data providers and detecting as well as indexing ontological concepts in multiple data sources, the web application provides the client with an efficient concept-based cube browsing and periodic refreshing of the underlying resources. As depicted in Figure 1, the results of the cubing platform are the appropriate aggregates and related data cubes to support the cube definition based on a thorough understanding of the cube model, how the cube will be queried, the amount

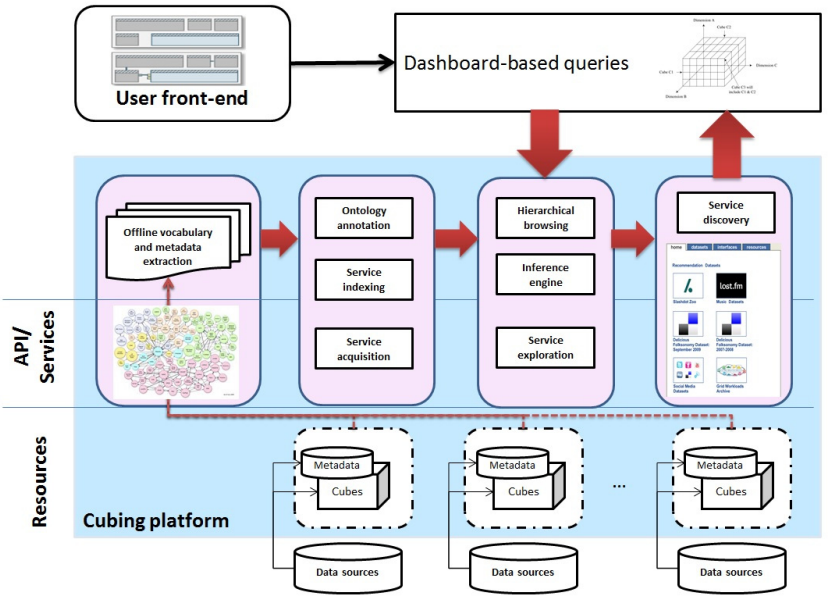


Fig. 1. The conceptual architecture of Cubing platform

of space allocated, and statistics from the datasets [3]. Hereafter, large populations of ad hoc information consumers can then create their own self- customized reports that can in turn be shared and modified within created workgroups.

3.1 Illustrative Example

We now introduce the running example that will be used to illustrate the use of cubing platform for Food sale analysis (FSA) [14]. Applications of food access indicators include Sources of food and income, Consumption of famine foods, Access to natural resources, Food self-sufficiency, etc. The components of the analysis plan are defined as follows.

- **Indicator:** A specific variable or combination of variables that gives insight into an aspect of the objectives. For example, if indicators for the access to food through cash crop sales and market purchases might be defined as the area currently planted with selected cash crops, compared with that under normal circumstances and the ratios of selling prices of selected cash crops to the costs of staple foods, now and under normal circumstances.
- **Data required and Data sources:** The information that must be collected to satisfy the broad information needs and the indicators. Examples include information about livelihoods, social structure; information about areas of land planted, average yields andmarket prices of items bought and sold.

- **Contextual information:** Details of the processes that led to the food emergency and identify potential responses.
- **Analysis type:** The type of parametric or non-parametric analyses that can be used to explore and interpret the data, e.g. non-parametric analyses of primarily quantitative data; or parametric analyses of statistical data.

Current serious concerns about food security have raised multiple open access policy datasets to provide over 40 indicators related to food security, commodity prices, economics, and human well-being. Much of this data is available for every country in the world and goes back over 50 years and is drawn from public, authoritative data sources like the World Bank, the FAO, UNICEF, and others, as well as IFPRI's own data.

3.2 Data Publishing

Being able to directly serve each individual resource as linked data, the cubing services can automatically provide multi-format representations for each resource in the dataset. Moreover, it is possible to use the various standard APIs provided by the platform, as well as to define additional custom APIs over the dataset, e.g. query language to perform structured queries against a dataset or perform a free-text search against the text fields indexed in a dataset.

The Semantic Cube Model. This section describes a semantic cube model that serves data cube marketplace , which extends object-oriented technology for data warehouses [5] and based on Data Cube vocabulary [15], taking into account connectivity by identifying the relationship between data cubes so as to reduce duplication of data in warehouse cubes, e.g. generalization, aggregation, and categorization, between different cubes. Especially, the cube model also supports the semantics of cube aggregation, i.e. while viewing the relevant data cube information, a user would be enabled to browse other data cubes' data and return the original data cube information.

When publishing a new dataset, warehouse providers offer a set of description metadata, such as typical example resources, links to APIs and endpoints, licensing information, categories and vocabularies used, or browsing methods on sample resources, etc. The actual data is then published to cubing platform either through the web interface or to the platform API, after which the platform will analyze the dataset. For example, the Global Hunger Index (GHI) is a dataset to measure and track global hunger, incorporating three interlinked hunger-related indicators - the proportion of under-nourished in the population, the prevalence of underweight in children, and the mortality rate of children. The modeled GHI [16] is published with four dimensions and relevant descriptions of GHI DataCube (described in RDF format in [16])

- Measure (*http://data.ifpri.org/rdf/ghi/2010/scovo/Measure*), indicating whether it is the GHI, or some other supporting statistic being provided.

- Country (*http://data.ifpri.org/rdf/ghi/2010/scovo/Country*),including labels and identifier for countries.
- DateRange (*http://data.ifpri.org/rdf/ghi/2010/scovo/DateRange*), each with a custom Date Range resource.
- EstimateOrNot (*http://data.ifpri.org/rdf/ghi/2010/scovo/EstimateOrNot*), noting whether the data is an official statistics, or is estimated based on methods outlined in the GHI report.

Automatically, based on a combination of the metadata provided by the providers and an analysis of the dataset itself, the platform will add dynamic template queries of the cube contents and make it available under the dataset's description. These contextual queries play the crucial component of the primary data collected in data cubes selection. Examples of contextual information for Economy and markets, which is essential to the interpretation of mortality, nutrition and food sale data and the development of response options, include *What have been the trends in the consumer price index over recent months and years? How accessible are the main markets to people affected by the crisis? What was the status of market food availability and access?* etc.

```
<rdf:Description rdf:about="http://data.ifpri.org/rdf/ghi/2010/qb/observation-1990-VN">
 <rdf:type rdf:resource="http://purl.org/linked-data/cube#Observation"/>
</rdf:Description>
<rdf:Description rdf:about="http://sws.geonames.org/1562822/">
 <rdfs:label>Vietnam</rdfs:label>
 <owl:sameAs rdf:resource="http://ontologi.es/place/VN"/>
 <rdfs:seeAlso rdf:resource="http://dbpedia.org/resource/Vietnam"/>
</rdf:Description>
<rdf:Description rdf:about="http://data.ifpri.org/rdf/ghi/2010/qb/observation-1990-VN">
 <ghiqb:refArea rdf:resource="http://sws.geonames.org/1562822/"/>
 <ghiqb:refPeriod rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1990-01-01</ghiqb:refPeriod>
 <ghiqb:ghi>24.8</ghiqb:ghi>
 <ghiqb:supportingData-pun rdf:datatype="http://www.w3.org/2001/XMLSchema#double">28
 </ghiqb:supportingData-pun>
 <ghiqb:supporingData-cuw rdf:datatype="http://www.w3.org/2001/XMLSchema#double">40.7
 </ghiqb:supporingData-cuw>
 <ghiqb:supportingData-cm rdf:datatype="http://www.w3.org/2001/XMLSchema#double">5.6
 </ghiqb:supportingData-cm>
 <qb:dataSet rdf:resource="http://data.ifpri.org/rdf/ghi/2010/qb/2010-GHI-Report"/>
 <sdmx-attribute:obsStatus rdf:resource="http://data.ifpri.org/rdf/ghi/structure/#obsStatus-A"/>
</rdf:Description>
<rdf:Description rdf:about="http://data.ifpri.org/rdf/ghi/2010/qb/observation-2010-VN">
 <rdf:type rdf:resource="http://purl.org/linked-data/cube#Observation"/>
```

Fig. 2. Dataset modeling example

Semantic Indexing. As already mentioned, the cubing platform leverages semantics in order to hide the composition complexity from the users. Concept annotation is performed to detect ontological concepts from a given ontology. After importing of and mediating between existing ontologies, the ontology is used in the actual selection phase by the users. The data cubes can be annotated with additional semantics, such as user preferences and business context.

The pre- and post-conditions of cube definition can be matched automatically. This enables the platform to suggest data cubes which can be connected to

other cubes, or which cubes are missing in order to make it equally described in terms of the ontology and matched with the cubes and dataset descriptions, thus suggests the right building blocks for a given query. Moreover, descriptions of existing third-party resources can also be mapped to the cube ontology in order to make them available to the cubing platform.

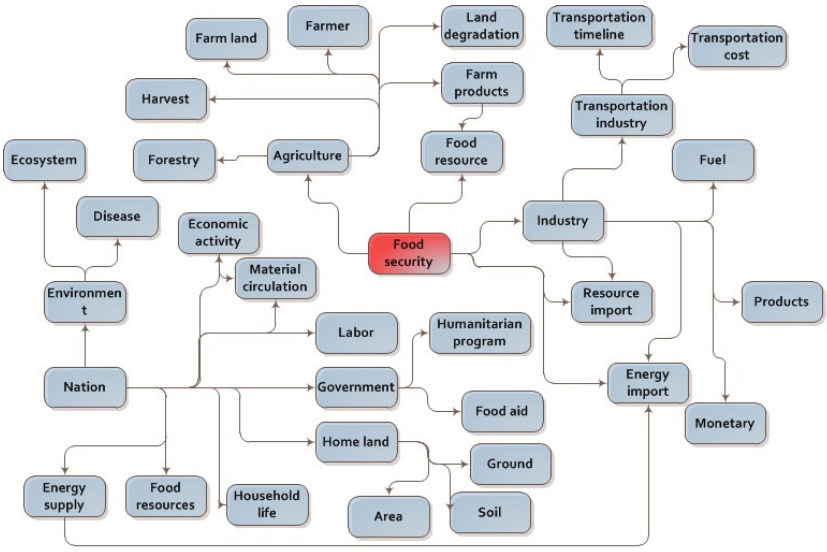


Fig. 3. Food market indicator indexing

As in our Food sale scenario, markets are of critical significance to food access in most situations. Many of the food access indicators described in the running example are based on market interactions, so it is essential to have indicators illustrating the ways in which markets function. Based on the relationship between demand and supply of grain, the cubing service needs to introduce key concepts and presenting the indicators used in an FSA to estimate food sales. Some of these may not be appropriate in every situation, and additional market indicators can be added, according to specific context.

4 Cube Selection and Querying

In current data marketplace, the data cubes become isolated bits of information in lack of a comprehensive semantic relationship among the cube datasets. Users retrieve the knowledge from one single angle and not from a global view; therefore, problems like data duplication, inconsistency, and query integrity could occur [17]. Therefore, the platform facilitates the exploration and discovery of unexpected patterns in concept co-occurrences across cubes, which might lead

to the generation of new warehouse cubes. It supports both data drill-down to focus the results, and roll-up to generalize the queries over the dataset.

4.1 Knowledge-Based Data Cube Exploration

In this context, the central hub in cubing platform for each user follows the familiar dashboard metaphor. Each data cube possesses dimension-related information and measure- related information; also, each data cube represents a view of the active fact. From here, users can interact with related information while analyzing specific data cubes, retrieving a subset of the source data for the cube being analyzed, navigating to a related cube of data - a more detailed view of particular element or maybe a document describing the data or the portion of it the user indicates. Hence, the platform supports an interface that presents to the user the result of the user's request, defined by picking cube terms by hierarchical browsing or through keyword search, and users could change their view of the data of the data cube. The initial view of the ontology-based resources shows a default data source and the root terms of the different ontologies. Each time the user adjusts the query by picking one or more terms in a cube, the results table is updated showing items and detailed usage from the selected data resources.

For each of the following three food access indicators, examples of market data that would be incorporated in the indicator are given. Other useful market data should be determined according to the context.

- Food sources: Where does the food in the market come from? If it is imported - internally or from abroad - how reliable is the supply?
- Labor market: How many days per month can a casual laborer expect to find work? Is this stable?
- Price stability: Is the cost of essential food and non-food items increasing, decreasing or remaining stable in relation to normal for this time of year?

4.2 Cube Browsing Services

Source Data Retrieval. A user may want to view the source data behind a particular element presented in a Web reporting and analysis interface: a pie slice or table cell, etc. The platform is aware of its published resources, and can retrieve these datasets with appropriate parameters, which are sent by the dashboard viewer to identify the cube and detailed coordinate information about the data element the user selected. With this information, a platform component can refer back to the original data source with selection parameters to narrow the request and user parameters so that access control can be applied. The platform then has the opportunity to format the data and return it to the user.

For example, one objective of an FSA is to identify the degree - severe or moderate - of food insecurity in the area. All of these indicators are context-specific, e.g. the choice of food availability indicator implies that the area is agricultural and it would not be useful in an urban setting; or the choice of food access indicator implies that daily labor is a significant source of livelihood.

Arbitrary Cube to Cube Browsing. Cubing platform supports users in exploring more detail for a particular element in a cube reporting and data analysis session or moving to a cube in another location in the data space altogether. The inference engine turns the user's request into a particular cube selection/definition and returns that cube to the user.

Possible linkages among factors are identified during local adaptation of the cubing services. Indicators that are to be collected during semantic indexing to investigate these linkages are defined. In cross-tabulation and comparison, two or more indicators are combined to gain insights into the prevalence and causes of malnutrition and food insecurity, e.g. the link between main household income source(s) and household food sale status, i.e. Indicators to be collected for this analysis would be related to food access, food consumption and income sources.

Indicators are cross tabulated during the analysis to provide insights into the factors that affect food security status. The results are used in the response analysis. When advanced computing capacity is lacking, simple cross-tabulation of two or three variables or indicators can yield valuable information.

Prioritization of Indicators. The use of the ontology-based classification to describe current scenario indicators should be triangulated with other related indicators, such as the food and income and production indicators. However, data and indicators must be collected and analyzed carefully for each scenario. If too much information is collected, time is wasted during the data collection and analysis stages. If too little information is collected, it may be impossible to answer the assessment's key questions. In FSA context, minimum information requirements can be determined as follows [14]: Food sale is assessed from household food consumption, taking into account food access. The food consumption score should be calculated for each household with at least one relevant food access indicator should be defined.

5 Conclusions and Future Works

This paper presents the conceptual approach towards a cubing service platform that provides a flexible, cost-effective and efficient delivery platform for cubing services over open data cube marketplace. The collective warehouse resources then can be rapidly deployed and scaled based on a thorough understanding of the cube model, with all processes and services provisioned on-demand, and better align with dynamic business requirements. As a result, the future works of our approach could then be able to support users in building data cubes in cost-efficient and elastic manner that spans all aspects of cube building lifecycle, i.e. cube definition, cube computation, cube evolution as well as cube sharing [18]. To establish the practical feasibility of our approach, the implementation of tools for proposed cubing platform has been designed and is under development. It adapts its resources according to demand, allows for on-line, fast and efficient storage/processing of large amounts of data and is cost-effective over both the required hardware and software components.

References

1. Ponniah, P.: Data warehousing fundamentals: a comprehensive guide for IT professionals, vol. 1. Wiley-Interscience (2001)
2. Insightreport.com: Data Publishing Architecture for the Extended Enterprise. Technical Report (March 2003)
3. IBM Software: Build high-speed, scalable analytics into the data warehouse. Technical Report (May 2010)
4. Hung, E., Cheung, D.W., Kao, B.: Optimization in Data Cube System Design. Journal of Intelligent Information Systems 23(1), 17–45 (2004)
5. Nguyen, T.B., Tjoa, A.M., Wagner, R.: Conceptual Multidimensional Data Model Based on MetaCube. In: Yakhno, T. (ed.) ADVIS 2000. LNCS, vol. 1909, pp. 24–33. Springer, Heidelberg (2000)
6. Xin, D., Han, J., Li, X., Wah, B.W.: Star-Cubing: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration. In: VLDB, pp. 476–487 (2003)
7. Jiang, F., Pei, J.: IX-cubes: iceberg cubes for data warehousing and olap on XML data. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, pp. 1–4 (2007)
8. Messaoud, R.B.E.N., Boussaid, O., Rabaséda, S.L.: A Multiple Correspondence Analysis to Organize Data Cubes. Information Systems Frontiers 1, 133–146 (2007)
9. Pitarch, Y., Favre, C., Laurent, A., Poncelet, P.: Context-aware generalization for cube measures. In: Proceedings of the ACM 13th International Workshop on Data Warehousing and OLAP - DOLAP 2010, pp. 99–104. ACM Press (2010)
10. Balazinska, M., Howe, B.: Data Markets in the Cloud: An Opportunity for the Database Community. In: VLDB, pp. 1482–1485 (2011)
11. Virgilio, R.D., Orsi, G., Tanca, L., Torlone, R.: Semantic data markets: a flexible environment for knowledge management. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 1559–1564. ACM (2011)
12. Möller, K., Dodds, L.: The Kasabi Information Marketplace. In: WWW 2012 Developer Track, pp. 3–6 (2012)
13. DataMarket Blog: Data-as-a-Service Market Definitions (2010)
14. United Nations World Food Programme: Emergency Food Security Assessment Handbook (EFSA) - 2nd edn. Technical report (2009)
15. W3C: The RDF Data Cube Vocabulary (2012)
16. International Food Policy Research Institute (IFPRI): Global Hunger Index - RDF Version (2012)
17. Roy Chowdhury, S., Rodríguez, C., Daniel, F., Casati, F.: Wisdom-Aware Computing: On the Interactive Recommendation of Composition Knowledge. In: Maximilien, E.M., Rossi, G., Yuan, S.-T., Ludwig, H., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6568, pp. 144–155. Springer, Heidelberg (2011)
18. Nguyen, T.B., Wagner, F., Schoepp, W.: Cloud Intelligent Services for Calculating Emissions and Costs of Air Pollutants and Greenhouse Gases. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011, Part I. LNCS, vol. 6591, pp. 159–168. Springer, Heidelberg (2011)

To Approach Cylindrical Coordinates to Represent Multivariable Spatio-temporal Data

Phuoc Vinh Tran

University of Information Technology (UIT), Vietnam National University - HCMC
Phuoc.gis@gmail.com, Phuoc.gis@uit.edu.vn

Abstract. Data representing a moving object include the data of time, position, and attributes. The data of positions and attributes of a moving object, which change over time may be recorded asynchronously because of the difference of sampling methods. Mathematically, these data may be synchronized over time by space-time conversions to constitute the data tuples at various time moments. In this article, we proposed the concept of data plane to represent data according to each tuple at each time moment. Subsequently, we integrated the data planes into the dimensions of a cylindrical coordinate system to represent the movement of objects in a space-time cylinder (STCy). In a space-time cylinder, positions of moving objects are indicated on the data planes which are constituted by the cylinder axis employed as the cylindrical axis of the cylindrical coordinate system, and the polar vectors of the cylindrical coordinate system. Each data plane indicates the data of objects at a time moment. The position of a moving object at a time moment is indicated by its coordinates on the data plane and the time moment by the angular coordinate of this plane. The attributes of moving objects are represented on data planes as the attribute bars parallel to the cylinder axis. The space-time path of a moving object surrounds the cylinder axis. Hence, the space-time cylinder is consistent with the representation of cyclic movements.

Keywords: space-time cylinder, spatio-temporal data, movement data, visualization.

1 Introduction

Three main components of the real world, object, space, and time are described in the triad of “what”, “where”, “when” by Peuquet [16],[17], and analyzed further by Andrienko in the triad of “objects”, “locations”, and “times” [1],[2]. These analyses mentioned the individual characteristics of sets of objects, locations, and times, the relations between elements of a set and the relations between elements of different sets. These relations classify objects as spatial objects, temporal objects, spatio-temporal objects, or moving objects according to the relations of objects with locations, objects with times, objects with locations and times, objects with locations, times, and trajectories, respectively.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 21–28, 2012.

The movement of an object is depicted by the continuous change of the position of the object through space. Proposed by Hargertrand in 1970 [10], the Cartesian coordinate system of three dimensions is employed as a space-time cube to represent the data of positions of moving objects over time. In the coordinate system, the data of positions of moving objects are indicated by their coordinates (x, y) at each time moment t. The space-time cube has been employed to represent movement data because it visualizes the change of the moving objects' positions over time. Space-time paths or temporal trajectories are the curves representing the relations between space and time of moving objects [2-4],[6],[9-10],[16]. A challenge is how to represent the attributes of moving objects over time in a space-time cube. Some authors have expanded the space-time cube to represent the attributes of moving objects over time. The expansions integrated the parallel coordinates into a cube to represent the attributes of moving objects. For unmoving objects, it is possible to represent the positions and attributes on only one cube [18],[19]. For moving object, it is possible to represent the positions and attributes on two cubes [14], or integrate the positions and attributes on one cube [20],[22].

The main idea of this article is to represent the data of positions and attributes of moving objects at each time moment on the same plane, called data plane. The methods recording data provide with the data of position and the data of attributes of a moving object at each time moment [7],[15]. Each data tuple indicates the data of the positions and the attributes of moving objects at the same time moment. On a data plane, the positions of moving objects are referred to their coordinates (x, y) on the axes of the plane, and the attributes are indicated by the bars parallel to one of the axes of the plane.

A subsequent idea is to approach the cylindrical coordinates to representing the data planes as a spatio-temporal cylinder. In a cylindrical coordinate system, angular coordinates indicate the time data of the data tuples, positions on the cylindrical axis and magnitudes of polar vectors indicate the position data of the data tuples, the bars parallel to the cylindrical axis indicate the attribute data.

The paper is structured as follows. In the item 2, we briefly present related researches and conceptual framework employed in the article; in the item 3, we propose the model of data plane to represent the data of objects at a time moment; in the item 4, we approach the cylindrical coordinate system to representing multivariable spatio-temporal data in spatio-temporal cylinders. The modes of cylinders represent data in different cases. The static mode of cylinder represents the data of moving objects during the entire movement period. The dynamic mode of the cylinder revives the activities of objects implicit in data. The hide mode of the cylinder is employed to stand out the data of movements.

2 Conceptual Framework and Related Works

Movement is the change of the position of an object over time [1][7]. An object of which existing position changes continuously is called a moving object. The positions of a moving object are indicated by its coordinates (x, y) in the 2-D domain of the

observed area. The curve time-ordered connecting the positions of the coordinates (x, y) where the moving object visited is called the trajectory [6].

The time is indicated on the time axis. Time moments are indicated as points on the time axis t. Time intervals are indicated as segments on the time axis, from a point t_i to a point t_j, where $i, j \in \{0,1,2,..\}$, symbolized by Δt. The position of a moving object is a function (mapping) from time to position: $T : t \to (x, y)$, or $T(t) = (x, y)$, and the tuple (x, y, t) is spatio-temporal data of the moving object. The curve $T = (x, y, t)$ time-ordered connecting the points (x, y, t) of a moving object in the 3-D domain is the space-time path or the temporal trajectory of the object.

Each object has its thematic attributes [2],[7],[14],[15],[20]. The attributes of an object can change over time. Some attributes of an unmoving object also change over time (e.g. a gauge station is an unmoving object, the values recorded by the sensors at the station are attributes changing over time [19]). Meanwhile, some attributes of a moving object change over time (e.g. a bus is a moving object and its passengers are an attribute changing on its route; a vehicle is a moving object and its goods is an attribute changing on its route [20]). Attributes are recorded by different sampling methods may be synchronized over time by inferring from the temporal trajectory $T = (x, y, t)$ of the moving object.

The data of a moving object include the data of positions and attributes changing over time [1],[7],[13],[19],[20]. Movement data is a set of multivariable spatio-temporal data of moving objects including data of positions and attributes, which change over time. The movement data are depicted by a table including several data records of positions and attributes at various time moments.

3 Data Plane

A movement is a continuous activity over time. However, data of a movement are recorded discretely at various time moments according to its sampling period. At each

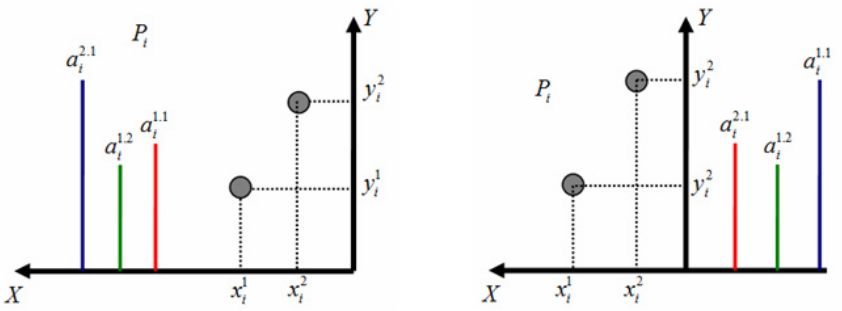


Fig. 1. The data plane P_i at t_i

sampling time, the data of positions and attributes of moving objects constitute a tuple <identifiers, time, positions, attributes>. In the data table of moving objects, each tuple is represented as a data record $(o^k, t_i, x_i^k, y_i^k, a_i^{k.m})$, where $(x_i^k.y_i^k)$ is the position of the object o^k at the time moment t_i, and $a_i^{k.m}$ are the attributes a^m of o^k at t_i

In this article, we employ planes of 2-D domain to represent the data tuples at various time moments, called data plane P_i (figure 1). A data plane refers to a plane representing data of positions and attributes of moving objects at a time moment. In a data plane, the two axes of the plane indicate the positions of objects, the bars parallel to an axis indicate the attributes of objects. The height of an attribute bar on the data plane is in proportion to the value of the attribute at the time moment of the data plane. Accordingly, all data concerning with moving objects at a time moment are represented on a data plane. In other words, the data of each record on the data table are converted into a data plane.

4 Space-Time Cylinder for Visualization

4.1 Space-Time Cylinder

We propose a novel approach to representing visually spatio-temporal data based on cylindrical coordinates. This approach is called space-time cylinder (figure 2). A cylindrical coordinate system consists of three dimensions: the cylindrical axis, polar vectors starting at and perpendicular to the cylindrical axis, and angular coordinates constituted by different polar vectors and the original polar vector. For a space-time cylinder, the dimensions of a cylindrical coordinate system are assigned to the cylinder as follows. The cylindrical axis is assigned to the axis of the cylinder, the position coordinates x and y of moving objects are indicated by the magnitudes of polar vectors and the axial positions on the cylindrical axis, and the times t are indicated by angular coordinates α, where $0 \le \alpha \le 2\pi$.

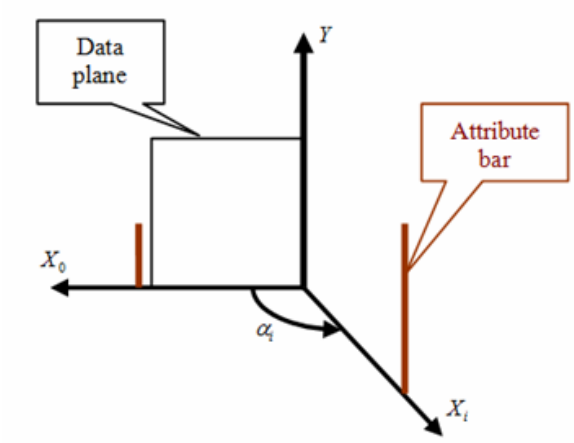


Fig. 2. A cylindrical coordinate system to represent multivariable spatio-temporal data

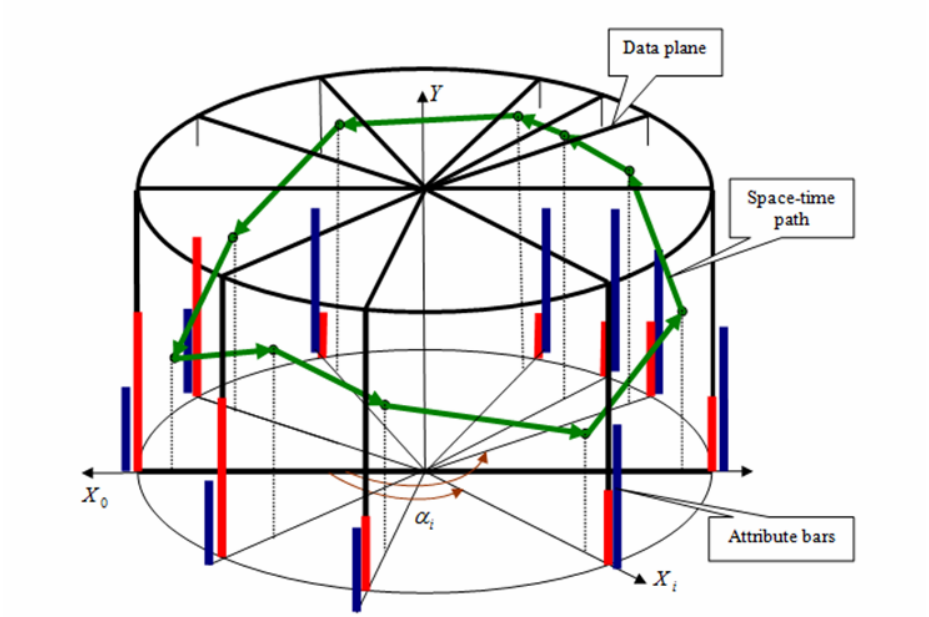


Fig. 3. Space-time cylinder for representing multivariable spatio-temporal data

The data planes of moving objects in a space-time cylinder are made up by the cylindrical axis and polar vectors. The time moments of the data planes are in proportion to angular coordinates α of the polar vectors. Each position of the data plane P_i at t_i is determined by an angle α_i formed by the plane P_0 at t_0 and the plane P_i at t_i. The attribute bars on the data planes make up the surfaces of the cylinder. Accordingly, each data plane in a space-time cylinder represents the positions and attributes of moving objects at a time moment. In other words, each data plane represents all data of a record of the data table. The curve time-ordered connecting the positions of a moving object on the data planes is the temporal trajectory or the space-time path T of the object. The temporal trajectories of moving objects surround the cylinder axis (figure 3).

We considered that several movements are cyclic, a moving object departs from a place to visit one or many places and turn back the departure place (e.g. buses depart from their departure station to visit several bus stops to pick up and drop out their passengers and return departure station, workers leave their home in the morning for their offices and come back home in the evening). In a space-time cylinder, the temporal trajectory of a moving object is a curve time-ordered connecting the object positions on data planes. For a cyclic movement, the ending position of the route on the plane of $t_i = 2\pi$ fits in with its departure position on the first data plane of $t_i = 0$. Accordingly, the space-time cylinder is consistent with the representation of the multivariate data of cyclic movements.

4.2 Modes of Space-Time Cylinder for Data Geo-visualization

Mode of Static Visualization. For the static mode of a space-time cylinder, all data planes of moving objects at all time moments are displayed (figure 3). In other words, all data of the table are shown completely. To represent an available data table with space-time cylinder, the data planes are designed so that the number of data planes is equal to the number of data records of the table. Each data plane represents all data fields of one record on the data table. The angular coordinate α of each data plane is in proportion to the time moment of the record. The positions of data planes in the cylinder are determined by their angular coordinates α, which are so calculated that the entire movement period of moving objects fits in with 2π, the maximum of the angle α.

Mode of Dynamic Visualization. In the dynamic mode of a space-time cylinder, each data plane of moving objects at a time moment is shown one after another in time line (figure 4). A cursor moves slowly with the automatic or manual control on a time axis to display data planes. When the cursor moves from starting time to ending time of the time axis, each data plane is shown each time the cursor reaches a time point of the plane. On the contrary, when the cursor moves from ending time to starting time of the time axis, each data plane is hide each time the cursor reaches a time point of the plane. In the dynamic mode, the data plane at t_0 rotates around the cylinder axis each time the cursor moves from a time moment to another, the data plane corresponding to the time moment of the cursor is always shown at the position perpendicular to the user's view.

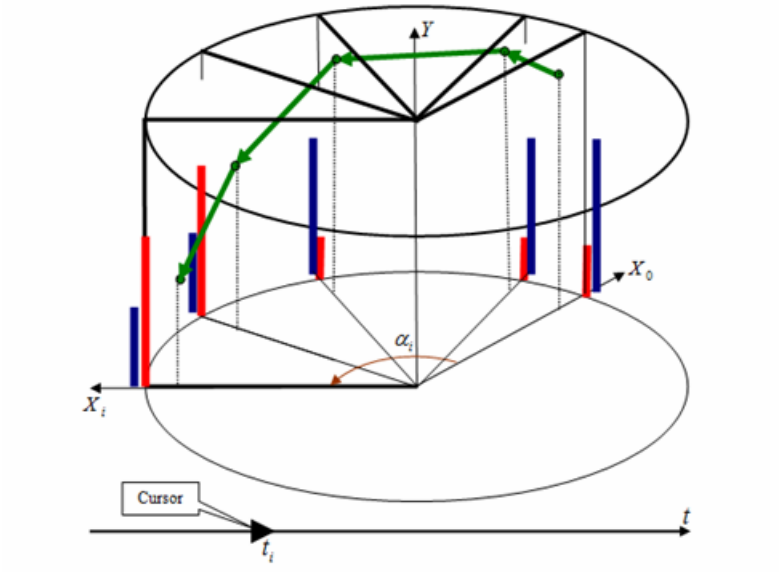


Fig. 4. The dynamic mode of a space-time cylinder

Mode of Hide Visualization. The hide mode of space-time cylinder is applied for the cases of overcrowded data on screen. The goal of the hide mode is to only visualize the data necessary for users. We consider that there are a lot of spatial data displaying repeatedly on all data planes. Data of geographic area and frames of data planes are shown on all data planes of the cylinder. In many cases, they are not really necessary to be displayed on all data planes. Only spatial data different from the last plane should be shown on each data plane. When the hide mode of a space-time cylinder is turned on, the repeated data on data planes of $i \neq 0$ are filtered and only the positions and attributes of moving objects are displayed on data planes of $i \neq 0$.

5 Conclusion

In this article, we proposed the approach of the concept of data planes to represent visually the data of positions and attributes of moving objects at different time moments. Movement data including the data of positions and attributes of moving objects are recorded discretely at various time moments. Each data tuple of the movement at a time moment is recorded as a record on a data table. Each data record, including data of positions and attributes, is represented on a data plane. We also proposed to employ cylindrical coordinates to represent the data of moving objects by arranging the data planes around the axis of a cylinder, where the angular coordinates of the data planes are in proportion to their times. The space-time cylinder is consistent with the representation of the multivariate data of cyclic movements.

Acknowledgements. I am very grateful to the Advanced Program of the University of Information Technology (UIT), Vietnam National University – HCMC, for its valuable grant to create this article.

References

1. Andrienko, N., Andrienko, G.: Visual analytics of movement: an overview of methods, tools, and procedures (2012)
2. Andrienko, G., Andrienko, N., Bak, P., Keim, D., Kisilevich, S., Wrobel, S.: A conceptual framework and taxonomy of techniques for analyzing movement. Journal of Visual Languages and Computing 23, 213–232 (2011)
3. Andrienko, G., Andrienko, N., Keim, D., MacEachren, A.M., Wrobel, S.: Challenging problems of geospatial visual analytics. Editorial/Journal of Visual Languages and Computing 22, 251–256 (2011)
4. Andrienko, G., Andrienko, N., Demsar, U., Dransch, D., Dykes, J., Fabrikant, S.I., Jern, M., Kraak, M.J., Schumann, H., Tominski, C.: Space, time and visual analytics. International Journal of Geographical Information Science 24(10), 1577–1600 (2010)
5. Andrienko, G., Andrienko, N.: Dynamic Time Transformation for Interpreting Clusters of Trajectories with Space-Time Cube. In: IEEE Symposium on Visual Analytics Science and Technology, Poster (2010)

6. Andrienko, G., Andrienko, N.: Visual Analytics for Geographic Analysis, Exemplified by Different Types of Movement Data. In: Information Fusion and Geographic Information Systems, Part 1. Lecture Notes in Geoinformation and Cartography, pp. 3–17 (2009)
7. Andrienko, N., Andrienko, G., Pelekis, N., Spaccapietra, S.: Basic concepts of movement data. In: Giannotti, F., Pedreschi, D. (eds.) Mobility, Data Mining and Privacy, Geographic Knowledge Discovery, pp. 15–38. Springer (2008)
8. Andrienko, N., Andrienko, G., Gatalsky, P.: Exploratory spatio-temporal visualization: an analytical review. Journal of Visual Languages and Computing, Special Issue on Visual Data Mining 14(6), 503–541 (2003)
9. Dodge, S., Weibel, R., Lautenschütz, A.-K.: Towards a Taxonomy of Movement Patterns. Information Visualization 2008(7), 240–252 (2008)
10. Gatalsky, P., Andrienko, N., Andrienko, G.: Interactive Analysis of Event Data Using Space-Time Cube. In: Proceedings of the Eighth International Conference on Information Visualisation (IV 2004). IEEE Computer Society (2004)
11. Hagerstrand, T.: What about people in regional science? Papers of Ninth European Congress of Regional Science Association, vol. 24, pp. 7–21 (1970)
12. Kraak, M.J.: The Space-Time Cube Revisited from a Geovisualization Perspective. In: Proceedings of the 21st International Cartographic Conference (ICC) "Cartographic Renaissance", pp. 1988–1996 (2003)
13. Keim, D.A., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G.: Visual Analytics: Definition, Process, and Challenges. In: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (eds.) Information Visualization. LNCS, vol. 4950, pp. 154–175. Springer, Heidelberg (2008)
14. Li, X., Kraak, M.J.: New views on multivariable spatiotemporal data: the space time cube expanded. In: International Symposium on Spatio-temporal Modelling, Spatial Reasoning, Analysis, Data Mining and Data Fusion, vol. XXXVI, pp. 199–201 (2005)
15. Willems, N., van Hage, W.R., de Vries, G., Janssens, J.H.M., Malais, V.: An integrated approach for visual analysis of a multi-source moving objects knowledge base. International Journal of Geographical Information Science 24(9), 1–16 (2010)
16. Peuquet, D.J.: It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. Annals of the Association of American Geographers 84(3), 441–461 (1994)
17. Peuquet, D.J.: Representations of Space and Time. Guilford, New York (2002)
18. Tominski, C., Schulze-Wollgast, P., Schumann, H.: 3D Information Visualization for Time Dependent Data on Maps. In: Proceedings of the International Conference on Information Visualization (IV), pp. 175–181. IEEE Computer Society (2005)
19. Phuoc, T.V., Hong, N.T.: An Integrated Space-Time-Cube as a Visual Warning Cube. In: Proceedings of 3rd International Conference on Machine Learning and Computing, vol. 4, pp. 449–453. IEEE (2011)
20. Phuoc, T.V., Hong, N.T.: Visualization Cube for Tracking Moving Object. In: Proceedings of Computer Science and Information Technology, Information and Electronics Engineering, vol. 6, pp. 258–262. IACSIT Press (2011)
21. Li, X., Kraak, M.-J.: A temporal visualization concept: A new theoretical analytical approach for the visualization of multivariable spatio-temporal data. In: 18th International Conference on Geoinformatics, pp. 1–6 (2010), doi: 10.1109/GEOINFORMATICS.2010.5567529
22. Song, Y., Miller, H.J.: Exploring traffic flow databases using space-time plots and data cubes. Transportation 39(2), 215–234 (2012)

EFP-M2: Efficient Model for Mining Frequent Patterns in Transactional Database

Tutut Herawan[1], A. Noraziah[1], Zailani Abdullah[2],
Mustafa Mat Deris[3], and Jemal H. Abawajy[4]

[1] Faculty of Computer System and Software Engineering
Universiti Malaysia Pahang
[2] Department of Computer Science
Universiti Malaysia Terengganu
[3] Faculty of Science Computer and Information Technology
Universiti Tun Hussein Onn Malaysia
[4] Scholl of Information Technology
Deakin University
{tutut,noraziah}@ump.edu.my, zailania@umt.edu.my,
mmustafa@uthm.edu.my, jemal.abawajy@deakin.edu.au

Abstract. Discovering frequent patterns plays an essential role in many data mining applications. The aim of frequent patterns is to obtain the information about the most common patterns that appeared together. However, designing an efficient model to mine these patterns is still demanding due to the capacity of current database size. Therefore, we propose an Efficient Frequent Pattern Mining Model (EFP-M2) to mine the frequent patterns in timely manner. The result shows that the algorithm in EFP-M2l is outperformed at least at 2 orders of magnitudes against the benchmarked FP-Growth.

Keywords: Model, Frequent patterns, Data mining, Efficient.

1 Introduction

Mining patterns or Association Rules (AR) is important and established topic in data mining. It is a basic step in finding the associations among items (parameters or values). For example, the retail transaction is aimed at searching the association between the most frequent items that are bought together. By understanding the customers' behavior, it can help the management to design promotional strategies, determine potential buyers, increase profit-sales etc. This type of pattern is also known as frequent patterns. Apriori [1] was the first algorithm to capture sets of frequently bought products at stores. In AR, a set of item is defined as an itemset. The itemset is said to be frequent, if it occurs more than a predefined minimum support. In addition, confidence is another alternative measurement that always used in pair with support threshold. The AR is said to be strong if it meets the minimum confidence.

Until this recent, several works have been put forward in mining the frequent patterns [16-22]. Frequent pattern tree (FP-Tree) [2] has become one of the great

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 29–38, 2012.

alternative data structure to represent the vast amount of transactional database in compressed manner. Afterwards, numerous enhancements of FP-Tree have been suggested according to the implementation of multiple and single database scans. For the first category and including FP-Tree [2], the related studies are Ascending Frequency Ordered Prefix-Tree (AFOPF) [3], Adjusting FP-Tree for Incremental Mining (AFPIM) [4] and Extending FP-tree for Incremental Mining (EPFIM) [5]. The related researches in the second category are Compressed and Arranged Transaction Sequence (CATS) tree [6], Fast Updated FP-Tree (FUFPT) [7], Branch Sorting Method (BSM) [8] and Batch Incremented Tree (BIT) [9].

However, there are still two major shortcomings encountered. First, the construction of FP-Tree is still relied on extracting the patterns that fulfils the support threshold from the original databases. Second, if the existing databases are suddenly updated, the current FP-Tree must be rebuilt again from the beginning because of the invalidity of the items supports. In some research extensions, the structure of FP-Tree will be reorganized extensively due to the modification of databases. Therefore, highly computational cost in constructing FP-Tree is still an outstanding issue in mining the frequent patterns.

Therefore, in this paper we proposed an Efficient Frequent Pattern Mining Model (EFP-M2) to alleviate the mentioned above problems. The performance evaluation of the model is made based on two benchmarked datasets from UCI Data Repository.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 explains the details of the proposed method. This is followed by the comparison tests in section 4. Finally, conclusion and future direction are reported in section 5.

2 Related Work

Frequent patterns mining plays a fundamental role in data mining and has been received many attentions for the past decade. More than hundreds of papers have been published in an attempt to increase its efficiencies via enhancement or new algorithms developments. It was first introduced by Agrawal [1] to mine the ARs between items and also known as market basket analysis. Besides ARs, it also reveals the strong rules, correlation, sequential rules, causality, and many other important discoveries [23-30].

There are two important reasons of finding frequent patterns from data repositories. First, frequent patterns can effectively summarize the underlying datasets, and provide new information about the data. These patterns can help the domain experts to discover new knowledge hiding in the data. Second, frequent pattern serves as the basic input for others data mining tasks including association rule mining, classification, clustering, and change detection, and etc [10-13]. In real world, mining the frequent itemset may involve with the massive dataset and highly pattern dimensions. Therefore, minimizing the computational cost and ensuring the high efficiency in mining activities are very important. Hence, numerous strategies and improvement of data structures have been put forward until this recent.

3 Proposed Method

Throughout this section the set $I=\{i_1,i_2,\cdots,i_{|A|}\}$, for $|A|>0$ refers to the set of literals called set of items and the set $D=\{t_1,t_2,\cdots,t_{|U|}\}$, for $|U|>0$ refers to the data set of transactions, where each transaction $t\in D$ is a list of distinct items $t=\{i_1,i_2,\cdots,i_{|M|}\}$, $1\le|M|\le|A|$ and each transaction can be identified by a distinct identifier TID.

3.1 Definition

Definition 1. (Frequent Items). *An itemset X is called frequent item if* $\text{supp}(X)>\beta$, *where* β *is the minimum support.*

The set of frequent item will be denoted as Frequent Items and

$$\text{Frequent Items}=\{X\subset I\mid\text{supp}(X)>\beta\}$$

Definition 2. *Disorder Support Trie Itemset (DOSTrieIT) is defined as a complete tree data structure in canonical order of itemsets. The order of itemset is not based on the support descending order. DOSTrieIT contains n-levels of tree nodes (items) and their support. Moreover, DOSTrieIT is constructed in online manner and for the purpose of incremental pattern mining.*

Example 1. *Let* $T=\{\{1,2,5\},\{2,4\},\{2,3\},\{1,2,4\},\{1,3\},\{2,3,6\},\{1,3\},\{1,2,3,5\},\{1,2,3\}\}$. Graphically, an item is represented as a node and its support is appeared nearby to the respective node. A complete structure of DOSTrieIT is shown in Figure 1.

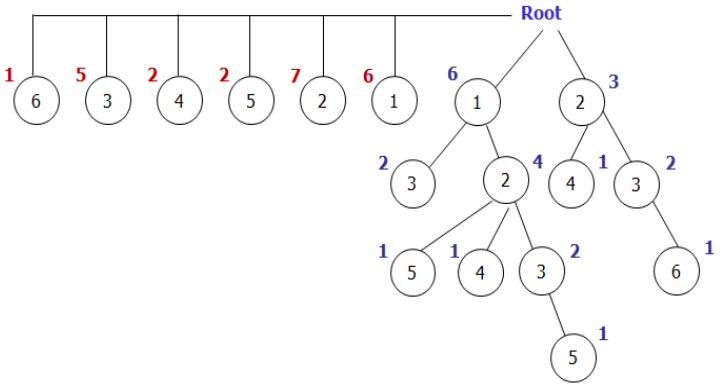


Fig. 1. DOSTrieIT

Definition 3. *Single Item without Extension (SIWE) is a prefix path in the tree that contains only one item or node. SIWE is constructed upon receiving a new transaction and as a mechanism for fast searching of single item support. It will be employed during tree transformation process but it will not be physically transferred into the others tree.*

Example 2. From Example 1, the transactions have 6 unique items and it is not sorted in any order. In Fig. 1, SIWE for DOSTrieIT i.e.,

$$\text{SIWE} = \{2,1,3,4,5,6\}$$

3.2 Efficient Frequent Patterns Mining Model

There are four major components involved in designing the EFP-M2. These components are interrelated and the process flow is moving in one-way direction. An overview model of efficiently constructing frequent pattern tree is shown in Fig. 2.

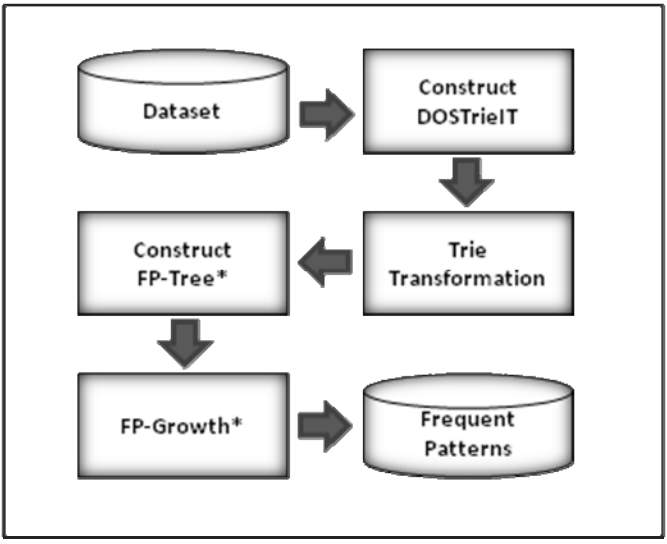


Fig. 2. An Overview of EFP-M2

(i) Dataset: In this model, dataset is in a flat file format. Each data (item) is separated by a single space. The datasets used for the empirical analysis are downloaded from UCI Machine Learning Repositories.
(ii) Construct DOSTrieIT: The first component in this model is to scan dataset and finally convert into DOSTrieIT data structure. The first sub-process involved is called Load Line of Transaction (Online). Once a new line of transaction is triggered, DOSTrieIT data structure is immediately constructed. In other words,

it is an online or instant tree construction. Generate Prefix Path is the second sub-process at this level. Items in the line of transaction are separated to form a vertical format of an itemset. The itemset is then transformed into DOSTrieIT. The third sub-process is Extend Prefix Path. The size of the existing prefix paths in terms of the nodes may involve with some modification. It is depend on a new arrival of prefix path. Update Latest Support is the fourth sub-process. Items supports for the existing prefix path may or may not update and it is based on the characteristic of a new prefix path. The final sub-process is Construct DOSTrieIT. DOSTrieIT is a complete tree data structure and aimed for incremental mining. It is automatically updated once the transactions from original dataset are modified.

(iii) Trie Transformation: The second component in this model is to scan all prefix paths in DOSTrieIT and convert into particular prefix paths before they can be used for constructing FP-Tree. The first sub-process involved is called Load DOSTrieIT (Online). All prefix paths in DOSTrieIT and its *SIWE* are loaded. The second sub-process is Determine Minimum Itemset Support. A support value to represent all items in intersected itemset is determined from its minimum items supports. Generate Frequent Itemset is the third sub-process. This itemset is extracted from *SIWE* and based on the predefined minimum support threshold. The fourth sub-process is Sort Frequent Itemset. The frequent itemset is sorted in support descending order. Intersection Operation is the fourth sub-process. All itemsets involved in this operation are converted to hash based data structure. The last sub-process is Intersected Itemset. The output from the previous intersection operation is an intersected Itemset. The order of the items in intersected itemset is following the items in frequent itemset

(iv) Construct FP-Tree*: The third component is to construct FP-Tree based on the particular prefix paths format supplied from the previous component. The final structure of FP-Tree* is similar to FP-Tree but it is different in term of input data. FP-Tree* uses the input data from DOSTrieIT rather than original input data. The first sub-process involved is called Generate Prefix Path. Items in the prefix paths are separated to form a vertical format of an itemset The itemset is then transformed into FP-Tree. The second sub-process is Extend Prefix Path. The size of the existing prefix path may involve with some modification. It is depend on a new arrival of prefix path. Update Latest Support is the third sub-process. Items supports for the existing prefix path may or may not update and it is based on the characteristic of a new prefix path. The final sub-process is Construct FP-Tree. This process will continue until entire prefix paths in DOSTrieIT are transformed into FP-Tree. FP-Tree data structure will be transformed into flat-file.

(v) FP-Growth*: The fourth component is to mine the complete set of frequent patterns from FP-Tree* by patterns fragment-growth and without using candidate itemsets generations. The implementation of FP-Growth* is similar to standard FP-Growth [14]. The first sub-process involved is called Load FP-Tree (Online). All prefix paths in FP-Tree are scanned and loaded into memory for further processing. Determine Conditional Patterns base is the second sub-process at this level. All prefix paths that ending with a particular suffix are selected. It is known as conditional pattern based. The third sub-process is Determine Conditional FP-Tree. All conditional pattern based are then converted into conditional FP-Tree

with respect to that particular suffix (item). This tree structure is similar to FP-Tree and it will be used to find frequent itemsets with that suffix. Generate Frequent Pattern is the fourth sub-process at this level. All possible combination of items containing a particular suffix will be extracted until finished. This process will recursively execute until no more conditional pattern base can be built. Frequent patterns are then counted and generated.

(vi) Frequent Patterns: All frequent patterns are stored in flat file for post-processing such to generate AR with different values of measurements, etc.

The pseudocode for constructing DOSTrieIT, implementing Trie Transformation and executing FP-Growth* are shown in Fig. 3, 4 and 5, respectively.

Pseudocode DOSTrieIT
Input: Transaction data
Output: DOSTrieIT
1. **Begin**
2. Load line of transaction
3. Generate prefix paths
4. Extend prefix paths
5. Update latest support
6. Construct DOSTrieIT
7. **End**

Fig. 3. Steps in generating DOSTrieIT

Pseudocode Trie Transformation
Input: DOSTrieIT
Output: FP-Tree
1. **Begin**
2. Load DOSTrieIT
3. Generate frequent patterns
4. Sort frequent patterns
5. Intersection operations
6. Intersected Itemset
7. Determine min itemset supp
8. **End**

Fig. 4. Steps in implementing trie transformation

Pseudocode FP-Growth*
Input: FP-Tree
Output: Significant patterns (frequent)
1. **Begin**
2. Load FP-Tree
3. Determine Conditional Items
4. Determine Conditional FP-Tree
5. Generate frequent patterns
6. **End**

Fig. 5. Steps in executing trie transformation

4 Comparison Tests

In this section, we do comparison tests between benchmarked FP-Growth and our FP-Growth*. The performance analysis is made by comparing the computational time required to completely mine the frequent patterns. We conducted our experiment in two benchmarked datasets. The experiment has been performed on Intel® Core™ 2 Quad CPU at 2.33GHz speed with 4GB main memory, running on Microsoft Windows Vista. All algorithms have been developed using C# as a programming language.

Two benchmarked datasets from Frequent Itemset Mining Dataset Repository [15] were employed in the experiment. The first dataset was Retails and it contains the retail market basket data from an anonymous Belgian retail store. For the second experiment, Pumsb dataset was used. The Pumsb dataset contains census data for population and housing. Table 1 shows the fundamental characteristics of the datasets.

Table 1. Fundamental Characteristics of Datasets

Data sets	Size	#Trans	#Items	Average length
Retails	4.153 MB	88,136	16,471	10
Pumsb	16.59MB	49,046	2,113	74

There were variety of minimum supports were employed in the experiment. Duration in millisecond and in Logarithmic scale view was employed in the graph. The performance comparison between FP-Growth* and FP-Growth against Retails dataset is presented in Fig. 6. From the graph, FP-Growth* algorithm is faster at 944.23 times (99.89%) than FP-Growth in mining the frequent patterns. In other

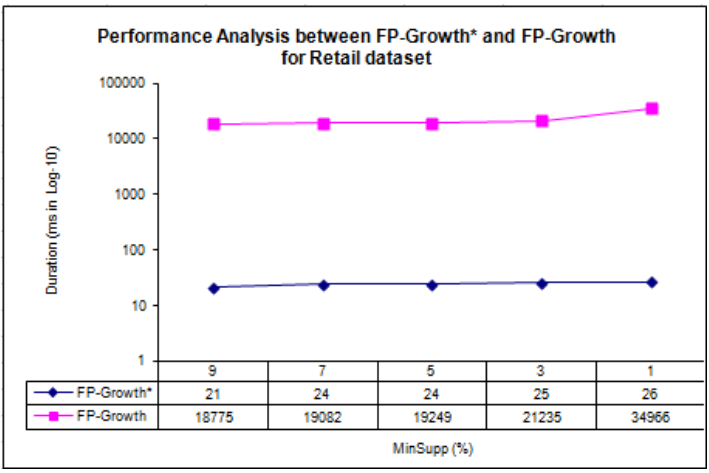


Fig. 6. Performance Analysis for both algorithms against Retails dataset

words, FP-Growth* is almost 3 orders of magnitude better than FP-Growth. Typically, the times consumed were decreased when the *Supp* values (minimum support) were increased. The main reason is the total frequent patterns being mined by the both algorithms are inversely proportional with the processing time.

Fig. 7 shows the performance of both algorithms against Pumsb dataset. From the graph, FP-Growth* is 193.00 times (99.48) fastest than FP-Growth in mining the frequent patterns. In other word, FP-Growth* is about 2 orders of magnitudes better than FP-Growth. Similarly with Retails dataset, the processing times for Pumsb dataset were decreased when the Supp values (minimum support) were increased. It is because the total frequent patterns being mined are inversely proportional with the processing time.

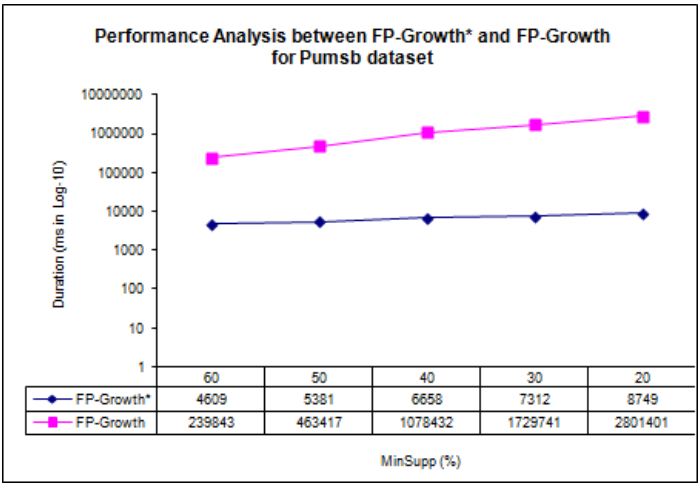


Fig. 7. Performance Analysis for both algorithms against Pumsb dataset

5 Conclusion

Frequent pattern mining is a core and one of the active research themes in data mining. It aims at discovering the patterns that frequently appear together in the transactional database. Since the generation of frequent patterns is computational extensive, thus an efficient model has become a necessity. Therefore, we propose we propose a new model called Efficient Frequent Pattern Mining Model (EFP-M2) to mine the frequent patterns in timely manner. The experimental result shows that the algorithm in EFP-M2 is outperformed at least at 2 order of magnitudes against the benchmarked FP-Growth.

In a near future, we are going to evaluate the performance of FP-Growth* by applying into several real datasets and extended to mine the frequent patterns.

Acknowledgement. This work is supported by the research grant from Research Management Center of Universiti Malaysia Pahang.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Database Mining: A Performance Perspective. IEEE Transactions on Knowledge and Data Engineering 5(6), 914–925 (1993)
2. Han, J., Pei, H., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proc. of the 2000 ACM SIGMOD, pp. 1–12. ACM, Texas (2000)
3. Liu, G., Lu, H., Lou, W., Xu, Y., Yu, J.X.: Efficient Mining of Frequent Patterns using Ascending Frequency Ordered Prefix-Tree. Data Mining and Knowledge Discovery 9, 249–274 (2004)
4. Koh, J.-L., Shieh, S.-F.: An Efficient Approach for Maintaining Association Rules Based on Adjusting FP-Tree Structures. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, pp. 417–424. Springer, Heidelberg (2004)
5. Li, X., Deng, Z.-H., Tang, S.: A Fast Algorithm for Maintenance of Association Rules in Incremental Databases. In: Li, X., Zaïane, O.R., Li, Z. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 56–63. Springer, Heidelberg (2006)
6. Cheung, W., Zaïane, O.R.: Incremental Mining of Frequent Patterns without Candidate Generation of Support Constraint. In: Proc. of the 7th International Database Engineering and Applications Symposium (IDEAS 2003), pp. 111–117. IEEE Computer Society, New York (2003)
7. Hong, T.-P., Lin, J.-W., We, Y.-L.: Incrementally Fast Updated Frequent Pattern Trees. An International Journal of Expert Systems with Applications 34(4), 2424–2435 (2008)
8. Tanbeer, S.K., Ahmed, C.F., Jeong, B.-S., Lee, Y.-K.: CP-Tree: A Tree Structure for Single-Pass Frequent Pattern Mining. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 1022–1027. Springer, Heidelberg (2008)
9. Totad, S.G., Geeta, R.B., Reddy, P.P.: Batch Processing for Incremental FP-Tree Construction. International Journal of Computer Applications 5(5), 28–32 (2010)
10. Huan, J., Wang, W., Bandyopadhyay, D., Snoeyink, J., Prins, J., Tropsha, A.: Mining Protein Family-Specific Residue Packing Patterns from Protein Structure Graphs. In: Proc. 8th International Conference on Research in Computational Molecular Biology (RECOMB), pp. 308–315. ACM Press (2004)
11. Inokuchi, A., Washio, T., Motoda, H.: An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 13–23. Springer, Heidelberg (2000)
12. Jin, R., Agrawal, G.: A Systematic Approach for Optimizing Complex Mining Tasks on Multiple Datasets. In: Proc. the 22nd International Conference on Data Engineering, pp. 1–17. IEEE Press (2006)
13. Zaki, M.J., Gouda, K.: Fast Vertical Mining using Diffsets. In: Proc. the of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 326–335. ACM Press (2003)
14. Han, J., Pei, J.: Mining Frequent Pattern without Candidate Itemset Generation: A Frequent Pattern Tree Approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
15. Frequent Itemset Mining Dataset Repository, `http://fimi.cs.helsinki.fi/data/`
16. Abdullah, Z., Herawan, T., Noraziah, A., Deris, M.M.: Extracting Highly Positive Association Rules from Students' Enrollment Data. Procedia Social and Behavioral Sciences 28, 107–111 (2011)

17. Abdullah, Z., Herawan, T., Noraziah, A., Deris, M.M.: Mining Significant Association Rules from Educational Data using Critical Relative Support Approach. Procedia Social and Behavioral Sciences 28, 97–101 (2011)
18. Yun, U., Ryu, K.H.: Approximate weighted frequent pattern mining with/without noisy environments. Knowledge-Based Systems 24, 73–82 (2011)
19. Duraiswamy, K., Jayanthi, B.: A Novel preprocessing Algorithm for Frequent Pattern Mining in Multidatasets. International Journal of Data Engineering (IJDE) 2(3), 111–118 (2011)
20. Leung, C.K.-S., Jiang, F.: Frequent Pattern Mining from Time-Fading Streams of Uncertain Data. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 252–264. Springer, Heidelberg (2011)
21. Leung, C.K.-S., Jiang, F., Hayduk, Y.: A landmark-model based system for mining frequent patterns from uncertain data streams. In: Proc. IDEAS 2011, pp. 249–250. ACM Press (2011)
22. Abdullah, Z., Herawan, T., Deris, M.M.: Scalable Model for Mining Critical Least Association Rules. In: Zhu, R., Zhang, Y., Liu, B., Liu, C. (eds.) ICICA 2010. LNCS, vol. 6377, pp. 509–516. Springer, Heidelberg (2010)
23. Abdullah, Z., Herawan, T., Deris, M.M.: Mining Significant Least Association Rules Using Fast SLP-Growth Algorithm. In: Kim, T.-H., Adeli, H. (eds.) AST/UCMA/ISA/ACN 2010. LNCS, vol. 6059, pp. 324–336. Springer, Heidelberg (2010)
24. Abdullah, Z., Herawan, T., Deris, M.M.: An Alternative Measure for Mining Weighted Least Association Rule and Its Framework. In: Zain, J.M., Wan Mohd, W.M.B., El-Qawasmeh, E. (eds.) ICSECS 2011, Part II. CCIS, vol. 180, pp. 480–494. Springer, Heidelberg (2011)
25. Abdullah, Z., Herawan, T., Deris, M.M.: Visualizing the Construction of Incremental Disorder Trie Itemset Data Structure (DOSTrieIT) for Frequent Pattern Tree (FP-Tree). In: Badioze Zaman, H., Robinson, P., Petrou, M., Olivier, P., Shih, T.K., Velastin, S., Nyström, I. (eds.) IVIC 2011, Part I. LNCS, vol. 7066, pp. 183–195. Springer, Heidelberg (2011)
26. Herawan, T., Yanto, I.T.R., Deris, M.M.: Soft Set Approach for Maximal Association Rules Mining. In: Ślęzak, D., Kim, T.-H., Zhang, Y., Ma, J., Chung, K.-I. (eds.) DTA 2009. CCIS, vol. 64, pp. 163–170. Springer, Heidelberg (2009)
27. Herawan, T., Yanto, I.T.R., Deris, M.M.: SMARViz: Soft Maximal Association Rules Visualization. In: Badioze Zaman, H., Robinson, P., Petrou, M., Olivier, P., Schröder, H., Shih, T.K. (eds.) IVIC 2009. LNCS, vol. 5857, pp. 664–674. Springer, Heidelberg (2009)
28. Herawan, T., Deris, M.M.: A soft set approach for association rules mining. Knowledge Based Systems 24(1), 186–195 (2011)
29. Herawan, T., Vitasari, P., Abdullah, Z.: Mining Interesting Association Rules of Student Suffering Mathematics Anxiety. In: Zain, J.M., Wan Mohd, W.M.B., El-Qawasmeh, E. (eds.) ICSECS 2011, Part II. CCIS, vol. 180, pp. 495–508. Springer, Heidelberg (2011)
30. Abdullah, Z., Herawan, T., Deris, M.M.: Efficient and Scalable Model for Mining Critical Least Association Rules. A Special Issue from AST/UCMA/ISA/ACN 2010, Journal of The Chinese Institute of Engineer 35(4), 547–554 (2012)

Improved Sammon Mapping Method for Visualization of Multidimensional Data

Halina Kwasnicka and Pawel Siemionko

Institute of Informatics, Wroclaw University of Technology, Wroclaw, Poland
{halina.kwasnicka,pawel.siemionko}@pwr.wroc.pl

Abstract. Three improvements to the Sammon mapping method are proposed. Two of them concern calculation complexity reduction. Introducing the limit for *delta* parameter allows to eliminate error fluctuations during data projection. Calculating distances not for all data points but for the part of them results in important reduction of the calculation time without worsening the final results. The third improvement allows adding new data to the projected ones without recalculation of all data from the beginning. The paper presents details of the proposed improvements and the performed experimental study.

Keywords: Data visualization, Dimension reduction.

1 Introduction

Data mining techniques able to analyze huge data sets become very important research area. Human visual system has remarkable capabilities of cognition, therefore data visualization techniques are very useful in many applications in different areas [10, 11]. Thanks to them one can easily understand the structure of complex data sets [1]. A number of data visualization methods, that preserves information contained in the data, have been developed. A simple visualization method is *scatterplot*. It uses Cartesian coordinates to present two dimensions of the data set. If the data set is very large the graph becomes too dense. Becker [3] suggests the possibility of binding the individual vectors in the clusters as a solution to that problem. Chambers [4] proposed to draw *scatterplot matrix* filled with graphs that display all possible binary combinations of attributes. It allows discovering correlation and dependency between features, but for high-dimensional data the user is overwhelmed by the amount of charts that must be evaluated and compared with each other. Inselberg [5] proposed *parallel coordinate plots*, it can be used in a simple way to find correlations between different attributes of the displayed set of vectors. Unfortunately for high dimensional data sets vertical axes are dense.

Mentioned visualization methods can be useful for relatively low-dimensional data, but in real-life problems we deal with collections which have even hundreds of dimensions, their direct visualization is impossible. It is necessary to reduce the number of features in such a way as to preserve as much as possible the

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 39–48, 2012.

information contained in the data set. One of a good and popular method is *Sammon mapping*. Unfortunately it has serious limitations mentioned later.

The main goal of the paper is to present authors' improvements that eliminate some of mentioned weaknesses of the Sammon mapping method. The paper is structured as follows. The next section presents shortly Dimension Reduction Techniques with focusing on Sammon mapping. Third section presents authors' improvements. The experimental study of the proposed improvements is described in section 4. The last section summarizes the paper.

2 Dimension Reduction Techniques

Visualized data sets can be easily interpreted by human beings, but human perception has some limitations [6–8]. According to [9], people have limited ability to accurately characterize the values presented to them. In the short-term memory we can remember and compare with each other a maximum of seven heterogeneous characteristics. Therefore dimensions number of high dimensional data must be reduced for example by data projection.

Problem Definition: Having given a set of observations X ($[X]_{p\times n} - p$ features and n observations), we search the representation Y ($[Y]_{k\times n} - k$ features, n observations) of the vector X, with a smaller number of dimensions ($k < p$). In other words, we are looking for such a transformation $P : X \to Y$, where P is an element of a set of possible transformations Q, for which the value of criterion **J** is maximal.

Linear methods of data dimensionality reduction is one of the possible way of dimension reduction: $Y = WX$, i.e., each attribute of a new vector Y (transformed observation) is calculated as a linear combination of corresponding vector (observation) of X, where W is a linear transformation matrix. The task is to find the best transformation matrix W which maximizes a given criterion. Use of linear projection methods is relatively simple, however they are not suitable for complex data structures, e.g., if the multidimensional vectors lie on the crooked hyperplane.

Non-linear methods of data dimensionality reduction are able to maintain relationships in complex data structures. Such methods typically rely on optimizing the cost function. Iterative methods are usually used for the optimization. The problem arises when a new data vector (observation) is added, in such a case the whole optimization procedure has to be repeated.

The popular approach lies on considering distances between points (observations) in the original space. Basically, objects similar to each other on the original space are close together on the target space. To evaluate quality of projection one must define a measure (called stress function) that represents the mapping error. The optimization problem is very difficult, the error function is nonlinear and complex, the method requires intensive calculation.

Multidimensional Scaling (MDS) is a family of methods based on changing distribution of vectors in a target space. The goal is to obtain such configuration which

well approximates dependencies between the vectors in the original space [13]. One of the most popular multidimensional scaling is the *Sammon mapping.*

Sammon mapping [12] tries to preserve the internal structure of the input p-dimensional data when are transformed to k-dimensional space ($k < p$) by preserving the distances between pairs of patterns. Usually Sammon mapping is used for the projection of multidimensional data to a 1, 2 or 3 dimensional space.

The course of the method is relatively simple, it consists of four main steps:

1. Initial values of matrix Y are calculated. Elements of Y can be random values from the assumed range, very often these values belong to the interval $[-1, 1]$.
2. Distances between all pairs of input vectors (observations, matrix X), and all pairs of output vectors (matrix Y) are calculated.
   Distances between input data are calculated only once (they are stored). Both distances, in the original d^*_{ij} and target d_{ij} spaces, are calculated according to assumed Euclidean distance measure.
3. A value of the error function (stress function) E is calculated according to the formula:
$$E = \frac{1}{\sum_{i<j}^{n} d^*_{ij}} \sum_{i<j}^{n} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \tag{1}$$
4. STOP condition is checked. If YES then stop the method and return the Y, if NO then go to the next step.
   Usually two conditions are defined: a predefined value of E_{min} or assumed reduction of that error *deltaE.*
5. New values of all elements of the output matrix Y are calculated.
   An optimization procedure is used for minimizing the error function by changing values of elements of Y matrix. Sammon applied the simplified Newton method [12]:
$$y_{ij}(t+1) = y_{ij}(t) - \eta \; delta_{ij}(t) \tag{2}$$
   where: y_{ij} is *j-th* attribute of *i-th* parameter of the vector Y; t is the number of iterations; η – parameter named by Sammon *Magic Factor* – it defines the power of modification caused by *delta*, best value of η was determined experimentally, good results gave 0.3 - 0.4. $delta_{ij}$ is the ratio of the first derivative of the stress function E with respect to y_{ij} and absolute value of the second derivative of E (with respect to y_{ij}).
6. Go to step 2.

A major disadvantage of the Sammon algorithm is considerable computational complexity. In each iteration $n(n-1)/2$ distances and derivatives must be calculated for the error function. Another drawback of the method is the large dependence on initial values of Y matrix. Additionally, what is very important from the practical point of view, Sammon method has no ability of generalization, it means that to add new data we must run the method from the beginning.

3 Improvements of the Sammon Method

The main weaknesses of the Sammon method are: (1) weak efficiency of the method, (2) adding new vectors to the data projected earlier. We propose remedies for both problems.

Increasing the Efficiency of the Sammon Method
We proposed two ways of efficiency increasing. The idea of the first modification arose from observation of the plot of error changes in subsequent iterations. In some iterations we observed a big growth of the error, in the example shown in Fig. 1 (on the left) it is at the beginning of the process and in 25th iteration. After a detailed analysis of the course of the projection, it turns out that the reason for this are the rapid changes of *delta*, which destroy the order of vectors in the target space, and influence the computational time. The remedy of this seems to be simple: just *limit the value of delta* to the assumed interval $[-\beta, \beta]$. If the calculated *delta* is outside of this interval, we set its value depending on the border which has been exceeded.

The second improvement lies on *reducing computational complexity of single iteration*. In each iteration $n(n-1)/2$ distances must be calculated (equation 1). A time-consuming operation in each step of the algorithm is computation of *delta*. Modification of the attributes of each vector is carried out, taking into account the error of distances to all others. However, especially in the early stages of the projection, where the error is large, and scattering of data is considerable, we can focus more on speed. Therefore, we can reduce the calculation of *delta* value by considering a subset of the output points. How many vectors we should consider we can fix in advance. Having n vectors of data, during *delta* calculation we can take into account every *k-th* vector ($k = 2$ or $k = 5$).

Adding New Vectors in the Sammon Method
Lack of ability to generalize is a serious drawback of the Sammon method. Assume that we have made a projection of a set of 10-dimensional data consisting of 10000 observations (vectors) into 3D space. Further, let us possess new 10000 data records and we would like to check whether there are correlations between them and predecessors. What can we do? One option is to project the whole data set consisting of 20000 vectors. This process is not the best solution due to the computational complexity. When the output file contains 10000 elements located adequately in 3D space, adding new, completely random 10000 objects disturbs the balance of the entire system. Value of Sammon error significantly increases. This will entail increase of *delta*, which in subsequent iterations will modify the attributes of all the vectors in the target space, not just the newly added. In the effect, the initial position of the prior elements will be lost. This will result in creation of a chaotic set of items, like when the algorithm is run from the beginning. Fig. 1 (on the right) presents above problem, where as an input data set we used a set of 1374 of 49-dimensional vectors. After 75 iterations the total Sammon error fallen below the value of 0.006, then we added the new 1576 vectors. The error increased suddenly, 16000 times. Stabilization took 50 iterations, which is close to the required number of iterations when method is

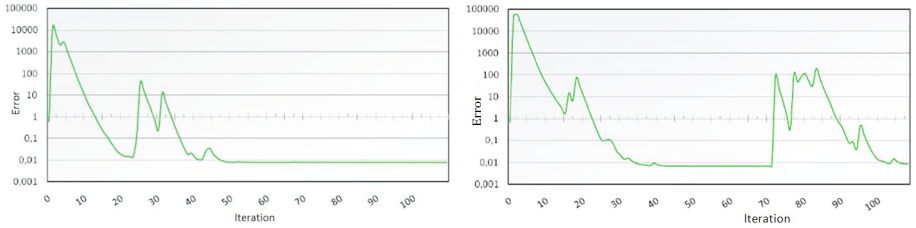


Fig. 1. Iterative process of projection of 49-dimensional data to 3D space (on the left) and the effect of adding new vectors after 75 iterations of the Sammon mapping (on the right)

run for the whole increased data set. To increase the efficiency of this process one should prevent premature modification of the existing vectors in the target data set. To increase the efficiency of this process one should prevent premature modification of the existing vectors in the target data set. To do this one needs to ensure proper positioning of the new elements in relation to those old. As with the original method, for the projection of new vectors we use Newton gradient optimization method. The process of adding new data is as follows:

1. At the beginning we add a new set of input vectors, and then we randomly select the output vectors corresponding to them, so both, the input and output sets, contain the same, increased number of vectors.
2. Next we calculate the projection error – function 1. If its value meets the STOP conditions, the algorithm is stopped. Otherwise, go to the next step.
3. We start the minimization of the error mapping **of the new points** using Newton gradient optimization method. For this purpose, we calculate new positions in the target space of newly added vectors according to the same formula as in the Sammon method (eq. 2). The old part of the output set remains unchanged and is used only as a reference during the counting error and its derivatives. In the evaluation of the first and second derivatives n is equal to the number of elements in the whole (enlarged) data set. After modifying all the new output vectors go to step 2.

With this relatively simple modification, we can add any number of new input vectors with the acceptable computational effort.

4 Experimental Studies

To conduct experiments, the dedicated environment, called 3DWIZ – *Multidimensional Data Visualizer* was implemented. It is a complete tool for analysis, reducing dimensions and visualization of data. In terms of logic, the program is divided into three main parts: the loading and preprocessing module, reducing dimensions module and visualization module. Two methods are implemented within 3DWIZ environment: Sammon mappings and authors' GENRED method based on a simple Genetic Algorithm. The main goal of the study was to verify if the proposed improvements make the idea of Sammon mapping better in quality

of results and its efficiency. We have performed a number of experiments with different data sets, inter alia real medical data. Method was fully tested on two real-life data sets.

Used Data Sets. For detailed studies we selected two data sets from the UCI ML Repository [14]. Additionally we applied developed method to real-life medical data. We decided to use a data that contain at least 20 attributes and at least 2000 vectors. *Mushroom* and *Landsat Satellite* sets were selected from the UCI ML repository. As the real-life data sets we experimented with two sets gathered from the Wroclaw Medical University, considered earlier in the rule association generating task. There is no place here to present these experiments. *Mushroom* data set consists of 8123 23-dimensional vectors. It classifies edibility of 23 mushroom species lamellate (Agaricus and Lepiota). The class label – edible or poisonous (2 classes), was not taken into account in the reduction of dimensions, it was used to determine the quality of clustering. We chose 3000 vectors for a set of basic vectors, and another 400 for testing our procedure of adding new data to the projected set. Data set *Landsat Satellite* consists of 4435 36-dimensional vectors. The data have been gathered using Landsat Multi-Spectral Scanner, whose task is to create pictures of the Earth surface in different spectral bands of light. Single vector consists of four images of the same piece of land, where first two images are made with green and red band of the visible spectrum, the last two in two near-infrared bands. Each image is a bitmap of the area size of 3×3, value of each pixel lies in the interval $[0, 255]$. The last attribute is class label, there are 6 classes.

Research Procedure for the Systematic Studies – Uncertainty of Measurement. Data projection using Sammon mapping methods strongly depend on the initial values of the vectors (points) in the target space. To have reliable results, it is important to carry out multiple measurements. After a number of initial trials, we observed that it is sufficient to perform 10 repetitions of each run. To validate obtained results we apply the statistical analysis of results and uncertainty of direct measurements. Let us assume that we measure n times the value of X and we receive a series of values $x_1, x_2, ..., x_n$. We consider X as a random variable, and $x_1, x_2, ..., x_n$ as a finite sample from an infinite set of all possible results. The best estimator of the value is the arithmetic average $\overline{x}$. The measure of uncertainty of a single measurement of the sample $x_1, x_2, ..., x_n$ is the standard deviation S_x.

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \ , \qquad S_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}, \qquad S_{\overline{x}} = \frac{S_x}{\sqrt{\overline{x}}} \tag{3}$$

Uncertainty of the single measurement of x_i is S_x which can be written as $x_i \mp S_i$. $S_{\overline{x}}$ is standard uncertainty. We can assume that the best estimation of value of X is x $\overline{x} \mp S_{\overline{x}}$.

Reducing the Value of *delta*. Level of restrictions on *delta* is denoted as β, it is changed in the experiment, and the time at which the algorithm reaches a predetermined value of the error is measured. Because a large amount of experiments

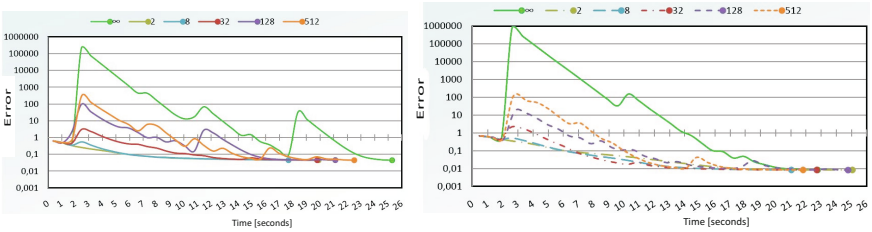


Fig. 2. The dependence of error on the duration of Sammon method for selected values of *delta* limits β, *Mushroom* data (left), *Landsat Satellite* data (right)

and the need to repeat each measurement ten times, it was decided to fix the error threshold, for *Mushroom* and **Landsat Satellite** equal to 0.045 and 0.0085 respectively. The data were projected on 3D space. The scope of the attributes in the input set was set as $[-1, 1]$. After the preliminary tests, it was decided to establish the tested values of *delta* as the successive powers of two.

According to the expectations, reducing the value of *delta* has improved the efficiency of the algorithm. We observed even 41% reduction in time required to attain the target, when $\beta = 8$. Fig. 2 shows the distribution of the time dependence of the algorithm to action on the value of β. For each of the tested values of β, the results were better than that obtained without restriction. When *delta* is reduced too much, in this case to less than 8, the algorithm must make a greater number of steps to correct the attributes of vectors and, therefore, time is longer. On the other hand, with increasing β, increases the vulnerability to fluctuations, which also extends the time.

Without limitation of *delta*, there are large fluctuations in the value of error of up to 100000 in the early stages of the process. For $\beta = 2$, this effect was virtually absent. With the increase in β, plot of the error becomes similar to the Sammon method without improvements. Subsequently, studies were performed using a set of *Landsat Satellite*, and the results are summarized in Fig. 2 (right).

In this case, the results are very similar. Again, the best result was obtained for $\beta = 8$. For values greater than 8 but less than 256 we observe a linear, positive correlation between the error and β. Only in the case of 256 and 512 there is an unexpected decrease. An interesting result was obtained for $\beta = 1$, the value of the error in this case is the largest. The reason is that it is too strong limitation of modifications of the vectors in the target space. Limitation of *delta* positively influences the time of reaching assumed mapping error.

Reducing Computational Complexity of a Single Iteration

We measured the time in which the algorithm reaches a predetermined value of the error depending on the parameter λ, where λ is the percentage of the set of input vectors used in the calculation of derivatives required in the error function calculation. The ceiling of error was determined individually for each data set as before to 0.045 and 0.0085, *Magic Factor*=0.4, data were projected into 3D space. We studied the behavior of the algorithm for ten values of the parameter λ, from 10% to 100% with the step of 10%. We used *delta* limited to 8. As we

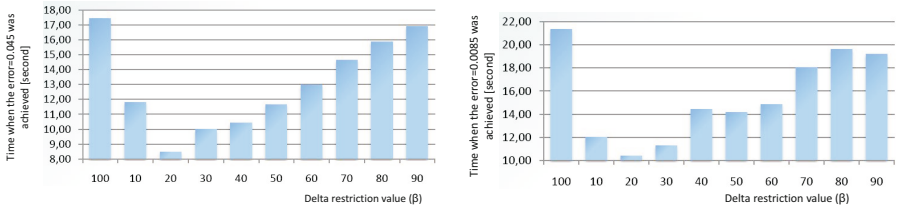


Fig. 3. Time required to reach an error=0.045 for Mushroom (left) and an error=0.0085 for Landsat Satellite data set (right), when different part (λ) of the vectors are used

expected, a significant improvement in performance of the method was observed. Use of 20% of the reference vectors improved the time of reaching assumed error by 105%, which is a very good result.

Fig. 3 (left) shows almost linear relationship between λ and the time needed to complete the task. Only when λ=10% the result differs from the scheme. In the case of a small number of reference vectors, the accuracy of the records correction in the target space is slightly reduced. We observed a much more precipitous decline in the value of error for smaller values of the parameter λ. The same test was conducted for *Landsat Satellite* data set. The results are presented in Fig. 3 (right). A significant increase in performance is seen also for this data set. As before, the extreme acceleration was achieved for $\lambda = 20\%$, and it was also 105%. For $\lambda = 10\%$ the same effect is seen as previous, but in this case, it affects to a lesser extent. Other results are very similar to those for Mushroom data set.

Effectiveness of the Mechanism of Adding New Vectors

The goal of this study was to verify the effectiveness of improvements enabling the adding of new vectors in the Sammon projection method. We used two reference data sets: *Mushroom* and *Landsat Satellite* with the size of 3000 vectors and two additional sets of size 400 records each, which were added to the projection process. At the beginning 3000 vectors of data were projected into 3D space, until the error reached values 0.045 and 0.0085 for *Mushroom* and *Landsat Satellite* respectively. Next we added 400 new vectors, and ran the method until it reached the error of 0.04 for *Mushroom* and 0.008 for *Landsat Satellite*. In this study we include additional stop condition, when the error change in subsequent iterations is less than 0.00001. The results were compared with these with 3400 vectors projected from the beginning.

The results for both data sets are collected in Table 1 and presented in Fig. 4. It contains the results of measurements of time of reaching the reference projection error of 0.04 when all 3400 vectors are projected, and when in initial projection is made with 3000 vectors and later 400 vectors are added, for *Mushroom* and *Landsat Satellite* data sets.

For *Mushroom* we obtained very interesting results. In the process of adding new vectors we obtained 30% decrease in time needed to achieve the target error value of 0.04 and a high reproducibility of the results. 400 additional vectors were added at seventeenth second of the projection process. This triggered an

Table 1. The results of measurements of time taken to reach the reference projection error of 0.04

Exp.	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10	Aver.	Error
Mushroom												
All	44.08	38.46	40.33	51.86	35.91	49.36	43.76	53.64	44.67	39.46	44.15	1.87
Adding	33.56	33.58	34.07	35.23	33.79	33.81	34.08	33.74	33.28	33.50	33.86	0.17
Landsat Satellite												
All	44.08	38.46	40.33	51.86	35.91	49.36	43.76	53.64	44.67	39.46	44.15	1.87
Adding	33.56	33.58	34.07	35.23	33.79	33.81	34.08	33.74	33.28	33.50	33.86	0.17

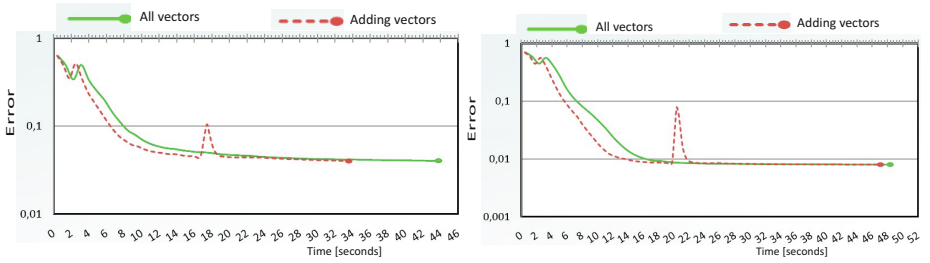


Fig. 4. The error as a function of time when all the 3400 vectors are projected and new 400 vectors are added to 3000 projected vectors, *Mushroom* (left) and *Landsat Satellite* (right)

immediate increase in the error but only to 0.1, which can be seen in Fig. 4. Positioning algorithm of new vectors almost immediately reduced the error. The whole process of increasing the number of vectors ended after 17 seconds. Without the described improvements, mapping the entire set of 3400 vectors from the beginning it would take an average of 44 seconds. So we saved 27 seconds.

For *Landsat Satellite* there was a small overall increase in speed. Adding new vectors occurred in twentieth second of the process. Again, there was a sudden increase in the value of the error to 0.1, which was quickly canceled out. Error value of 0.004 was achieved in 48 seconds. Adding new vectors took in this case 28 seconds. Carrying out the projection from the beginning of 3400 vectors spans more than 50 seconds. By streamlining the method we saved 22 seconds.

5 Summary

Sammon mapping method is one of the popular methods of dimension reduction, but it has some important weaknesses. The computational complexity is high, and, what is more important, when we obtain new data, we must start projection from the beginning. In the paper we propose three authors' improvements of this method. The first is calculation reduction based on elimination of fluctuation in the course of the method. Instability of the projection process is present especially in the initial stage of the process. As a remedy for that we proposed to introduce a limit for *delta* value. Experiments on two data sets proved the proposed solution. Study showed a significant reduction in fluctuation of error

values in the initial stages of the method, which resulted in the efficiency increase. The second improvement concerns also calculation reduction, but we reduced computational time of a single iteration of the method. Instead of calculation of $n(n-1)/2$ distances for all vectors (n is a number of vectors in the input data set), we can calculate distances for every k-th vector. We save a lot of time without worsening the results. Experiments confirm such results. We received up to 105% increase in speed.

The proposed mechanism for adding new vectors seems to be very important. Tests showed that we can insert new data in considerably shorter time than the original Sammon method. Developed application 3DWIZ that was used in this study allows for the tracking process of dimension reducing. It allows for any moves in three dimensional space in which the vectors are displayed. Thanks to this, discovery of relations and characteristics of the analyzed data sets are easier.

References

1. Card, S., Mackinlay, J., Shneiderman, B.: Readings in Information Visualization - Using Vision to Think. Morgan Kaufmann (1999)
2. Keller, P.R., Keller, M.M.: Visual Cues. IEEE Press, Los Alamitos (1993)
3. Becker, B.G.: Volume rendering for relational data. In: IEEE Symposium on Information Visualization (InfoVis 1997), pp. 87–91 (1997)
4. Chambers, J., Cleveland, W., Kleiner, B., Tukey, P.: Graphical Methods for Data Analysis. Wadsworth (1983)
5. Inselberg, A.: The Plane with Parallel Coordinates. Special Issue on Computational Geometry: The Visual Computer 1, 69–91 (1985)
6. Alley, T.R.: Physionomy and Social Perception. In: Social and Applied Aspects of Perceiving Faces, pp. 167–185 (1988)
7. Gibson, J.J.: The perception of the visual world. Houghton Mifflin Co., Boston (1950)
8. Pickett, R.M.: Response latency in a pattern perception situation. Acta Psychologica (27), 160–169 (1967)
9. Miller, G.: The magic number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63, 276–291 (1956)
10. Agrafiotis, D.K., Rassokhin, D.N., Lobanov, V.S.: Multidimensional scaling and visualization of large molecular similarity tables. Journal of Computational Chemistry 5(22), 488–500 (2001)
11. Lahdesmaki, H., Yli-Harja, O., Shmulevich, I., Zhang, W.: Distinguishing key biological pathways by knowledge based multidimensional scaling analysis: application to discriminate between primary breast cancers and their lymph node metastases. In: Yli-Harja, O., Shmulevich, I., Aho, T. (eds.) Proc. of the TICSP Workshop in Computational Systems Biology, WCSB 2003, Finland, vol. (21) (2003)
12. Sammon, J.W.J.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers, 401–409 (1969)
13. Karbauskaite, R., Dzemuda, G.: Multidimensional data projection algorithms saving calculations of distances. 123X Information Technology and Control (35) (2006)
14. Newman, A.A.: UCI Machine Learning Repository. University of California (2007), http://archive.ics.uci.edu/ml/index.html

Ontology Relation Alignment Based on Attribute Semantics

Marcin Mirosław Pietranik and Ngoc-Thanh Nguyen

Institute of Informatics, Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370, Wroclaw, Poland
{marcin.pietranik,ngoc-thanh.nguyen}@pwr.wroc.pl

Abstract. The problem of ontology alignment is based on finding mappings between instances, concepts and relations of two ontologies which (following Gruber's work [8]) can be defined as explicit specification of decomposition of some part of reality. This specification spreads over three levels of detail: the concept attribute level, the concept level and the relation level. This paper concentrates on identifying matches between relations of concepts which describe how these entities interact with each other. After careful analysis we have noticed that this level can be a source of many inconsistencies when two ontologies are blindly integrated. We take our work on attribute-based concept alignment and the consensus theory as a starting point. We extend it to handle the issues that appear when aligning relations. We give formal definitions along with careful formalization of set of requirements that eventual mapping algorithm should satisfy in order to reliably designate matches between ontologies on relation level.

1 Introduction

Providing reliable matches between ontologies is frequently a preliminary step to any task concerning knowledge management. In general, the problem can be described as the process of migrating contents of two ontologies. To illustrate its importance let's assume that there are two computer systems incorporating independent knowledge bases KB_1 and KB_2 with ontologies O (also referenced as source ontology) and O' (target ontology) acting as their schemas. The common situation occurs when the user makes a query to KB_2 utilizing its format (imposed by O'). There may be situation in which the data, that is the answer for such request, is present not in KB_2, but in KB_1. Therefore, the system must find proper content of KB_1 and provide it to the user. Due to the fact that both computer infrastructures utilize ontologies as formal foundations of their knowledge bases, considered issue comes down to finding those parts of O that most accurately match to specific parts of O' selected by user's query.

It has been stated in [10] that methods of integrating ontologies need to resolve conflicts that possibly occur between ontologies. These inconsistencies can be divided into three separated levels: *instance level*, *concept level* and *relation level*.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 49–58, 2012.

In the context of aligning ontologies, for each level there must be independent method of designating mappings of elements that are specific for considered layer.

In our previous work ([12]) we have formulated algorithms that designate mappings between concepts and their attributes (therefore working at the attribute and concept levels). In this paper we provide a novel framework for designating alignments between relations that occur between concepts. We utilize and extend aforementioned concept aligning algorithm, by assuming that designating mappings between relations can be properly formulated only for pairs of concepts that are already well aligned.

The problem can be stated as follows: *Having selected two concepts* c'_1 *and* c'_2 *from target ontology such that there exist two concepts* c_1 *and* c_2 *from source ontology that are well aligned, one should determine alignment between relations between concepts* c_1 *and* c_2*, and relations between* c'_1 *and* c'_2.

This problem can be decomposed into several subtasks. At first, the algorithm needs to identify which relations from analyzed ontologies are a source of potential conflicts. Then the method should determine relationships between relations, designate best matches for relations from target ontology among relations from source ontology and consequently select relations that are not present within target ontology. Consider the example from Figure 1.

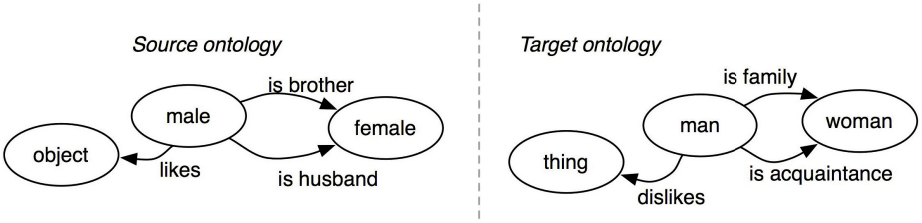


Fig. 1. Example relations between concepts

Obviously concepts presented in Figure 1 are pairwise well aligned (*female-woman*, *male-man*, *object-thing*). Ideally, the algorithm should determine that relations *is brother* and *is husband* are less general that relation *is family* and therefore they complement the description of possible interactions between selected concepts *man* and *woman* within target ontology. Moreover, the solution should clearly state that relations *likes* and *dislikes* are a potential source of conflicts and that the relation *is acquaintance* does not interfere with any relation nor is a source of any inconsistency.

The main contribution of our work refers to the analysis of formal semantics of relations that connect concepts within particular ontologies. By analogy to semantics of attributes, we assign logic sentences to every relation and therefore we are able to identify relationships between relations. In other words, we can unequivocally state which relations are equivalent, which can be ordered in hierarchy and which are contradictory (so are the source of possible inconsistency

on the relation level). Moreover, we are able to clearly calculate the distance between these annotations (utilizing algorithm from [15]) and despite lack of relationship between them, to reliably answer how close two relations are.

The paper is organized as follows. In Section 2 we provide an overview of related works that has been done in the field of ontology alignment. In Section 3 we give the description of basic notions used throughout the whole paper. In Section 4 we present our approach to designating mappings between concepts' relations. Brief overview of upcoming works and a summary are given in part 5.

2 Related Works

Formerly developed approaches to ontology alignment provide generic methodology for incorporating knowledge about relations between concepts into the process of designating mappings. In general, they are based on a analyzing taxonomic structure of particular ontology, using a similarity measure of Taxonomic Precision ([3]). The difference with our approach is the fact that previously proposed techniques incorporated the knowledge about relation in order to improve aligning concepts but not relations itself. These methods (classified in [5] as structure-based approaches) are frequently adapted to variety of practical applications, such as finding alignments between biomedical ontologies ([1]). Despite good results acquired by available mapping systems ([16]), the problem of aligning ontologies has been narrowed to designating mappings between ontologies stored using one certain representation format, which is OWL standard. Regardless of variety processing tools and broad acceptance, this tool has many restrictions that have been addressed in details in [9]). Recent progress in this approach has been addressed and broadly described in [6]. This work also contains overview of the main issues that still need to be solved, but none of the papers cited there do not relate directly to aligning relations between concepts, neither to incorporating semantics of attributes.

The problem of processing relation is more frequently related to the topic of ontology integration. This issue has been addressed in [11] where reliable methods for representing ontological conflicts were described in details. This publication also contains broad characterization of algorithms for conflict resolution based on consensus theory ([2]), that has been proved useful in terms of any kind of task concerning knowledge integration.

Because of the limited space for this paper, we are not able to present extensive overview of works that have been done in the field of ontology alignment. For more detailed descriptions please refer to our former publications, [12], [14] and [15].

3 Basic Notions

We take [10], [15] and [14] as a starting point. Therefore we define ontology as a triple $O = (C, R, I)$, where C is the set of concepts, R is a set of relations between them (the definition of it's members will be given further) and I is a

set of instances. The concept c from set C is defined as a tuple $c = (Id^c, A^c, V^c)$ in which Id^c is a concept's label (an informal name of a concept), A^c is a set of its attributes and V^c is a set of domains of attributes from A^c. The triple c can be also called *concept's structure*. Denoting some finite set of attributes as A and set of their valuations as V, in further parts of this article by *Real World* we will call a pair *(A,V)* and every ontology O such that $\forall_{c \in C} A^c \subseteq A$ will be called *(A,V)-based*.

Let S_A be the set of atomic descriptions of attributes given in natural language. By L_s^A we will call the formal language, using elements of S_A and logic operators $\neg, \vee, \wedge$. Therefore, L_s^A is a sublanguage of the sentence calculus language. We will use it to assign explicit semantics to attributes. Hence, by semantics of attributes we will call a partial function $S_{A,C} : A \times C \rightarrow L_s^A$. Logic sentences from L_s^A annotate every inclusion of attribute within concept. Such approach allows us to express the variety of characteristics that some attribute may show when included within a number of differing concepts. For example, the attribute *Address* indicate different meaning when incorporated in the concept *Webpage* and concept *Person*. Developing this idea we were able to formulate different relationships that describe how attributes relate with each other. Assuming the existence of two *(A,V)-based* ontologies O and *O'* with respective sets of concepts C and *C'* such that $c \in C, c' \in C'$, $a, b \in A$ we define relationships between attributes as follows:

- *equivalency* (denoted as $\equiv$) if the formula $S_{A,C}(a, c) \Leftrightarrow S_{A,C}(b, c')$ is a tautology (in classical logic interpretation) then attributes a and b are equivalent
- *generalization* (denoted as $\uparrow$) if the formula $S_{A,C}(a, c) \Rightarrow S_{A,C}(b, c')$ is a tautology then attribute b is more general than attribute a
- *contradiction* (denoted as $\downarrow$) if the formula $\neg(S_{A,C}(a, c) \wedge S_{A,C}(b, c'))$ is a tautology then attributes a and b are contradicting

For broad description of these relationships, please refer to our previous publications [15] and [12].

The foundation of our approach to aligning concepts are three functions $M_A^{c,c'}, M_A^c$ and M_C that are used to unequivocally designate the degree of alignment of two attributes, the degree of alignment for selected attribute from source ontology and the degree of alignment of two concepts. The postulates that function $M_A^{c,c'} \rightarrow [0, 1]$ must satisfy are as follows: *(i) The function* $M_A^{c,c'}$ *must not be symmetrical. (ii) If two attributes* a *and* b *are equivalent then* $M_A^{c,c'}(a, b) = M_A^{c,c'}(b, a) = 1$. *(iii) If* $a \uparrow b$ *and not* $a \equiv b$ *then* $M_A^{c,c'}(a, b) = 1$ *and* $M_A^{c,c'}(b, a) < 1$. This function incorporates the distance between two semantics (d_S). Due to the limited space we cannot present its full definition so for details, please refer to [15]. Eventually, $M_A^{c,c'}$ can be defined below:

$$M_A^{c,c'}(a, b) = \begin{cases} 1 & \text{if } a \equiv b \\ 1 & \text{if } a \uparrow b \text{ and not } a \equiv b \\ 1 - d_S(S_A(a, c), S_A(b, c')) & \text{otherwise} \end{cases} \quad (1)$$

The next function $M_A^c : A^c \to [0,1]$ identifies "the best match" for particular attribute from some source concept within set of attributes of target concepts. It is defined as follows:

$$M_A^c(a) = \begin{cases} \frac{1}{|Z^*|} \sum_{(a,b)\in Z^*} M_A^{c,c'} & \text{if } |Z^*| > 0 \\ M_A^{c,c'} & \text{if } |Z^*| = 0, \text{ for } b = argmax_{b\in A^{c'}} M_A^{c,c'}(a,b) \wedge M_A^{c,c'}(a,b) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The set Z^* used in this function is defined as $Z^* = \{(a,b) : a \in A^c, b \in A^{c'}, b = argmax_{b\in A^{c'}} M_A^{c,c'}(a,b) \wedge M_A^{c,c'}(a,b) \geq T\}$.

In order to provide a method for calculating the degree to which a concept from source ontology can be aligned to a concept from target ontology we need to formulate the function $M_C \to [0,1]$. It satisfies following postulates: *(i) The function M_C must not be symmetrical. (ii) Assuming the existence of concepts* c *and* c' *such that $A^c = \{a\}$ and $A^{c'} = \{b\}$ where $a \uparrow b$ and not $a \equiv b$ then the condition $M_C(c,c') \geq M_C(c'c)$ must be met.* Having that and incorporating definitions 1 and 2 we have developed an algorithm that calculates its values. The input data are two sets of attributes (A^c and $A^{c'}$). In the first step it discards the unnecessary redundancy from the source concept's structure A^c (for example - it removes equivalent attributes), eventually creating the working set $\overline{A_c}$. Then the following formula is executed $M_C = \frac{\sum_{a\in\overline{A_c}} M_A^c(a)}{|\overline{A_c}|}$ and eventual result is returned. Because of the limited space we cannot provide the full listing - for details, please refer to [12].

Definition 1. *Assuming that there exists two ontologies* O *and* O' *with respective sets of concepts* C *and* C', *by well aligned concepts we call a pair* (c,c') $c \in C, c' \in C'$ *such that* $M_C(c,c') \geq T$, *where* T *is assumed threshold value* $T \in [0,1]$.

As stated in the previous part of this paper within ontology $O = (C,R,I)$ we distinguish set R of binary relations between concepts from set C. Therefore the set R is defined as $R = \{r_1, r_2, ..., r_n\}$ such that $n \in N$ and $r_i \subset C \times C$ for $i \in [1,n]$. By analogy to semantics of attributes we wanted to assign similar semantic descriptions to every relation from ontology. Thus we introduce the set S_R containing atomic descriptions of relations. Then by L_S^R we will denote the sublanguage of sentence calculi, built from elements of set S_R and basic logic operators $\neg, \vee, \wedge$. Consequently, by semantics of relations between we call a partial function $S_{R,O} : R \to L_s^R$. Assuming the existence of two *(A,V)-based* ontologies O and O' with respective sets of relations R and R' such that $r \in R, r' \in R'$, we define relationships between relations as follows:

Definition 2. *Two relations r and r' are equivalent referring to their semantics (semantic equivalency) if the formula $S_{R,O}(r) \Leftrightarrow S_{R,O}(r')$ is a tautology. We denote this fact using the symbol (denoted as $\equiv$).*

For example, the relation *"is spouse"* is equivalent with the relation *"is partner"*.

Definition 3. *The relation r' is more general than the relation r referring to their semantics (semantic generality) if the formula* $S_{R,O}(r) \Rightarrow S_{R,O}(r')$ *is a tautology. To denote this situation we use the symbol* $\uparrow$.

For example, the relation *"is family"* is more general than the relation *"is brother"*.

Definition 4. *Two relations r and r' are in contradiction referring to their semantics (semantic contradiction) if the formula* $\neg(S_{R,O}(r) \wedge S_{R,O}(r'))$ *is a tautology. To denote this fact we use the symbol* $\downarrow$.

For example, the relation *"likes"* is in contradiction to the relation *"dislikes"*.

4 Aligning Concepts' Relations

In order to reliably transform the knowledge about possible relations between concepts, we need to define the function that calculates the degree to which we can align particular relation from source ontology into relation appearing in the target ontology. Assuming the existence the source ontology *O=(C,R,I)* and the target ontology *O'=(C',R',I')* the requirements for such function with a signature $M_R : R' \times R \rightarrow [0, 1]$ can be formulated with following postulates:

1. The function M_R must not be symmetrical.
2. If two relations $r \in R$ and $r' \in R'$ are equivalent $(r \equiv r')$ then $M_R(r, r') = M_R(r', r) = 1$
3. If $r \uparrow r'$ and not $r \equiv r'$ then $M_R(r, r') = 1$ and $M_R(r', r) < 1$
4. If $r \downarrow r'$ then $M_R(r, r') = M_R(r', r) = 0$

The first postulate concerns the intuition behind aligning ontologies. There are examples when it is easy to find proper alignments from selected relation from source ontology to some relation from target ontology, but it is more complicated to designate alignments in the other direction. Consider the example from Section 1 in which it should be fairly simple to find mapping for relation *is brother*, but more difficult to find alignment for relation *is family*. The next two postulates refer to situation in which two relations are in some relationship. Having in mind the intuition, we can say that when two relations are equivalent we should be able to unequivocally transform them, thus the degree of alignment must be maximal. Similar issue occurs when some relation is more general than the other. The degree of alignment should also be maximal, due to the fact that we can reliably transform the knowledge expressed with more details into less complex, than in the other way. For example (referring to Figure 1), knowing that someone *is brother* of someone else implicates that they are also a family, but knowing that few people are related with each other does not implicate the type of their affinity. The last of the postulates concerns the situation in which

two relations are in contradiction (for example, a pair *likes-dislikes*). In such situation the degree of the alignment between them should be minimal. Moreover, this value should indicate that such two relations are the source of potential inconsistencies. The knowledge about it must simplify the process of satisfying the condition for noncontradiction when two ontologies are integrated ([10]).

Bearing in mind listed postulates we were able to define function M_R as follows:

$$M_R(r, r') = \begin{cases} 1 & \text{if } r \equiv r' \\ 1 & \text{if } r \uparrow r' \text{ and not } r \equiv r' \\ 0 & \text{if } r \downarrow r' \\ 1 - d_S(S_{R,O}(r), S_{R,O}(r')) & \text{otherwise} \end{cases} \tag{3}$$

Presented function is a straightforward realization of postulates presented in this section. In the last section it incorporates the function d_S that calculates the distance between two logic sentences described in details in [15]. Due to the limited space for this paper, we cannot give its comprehensive overview.

To ensure that our algorithm analysis only relations for selected pairs of concepts, we define the auxiliary set *Rel*. Note that this set contains only directed relations from c_1 to c_2, and not from c_2 to c_1. This fact guarantees that the eventual algorithm will analyze only relations that are compatible in terms of their directions.

Definition 5. *By* $Rel(c_1, c_2)$ *we denote the set of directed relations from concept* c_1 *to concept* c_2 *as* $Rel(c_1, c_2) = \{r \in R | (c_1, c_2) \in r\}$.

As said in Section 1 the eventual algorithm should work threefold. It is expected to designate matching relations and find those pairs of relations which are source of potential conflicts between ontologies. Moreover, it must select relations from source ontology that are not present within target ontology and do not entail conflicts. Bearing this in mind, we have formulated the output of prepared algorithm as three sets:

- R_{Al} containing tuples of a form $(r, r', M_R(r, r'))$ which represent matching relations and the degree to which we can align them
- $R_{new} = \{r \in R | M_R(r, r') = 0 \wedge \neg \exists r' \in R. r \downarrow r'\}$ which contains relations that represent such connections between concepts that do not occur within target ontology
- $R_{con} = \{(r, r') | r \in R, r' \in R', (c_1, c_2) \in r, (c'_1, c'_2) \in r', r \downarrow r'\}$ containing pairs of conflicting relations

We assume the existence of two different ontologies $O = (C, R, I)$ and $O' = (C', R', I')$ and that within these ontologies there are two pairs of well aligned concepts (c_1, c'_1) and (c_2, c'_2) such that $c_1, c_2 \in C$ and $c'_1, c'_2 \in C'$.

Algorithm 1
Input: Two pairs of well aligned concepts (c_1, c_2) and (c'_1, c'_2)
Output: Sets R_{Al}, R_{new}, R_{con}
Procedure:
BEGIN
1. $R_{Al} = \phi, R_{new} = \phi, R_{con} = \phi$;
2. Remove redundancy from set $Rel(c_1, c_2)$
2.1. if in $Rel(c_1, c_2)$ there are two relations r,r' such that $r \equiv r'$ then remove from set $Rel(c_1, c_2)$ relation r or r';
2.2. if in $Rel(c_1, c_2)$ there are two relations r,r' such that $r \uparrow r'$ and not $r \equiv r'$ then remove from set $Rel(c_1, c_2)$ relation r';
3. For each relation $r \in Rel(c_1, c_2)$
Begin
3.1 $\widetilde{R_{Al}} = \{(r, r', 1) | r' \in Rel(c'_1, c'_2) \wedge r \equiv r'\}$;
3.2 $\widetilde{R_{Al}} = \{(r, r', 1) | r' \in Rel(c'_1, c'_2) \wedge r \uparrow r'\}$;
3.3 $\widetilde{R_{con}} = \{(r, r') | r' \in Rel(c'_1, c'_2) \wedge r \downarrow r'\}$;
3.4 if $\widetilde{R_{Al}} = \phi$ then
$\widetilde{R_{Al}} = \{(r, r', M_R(r, r')) | r' \in Rel(c'_1, c'_2) \wedge M_R(r, r') \geq T \wedge r' = argmax_{r' \in (c'_1, c'_2)} M_R(r, r')\}$;
3.5 if $\widetilde{R_{Al}} = \phi$ then
$\widetilde{R_{new}} = \{r\}$;
3.6 $R_{Al} = R_{Al} \cup \widetilde{R_{Al}}$, $R_{con} = R_{con} \cup \widetilde{R_{con}}$, $R_{new} = R_{new} \cup \widetilde{R_{new}}$;
End
4. Return sets R_{Al}, R_{new}, R_{con};
END

The algorithm at first generates empty result sets for further processing. Next it removes unnecessary redundancies from the set of relations extracted from source ontology. It discards relations that are equivalent and these relations that remain in *generalization* relationship with any other relation from processed set (for example, we do not simultaneously need relations *"is colleague"* and *"is coworker"* or relation *"is family"* if we own relation *"is brother"*). The reason why we alter only the set $Rel(c_1, c_2)$ is because we cannot modify the set from target ontology due to the fact that the end user makes requests utilizing its format of expressing knowledge and conceivable modification could easily restrict expected response. The next step is to find "best matches" for every relation taken from source ontology. This part can be decomposed into several stages. The first is based on incorporating conditions for relationships between relations (from Definitions 2, 3 and 4). Throughout steps 3.1 and 3.2 the algorithm utilizes postulates for function M_R and adds to the resulting set R_Al pairs of relation for which the alignment degree is maximal (equal to 1). On this stage we designate alignments for equivalent relation such as *"is partner"/"is spouse"* and similarly for relations that remain in generalization relationship (for example, *"is brother"/"is family"*). Consequently, in step 3.3, we identify these relations which are in contradiction and which are source of possible conflicts

(such conflicts occur when two ontologies are merged and within resulting ontology two relations that are in contradiction relationship are both present, for example *"likes"/"dislikes"*). Next, the algorithm needs to process relations that do not participate in any relationship. It is achieved within consecutive steps 3.4 and 3.5, which designate best alignments for selected relations (in terms of alignment degree from equation 3) and if none mapping is found, the algorithm decides that processed relations neither have a match nor cause any conflict. In such situation particular relation is added to the set containing relations that are not expressed within target ontology. In the following step 3.6 the algorithm adds partial results obtained for processed relations from source ontology into the final resulting sets and in the last step 4 returns designated outcomes.

Assuming that cardinalities of sets $Rel(c_1, c_2)$ and $Rel(c'_1, c'_2)$ are respectively m and n we are permitted to say that the complexity of our algorithm is approximately quadratic $O(mn)$. Despite the fact that the algorithm needs to process every pair of relations from analyzed sets, observations prove that values m and n are frequently low. Therefore not many comparisons are required and we can say that our algorithm is effective.

Despite obvious advantages (such as identifying relationships between attributes), our approach has few downsides. The main issue is the assumption about having two pairs of well-aligned concepts. As stated in [4], finding pair of matched entities within two large scale ontologies can be very difficult and time/resource consuming. This is the issue that we plan to examine in upcoming work. As stated in Section 2, the most common implementation method is OWL language. The problem of this solution is the lack of any kind of explicit mechanism for assigning logic sentences to attributes (that are stored as plain *key-value* pairs). For this reason it is difficult to conduct any kind of experiment utilizing currently available benchmarks. Issues related to difficulties with experimental testing of our approach have been described in details in [13]. Nevertheless, our method proves to be useful. Initial tests conducted with prepared experimental environment shows that our approach gives promising results and can be easily integrated into any application that requires designating mappings between knowledge representations (e.g. integrating federated data warehouses).

5 Future Works and Summary

In this paper we have presented the novel framework for aligning relations between concepts from two heterogenous ontologies. We have given concise theoretical foundations, detailed description of the main approach and careful explanation of possible use-cases in which the necessity of aligning relations appear. In the future we want to concentrate on implementing our ideas within experimental environment that currently remain under active development process. Second direction of our work concerns reducing complexity of the process of aligning whole ontologies, not only selected, single concepts or relations. We treat our bottom-up approach to aligning ontologies as an interesting redefinition of this broadly discussed topic. We move the focus from aligning OWL files,

therefore, creating a unified framework that can be easily integrated within any situation that requires mapping semantic descriptions of knowledge.

Acknowledgment. This paper was partially supported by Grant no. N N519 444939 funded by Polish Ministry of Science and Higher Education (2010-2013) and by grant No. 2011/01/N/ST6/01438 from the National Science Centre, Poland.

References

1. Batet, M., Sanchez, D., Valls, A.: An ontology-based measure to compute semantic similarity in biomedicine. Journal of Biomedical Informatics 44(1), 118–125 (2011)
2. Day, W.H.E.: Consensus methods as tools for data analysis. In: Bock, H.H. (ed.) Classification and Related Methods for Data Analysis, pp. 312–324. North-Holland (1988)
3. Dellschaft, K., Staab, S.: On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 228–241. Springer, Heidelberg (2006)
4. Duong, T.H., Jo, G.S., Jung, J.J., Nguyen, N.T.: Complexity analysis of ontology integration methodologies: A comparative study. Journal of Universal Computer Science 15, 877–897 (2009)
5. Euzenat, J., Shvaiko, P.: Ontology Matching, 1st edn. Springer, Heidelberg (2007)
6. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology Alignment Evaluation Initiative: Six Years of Experience. In: Spaccapietra, S. (ed.) Journal on Data Semantics XV. LNCS, vol. 6720, pp. 158–192. Springer, Heidelberg (2011)
7. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Semantic Schema Matching. In: Meersman, R. (ed.) OTM 2005. LNCS, vol. 3760, pp. 347–365. Springer, Heidelberg (2005)
8. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
9. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Web Semantics: Science. Services and Agents on the World Wide Web 6, 309–322 (2008)
10. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management. Advanced Information and Knowledge Processing. Springer (2008)
11. Nguyen, N.T.: A Method for Ontology Conflict Resolution and Integration on Relation Level. Cybernetics and Systems 38(8), 781–797 (2007)
12. Pietranik, M., Nguyen, N.T.: A Method for Ontology Alignment Based Attribute Semantics. Cybernetics and Systems 43(4), 319–339 (2012)
13. Pietranik, M., Nguyen, N.T.: A Multi-attribute and Logic-Based Framework of Ontology Alignment. In: Zgrzywa, A., Choroś, K., Siemiński, A. (eds.) Multimedia and Internet Systems: Theory and Practice. AISC, vol. 183, pp. 99–108. Springer, Heidelberg (2013)
14. Pietranik, M., Nguyen, N.T.: Attribute Mapping as a Foundation of Ontology Alignment. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011, Part I. LNCS, vol. 6591, pp. 455–465. Springer, Heidelberg (2011)
15. Pietranik, M., Nguyen, N.T.: Semantic Distance Measure between Ontology Concept's Attributes. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part I. LNCS, vol. 6881, pp. 210–219. Springer, Heidelberg (2011)
16. Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I.: Proceedings of the 5th International Workshop on Ontology Matching (2010), http://ceur-ws.org/Vol-689/

Data Deduplication Using Dynamic Chunking Algorithm

Young Chan Moon[1,⋆], Ho Min Jung[1], Chuck Yoo[2], and Young Woong Ko[1]

[1] Dept. of Computer Engineering, Hallym University, Chuncheon, Korea
{ycmoon,chorogyi,yuko}@hallym.ac.kr
[2] Dept. of Computer Science and Engineering, Korea University, Seoul, Korea
hxy@os.korea.ac.kr

Abstract. Data deduplication is widely used in storage systems to prevent duplicated data blocks. In this paper, we suggest a dynamic chunking approach using fixed-length chunking and file similarity technique. The fixed-length chunking struggles with boundary shift problem and shows poor performance when handling duplicated data files. The key idea of this work is to utilize duplicated data information in the file similarity information. We can easily find several duplicated point by comparing hash key value and file offset within file similarity information. We consider these duplicated points as a hint for starting position of chunking. With this approach, we can significantly improve the performance of data deduplication system using fixed-length chunking. In experiment result, the proposed dynamic chunking results in significant performance improvement for deduplication processing capability and shows fast processing time comparable to that of fixed length chunking.

Keywords: Deduplication, Metadata, Chunking algorithm, File similarity.

1 Introduction

Data deduplication is commonly used as a data compression technique for eliminating redundant data. The key advantage of this technique is to improve storage utilization and can also be used low-bandwidth network to reduce the number of bytes. Generally, data deduplication technique identifies and eliminates duplicated data blocks with a cryptographic hash function. In data deduplication, the basic idea is to split a file into blocks (chunking) and applies hash functions to compute hash values. To check data duplication, the client sends the hash key list to the server. The hash key for each chunk is used to determine if that chunk exists in the multiple locations by comparing hash keys. If there are same hash keys on another location, we assume that the chunk is duplicated. Therefore,

⋆ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2011-0029848) and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2009-0076520)

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 59–68, 2012.

we can prevent duplicated data blocks to be transferred. Generally, the chunking algorithms are divided into two; fixed-length chunking and variable-length chunking. The fixed-length chunking approach achieves very fast data deduplication result but the performance is not good, because boundary shift problem degrades the deduplication performance. On the contrary, the variable length chunking achieves high degree of performance while causing high computation overhead and longer processing time.

We assume that a file contains duplicated data blocks with spatial locality. When we modify a file or append new data blocks to an old version file, the new version of a file usually contains lots of duplicated region of data blocks compared with previous version of a file. Therefore, if we find one duplicated data block then we can find lots of duplicated data blocks around this position. Sometimes there exists a special file that completely changes the contents of the file with a small change of data such as zip compress format and jpg image format. However, in almost all data format, data modification in a file limits the data change within a small area. In this paper, we try to minimize chunking time close to the processing time of fixed length chunking by applying dynamic chunking.

In this paper, we propose an efficient dynamic chunking mechanism based on fixed-length chunking and file similarity scheme. The key idea of this work is to utilize duplicated data information in the file similarity information. We can easily find several duplicated point by comparing hash key value and file offset within file similarity information. We consider these duplicated points as a hint for starting position of chunking. The proposed system can search lots of duplicated data blocks around these starting position with fixed length chunking overhead.

The rest of this paper is organized as follows. In Section 2, we describe related works about deduplication system. In Section 3, we explain the design principle of proposed system and implementation details for data deduplication using file similarity. In Section 4, we show performance evaluation result of exploiting file pattern and we conclude and discuss future research plan.

2 Related Work

In a backup system, a version control program, P2P system and CDN system, data deduplication scheme is widely used for minimizing disk capacity and reduce network traffic[1] [2] [3]. The state of art works related to data deduplication is Rsync[4], DEDE[5], Venti[6], LBFS[7] and Multi-mode[8]. Rsync is a software application for Unix systems which synchronizes files and directories from one location to another locations while minimizing network traffic using delta encoding. An important feature of Rsync is open source and it use single round approach. Rsync can copy or display directory contents and files, optionally using compression and recursion. Venti is a network storage system that permanently stores data blocks. A 160-bit SHA-1 hash of the data acts as the address of the

data. This enforces a write-once policy since no other data block can be found with the same address. The addresses of multiple writes of the same data are identical. So duplicate data is easily identified and the data block is stored only once. LBFS, a network file system designed for low bandwidth networks. LBFS exploits similarities between files or versions of the same file to save bandwidth. It avoids sending data over the network when the same data can already be found in the servers file system or the clients cache. Using this technique, LBFS achieves up to two orders of magnitude reduction in bandwidth utilization on common workloads, compared to traditional network file systems.

DEDE is a decentralized deduplication system designed for SAN clustered file systems that supports a virtualization environment via a shared storage substrate. Each host maintains a write-log that contains the hashes of the blocks it has written. Periodically, each host queries and updates a shared index for the hashes in its own write-log to identify and reclaim storage for duplicate blocks. Unlike inline deduplication systems, the deduplication process is done out-of-band so as to minimize its impact on file system performance. In [9], they propose a data deduplication system using multi-mode(source-based approach, inline approach and post processing approach). The multi-mode system can be operated in several modes that a user specifies during system operation, therefore, this system can be dynamically adjusted under consideration of system characteristics.

3 System Design

3.1 File Similarity Information

In this work, we utilize the file similarity information that has two tuples, hash key and file offset information. With that information, we can easily find duplicated region on a file by comparing hash key between two files. If there is same hash key, we use corresponding file offset where we apply fixed-length chunking, otherwise, we skip data deduplication. Therefore, the processing time of the proposed system is very short compared with variable-length chunking approach.

The key idea of this paper is applying file similarity information to find duplicated points between two files. In this work, we have to decide how much duplicated data blocks exist between two files. As a fast and efficient file comparison mechanism, we exploit the representative hash list that is used for evaluating the degree of similarity between two files. We made representative hash list for a given file by searching and composing the maximum hash list.

As can be seen in figure 1, Rabin hash function is used for computing a hash key for a block. The Rabin hash starts at each byte in the first byte of a file and over the block size of bytes to its right. If the Rabin computation at the first byte is completed then we have to compute the Rabin hash at the second byte incrementally from the first hash value(fingerprint). Now that the hash value at the second byte is available then we use it to incrementally compute the hash

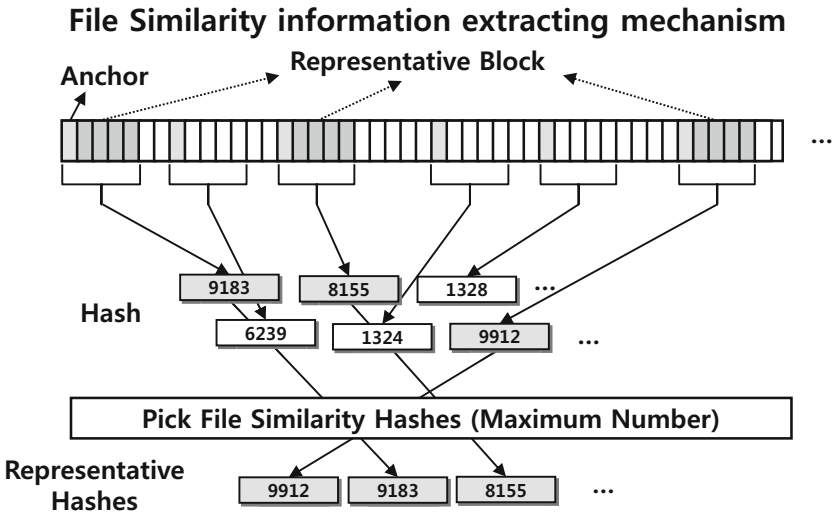


Fig. 1. File similarity information extracting mechanism

value at the third, and continue this process. We have to sort the Rabin hash value and choose only 10 maximum values as a representative hash.

In this work, we made the representative hash list for all files before data deduplication. We extract one representative hash for 1 MByte therefore the amount of additional information for file similarity is not critical for metadata management.

Algorithm 1. File similarity extracting algorithm

```
Input: FileStream
Output: Hasharray
begin
    offset ← 0;
    length ← Length(FileStream);
    while offset < length do
        offset ← seek(FileStream, seek-cur);
        byte ← readbyte(FileStream);
        if FindAnchor( Byte ) = true then
            hasharray[0].offset ← offset;
            hasharray[0].value ← RabinHash( readblock(FileStram, offset) );
            quicksort(hasharray);
        end
    end
    return hasharray;
end
```

Algorithm 1 explains how we can get file similarity information from file stream. First, the algorithm seeks the current file position using seek() function. By generating Rabin hash function with the byte of current position, we can get new hash key. The hash key is compared with previous hash keys. If the hash key is bigger than a hash key in hasharray, we replace the new hash key with minimum value in hasharray. This procedure repeats to the end of file and the output is the representative hash list for used file similarity information.

3.2 How to Find Duplicated Position Using Representative Hash?

Figure 2 shows how we can find duplicated points between files using the representative hash information. The target file is located on the server and the source file is on the client. The client creates the source file by modifying target file. In this example, we made the source file by inserting one new block in front of A, deletes B block and finally inserts one block and 246 byte data in front of C block. The block size is 8Kbyte. With the representative hash, we can get two duplicated points A and C. By comparing offset information, we can easily guess how each point is located on the file. With this information, we can start the fixed length chunking and find most of the duplicated data.

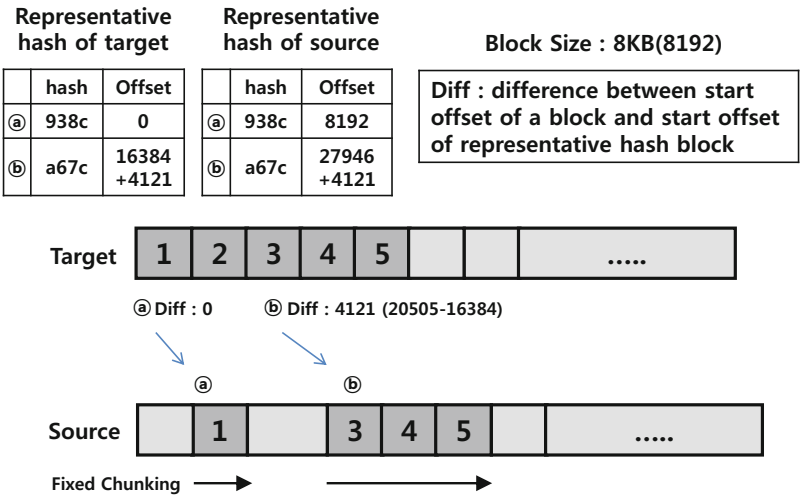


Fig. 2. Duplicated point search mechanism using representative hash

Figure 3 shows the conceptual diagram of dynamic chunking scheme. First, the client searches the representative hash for the file to be deduplicated. If the client cannot find the pre-computed representative hash then compute it using Rabin hash function. Second, the client sends the representative hash to the server. The server can find high similar file with the representative hash by comparing

the hash values between files on the server. If the system finds a similar file then it computes duplicated points by computing diff value. We can get diff value by difference between start offset of a block and start offset of representative hash block. In the client, this diff value is used to locate duplicated point and dynamic chunking.

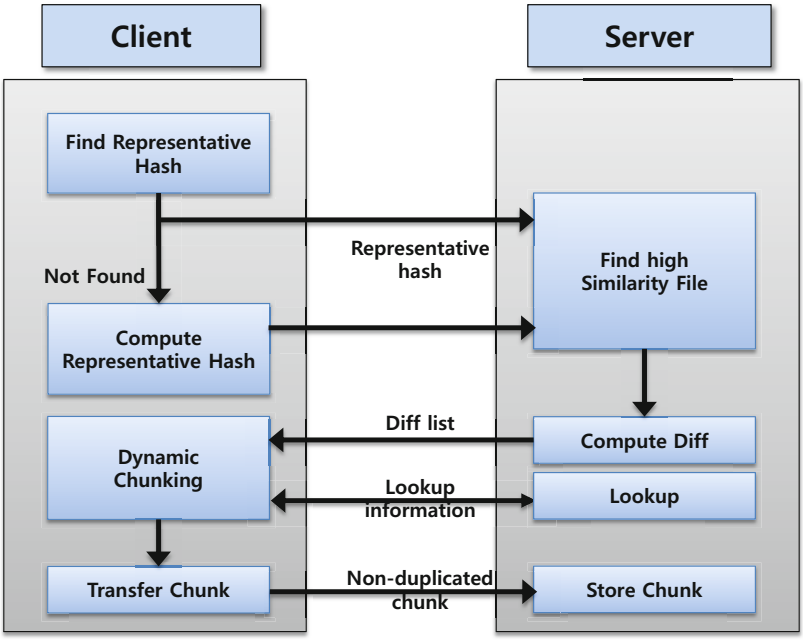


Fig. 3. Dynamic chunking processing concept

The client sends lookup information that is generated from dynamic chunking. The server check the lookup list and sends non-duplicated chunk information to the client. Finally, the client sends non-duplicated data blocks to the server.

3.3 Similarity Based Dynamic Chunking Algorithm

In this work, we implemented file similarity-based deduplication system with fixed-length chunking, called dynamic chunking. Fixed-length chunking lets files be divided into a number of fixed-sized blocks, and then applies hash functions to extract a hash key of the blocks. The reason why we used fixed-length chunking is two-folds: First, fixed-length chunking is very light-weight for data deduplication in term of processing time. Second, fixed-length chunking is easy to implement for evaluation purpose than variable-length chunking. Algorithm 2 explains how data deduplication works with file similarity information. ArrayServer contains

hash key set on the server and has three elements including offset, hash key and diff value. In DifferenceCheck function, the location of identical block is listed on the ArrayServer[i].diff and this information is used for fixed-length chunking. DuplicationCheck() function performs data deduplication with fixed-length chunking using the information of ArrayServer and ArrayClient. First, it reads the file offset from ArrayClient and adds the difference from ArrayServer, which makes new location for fixed-length chunking. In the fixed-length chunking, we can find duplicated blocks using hash key comparison. The key idea of this algorithm is to find the exact location where fixed-length chunking happens. Therefore, we can avoid unnecessary hash comparison for data deduplication.

Algorithm 2. Similarity bassed fixed-length chunking algorithm

```
Input: ArrayServer, ArrayClient, FD
begin
    for i ← 0 to ArrayServer.length do
        offset ← ArrayClient[i].offset + ArrayServer[i].diff;
        while offset < Length(FD) do
            block ← readblockFD, offset);
            hash ← ComputeSha1(block);
            if lookup(hash) = true then
                HashList.add(hash, offset, true);
            else
                break;
            end
        end
        while offset < Length(FD) do
            offset ← seek(FD, offset - blocksize) block ← readblockFD, offset);
            hash ← ComputeSha1(block);
            if lookup(hash) = true then
                HashList.add(hash, offset, true);
            else
                break;
            end
        end
    end
end
```

On both the client and the server, the server must index a set of files to recognize data chunks. It can avoid sending data blocks over the network. To save chunk transferring, the proposed system relies on the collision resistant properties of the SHA-1 hash function. The probability of two inputs to SHA-1 producing the same output is far lower than the probability of hardware bit errors. Thus, our system follows the widely-accepted practice of assuming no hash collisions. If the client and server both have data chunks producing the same SHA-1 hash, they assume the two are really the same chunk and avoid transferring its contents over the network. The central challenge in indexing file

chunks to identify commonality is keeping the index a fixed size while dealing with shifting offsets.

4 Performance Evaluation

In this work, we developed a deduplication storage system and evaluate the performance of the proposed algorithm. The server and the client platform consist of 3 GHz Pentium 4 Processor, WD-1600JS hard disk, 100 Mbps network. The software is implemented on Linux kernel version 2.6.18 Fedora Core 9. To perform comprehensive analysis on similarity based deduplication algorithm, we implemented several deduplication algorithms for comparison purpose including Fixed-length Chunking (FLC), File Similarity-based Fixed-length Chunking (FS-FLC) and Variable-length Chunking(VLC). We made experimental data set using for modifying a file in a random manner. In this experiment, we modified a data file using lseek() function in Linux system using randomly generated file offset and applied a patch to make test data file. The SRP(Space Reduction Percentage) ratio of the data file is fixed 50%. For each run, we did multiple runs with different data sets, and plot the average resulting value.

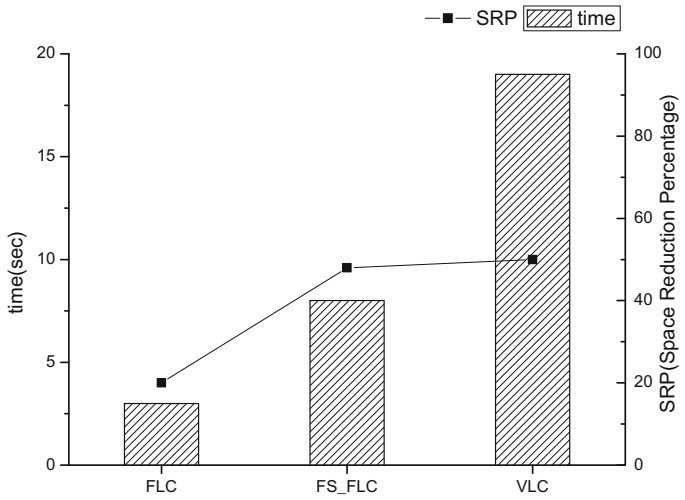


Fig. 4. Evaluation result for processing time and SRP

Figure 4 depicts the processing time and space reduction percentage graph when we applied FLC, FS-FLC and VLC. As can be seen, FLC scheme shows very fast processing time for handling data deduplication, however the SRP result is poor than other scheme. VLC shows very high SRP result while it takes very long time for processing data deduplication. The proposed scheme(FS-FLC) shows fast and high SRP result compared with other approaches. The significant result is SRP result between FS-FLC and VLC.

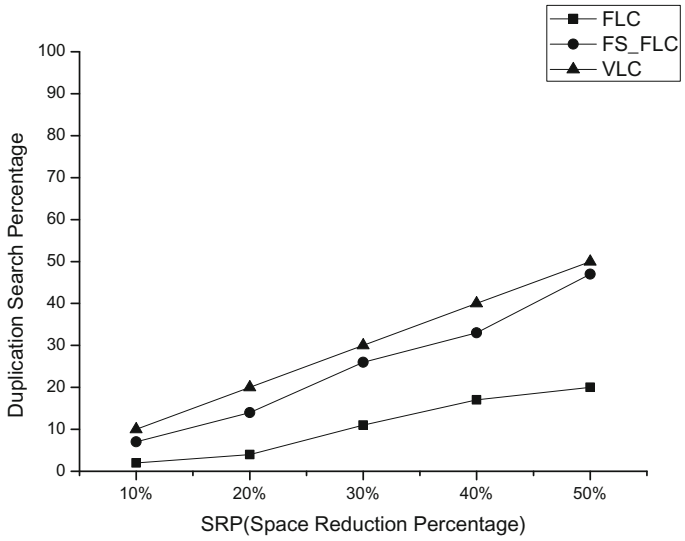


Fig. 5. Evaluation result varying SRP value

Figure 5 demonstrates the evaluation result for detecting how much duplicated data while varying SRP value from 10% up to 50%. FLC shows poor performance, it only detect less than 20% for 50% SRP data. However, VLC and FS-FLC can detect almost all duplicated data blocks. We believe that the evaluation result shows significant impact of the proposed system.

5 Conclusion

The well-known data deduplication algorithms are divided into fixed-length chunking and variable-length chunking. The fixed-length chunking is very fast for processing data deduplication but degrades the deduplication performance. However, the variable length chunking can achieve significant data deduplication performance with high computation overhead and longer processing time. In this paper, we suggest a dynamic chunking approach that overcomes the inherent problem of fixed-length chunking by adapting file similarity technique. The key idea of this work is to find several duplicated point by comparing hash key value and file offset within file similarity information. The proposed system shows significant performance improvement in processing time comparable to that of FLC. Also it shows high deduplication capability comparable to that of VLC.

Several issues remain open. First, our work has limitations on supporting simple data file which has redundant data blocks with spatial locality; therefore, if the file has several modifications then overall performance will be degrade. For future work, we plan to build a massive deduplication system with huge number of files. In this case, handling file similarity information needs more elaborated scheme.

References

1. Mokadem, R., Hameurlain, A.: An efficient resource discovery while minimizing maintenance overhead in sdds based hierarchical dht systems. International Journal of Grid and Distributed Computing 4(3), 1–23 (2011)
2. Bagchi, S.: Vmdfs: Virtual memory based mobile distributed file system. International Journal of Multimedia and Ubiquitous Engineering 2(3), 1–14 (2007)
3. Jiang, H., Li, J., Li, Z., Bai, X.: Efficient large-scale content distribution with combination of cdn and p2p networks. International Journal of Hybrid Information Technology 2(2), 4 (2009)
4. Tridgell, A.: Efficient algorithms for sorting and synchronization. PhD thesis, The Australian National University (1999)
5. Clements, A., Ahmad, I., Vilayannur, M., Li, J.: Decentralized deduplication in san cluster file systems. In: Proceedings of the 2009 Conference on USENIX Annual Technical Conference, p. 8. USENIX Association (2009)
6. Quinlan, S., Dorward, S.: Venti: a new approach to archival storage. In: Proceedings of the FAST 2002 Conference on File and Storage Technologies, vol. 4 (2002)
7. Muthitacharoen, A., Chen, B., Mazieres, D.: A low-bandwidth network file system. ACM SIGOPS Operating Systems Review 35(5), 174–187 (2001)
8. Jung, H.M., Park, W.V., Lee, W.Y., Lee, J.G., Ko, Y.W.: Data Deduplication System for Supporting Multi-mode. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011, Part I. LNCS, vol. 6591, pp. 78–87. Springer, Heidelberg (2011)

Applying MapReduce Framework to Peer-to-Peer Computing Applications

Huynh Tu Dang, Ha Manh Tran, Phach Ngoc Vu, and An Truong Nguyen

Computer Science and Engineering, International University - Vietnam National University
{dhtu,tmha,vnphach,ntan}@hcmiu.edu.vn

Abstract. MapReduce is a programming framework for processing large amount of data in distribution. MapReduce implementations, such as Hadoop MapReduce, basically operate on dedicated clusters of workstations to achieve high performance. However, the dedicated clusters can be unrealistic for users who infrequently have a demand of solving large distributed problems. This paper presents an approach of applying the MapReduce framework on peer-to-peer (P2P) networks for distributed applications. This approach aims at exploiting leisure resources including storage, bandwidth and processing power on peers to perform MapReduce operations. The paper also introduces a prototyping implementation of a MapReduce P2P system, where the main functions of peers contain contributing computing resources, forming computing groups and executing the MapReduce operations. The performance evaluation of the system has been compared with the Hadoop cluster using the prevailing word count problem.

Keywords: MapReduce, Peer-to-Peer, Distributed Computing.

1 Introduction

MapReduce [1] is an autonomic parallelization and distribution framework for processing large amount of data. The framework basically contains two operations: the *map* operation takes a list of input key-value pairs and produces a list of intermediate key-value pairs; the *reduce* operation merges the intermediate values which have the same key together to produce the meaningful output to the users. Several MapReduce implementations including Google MapReduce and Hadoop MapReduce are based on dedicated clusters of workstations to achieve high performance. In the MapReduce clusters, a client submits a job including the map and reduce operations to a server acting as the master which is responsible for distributing data and tasks to other servers acting as slaves. After finishing the tasks, the slaves return their output to the master which is also responsible for producing the result to the client. Although providing a lot of advantages, the MapReduce clusters combined with the distributed file systems can be unrealistic for the users who infrequently have a demand of solving large distributed problems due to the high cost of establishing and maintaining the clusters.

Peer-to-Peer (P2P) networks contain several remarkable characteristics including self-organization in management, scalability in architecture, and flexibility in search in decentralized and federated environments. P2P applications largely concentrate on

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 69–78, 2012.

resource sharing and lookup such as multimedia files sharing [2, 3], resource searching and retrieval [4–6]. Especially, P2P distributed computing has also been addressed by multiple research activities [7–9]. Peers can contribute computing resources, such as storage, bandwidth and processing power, and establish peer groups for solving distributed problems. However, the applicability of P2P distributed computing encounters several performance issues. Peers possess unstable and heterogeneous bandwidth and processing power while solving distributed problems may require computing resources with some degree of stability and reliability. Peers reduce high communication load by performing computing tasks individually on the local database, while solving distributed problems may require close collaboration between peers.

We propose an approach of applying the MapReduce framework on P2P networks for solving distributed problems. This approach not only employs the MapReduce framework but also exploits the characteristics of P2P technology for distributed computing purposes. The advantages of this approach are to exploit leisure resources on peers rather than using the dedicated clusters and to provide a distributed computing testbed for users who infrequently have demands of solving large distributed problems. In this approach, capable peers are invited to form the MapReduce groups as the MapReduce clusters. These peers use their computing resources to complete the assigned map and reduce operations. Multiple issues addressed in this approach include peer heterogeneity, peer communication for group formation and task assignment, and data distribution. The contribution is thus threefold:

1. Proposing a feasible approach of applying the MapReduce framework to solving large distributed problems on P2P networks
2. Extending the Gnutella P2P protocol to enable the MapReduce operations on peers, and providing a mechanism of distributing data sets on peers
3. Implementing and evaluating a prototype of the MapReduce P2P system

The rest of the paper is structured as follows: the next section includes some background of P2P networks and related study of the MapReduce framework on P2P networks. Section 3 describes the design architecture of the proposed MapReduce P2P system that solves the above issues of the approach. Section 4 presents the prototype implementation of the system based on the Gnutella protocol. The performance evaluation is reported in Section 5 with some explanation and comparison to the Hadoop MapReduce implementation before the paper is concluded in Section 6.

2 Related Work

A P2P network contains a large number of networked computers that share resources including storage, bandwidth and processor power to provide services. The P2P network has the capability of maintaining the stability of an overlay network where peers dynamically join and leave. This network also has advantages in reducing collaboration cost through ad-hoc communication process and providing high fault-tolerance and scalability. P2P networks are classified by structured and unstructured networks.

The structured P2P network is tightly controlled in topology and a peer is fixed in a logical location when connecting to other peers. This kind of networks uses Distributed Hash Table (DHT) to generate uniquely consistent identifiers for peers and resources

such that the peers hold the resource indexes if their identifiers are in the same identifier space. Lookup queries are forwarded to the peers which are closer to the resources in the identifier space. The prevailing structured P2P systems are CAN [10], Chord [11], Kademlia [12], ... The unstructured P2P network is loosely controlled in topology and a peer connects to other peers in a random fashion. Each peer maintains a list of resources in the local repository. Flooding-based search is a common mechanism used to find resources in this kind of networks. Peers send queries to the neighboring peers for queryhits. The key disadvantage of these networks is severe scalability problem as the number of queries and peers increase. The prevailing unstructured P2P systems are Gnutella [13], Freenet [14], BitTorrent [2], ...

The super peer P2P network is a hybrid network that combines the characteristics of the P2P network with the client-server network to address the problem of heterogeneous peers, i.e., peers possess various capability of storage, bandwidth and processing power. The study of Yang et al. [15] has presented guidelines for designing the super peer network to take advantage of peer capabilities. The super peer network comprises many clusters connected to each other to form either structured or unstructured P2P networks, in which each cluster contains a super peer and a set of clients. The clients submit queries to, and also obtain queryhits from, their super peer while the super peers forward the queries and receive the queryhits on the super peer network. The latest version of the Gnutella protocol has included this super peer concept.

The study of Fabrizio et al. [16] focuses on managing MapReduce applications in dynamic distributed environments, such as Grid or P2P. In Internet-based computing environments, failures are likely to happen since peers join and leave the network at an unpredictable rate. The study deals with managing intermittent peer participation, master failure and job recovery issues of the MapReduce framework that can be applied to computational Grids or P2P systems. The study has included a proposal of the P2P-MapReduce architecture, where each peer can act as either master or slave, thus creating a pool of backup masters. In case of the master failure, the backup master is promoted to the master by the election mechanism of the backup masters. Although the proposed system handles the master failure and job recovery, the system still suffers from the problem of heterogeneous peers. Peers are different in storage, bandwidth and computing power, thus choosing the master based on the smallest workload seems inefficient. When using this system for solving large distributed problems, the master failure causes the new master to be elected and several operations to be recovered, the system thus wastes a lot of time and effort for election and recovery, increases the complexity of job recovery management, and reduces the performance of solving the problems. Moreover, using the JXTA open source package to build the structured P2P network that tightly controls peers and resources, the system encounters difficulty in choosing capable peers.

Our approach proposed in this study aims at exploiting leisure resources including storage, bandwidth and processing power on peers to deal with the problem of peer heterogeneity, peer group formation, task assignment and data distribution for P2P computing applications. The approach focuses on extending the Gnutella protocol to enable the MapReduce operations on capable peers, which can be used for searching and retrieving resources more efficiently on P2P networks.

3 Architecture

A super peer described in the Gnutella protocol version 0.6 needs to satisfy some requirements, such as being not behind a firewall, having sufficient bandwidth, uptime, and processing speed. In the Gnutella networks, each peer only connects to one of the super peers while the super peers connect to each other and to the peers. The super peers also act as proxies to the Gnutella network for the peers. Queries are forwarded among the super peers using various routing mechanisms. When receiving the queries, the super peers only forward the queries to the peers that match the queries' search keys. The super peers increase the scalability of the Gnutella network by reducing the number of the incapable peers incorporating in the query routing and thus reducing network traffic. The super peers also resolve heterogeneity problem.

Based on the advantages of the Gnutella network, we design a MapReduce-P2P system that only uses the super peers to perform MapReduce operations. The Gnutella protocol is extended to enable the MapReduce operations on the Gnutella networks by adding four types of messages: *group*, *join*, *mrinit*, and *mrfin*. The group and join messages work as the request and response to form the MapReduce group, whereas the mrinit and mrfin messages work as the request and response to handle the MapReduce task execution. For the sake of simplicity, we refer MapReduce group, MapReduce task and super peer as group, task and peer respectively in this study.

```
Gnutella message header
+-------------------------------------------------------+
| message id | descriptor | ttl | hops | payload length |
+-------------------------------------------------------+
```

A Gnutella message consists of header and content. The attributes of the Gnutella message's header are shown above. The *message id* field is used to detect whether a message already arrived at a particular peer before. The *payload descriptor* field indicates the types of messages such as, ping, pong, query, ... The *ttl* field is the number of times that the message can be forwarded in the network whilst the *hops* field is the number of times that the message has been forwarded. The *payload length* field is the size of the content in byte. Immediately following the header is the message's content.

The group and join messages contain the *group id* field which is the unique identifier of a group in the network. While connecting to the network, a peer maintains a pool of connections to the neighboring peers. To form a group, the peer sends the group messages to the neighboring peers that respond with the join messages including the same group identifier of the group message and their addresses if they agree to join in the group. These peers also forward the group messages to other peers. The peer initiating the group becomes the master while the respondents become the slaves.

```
group/join                 mrinit                            mrfin
+----------------------+   +-----------------------------+   +------------------+
| group id | ip address |   | task id | task | input path |   | task id | output |
+----------------------+   +-----------------------------+   +------------------+
```

The mrinit and mrfin messages contain the *task id* field which is the unique identifier of a task. The *task* field specifies a part of the MapReduce operations related to processing the input data. The *input path* field instructs the slaves to retrieve the input data.

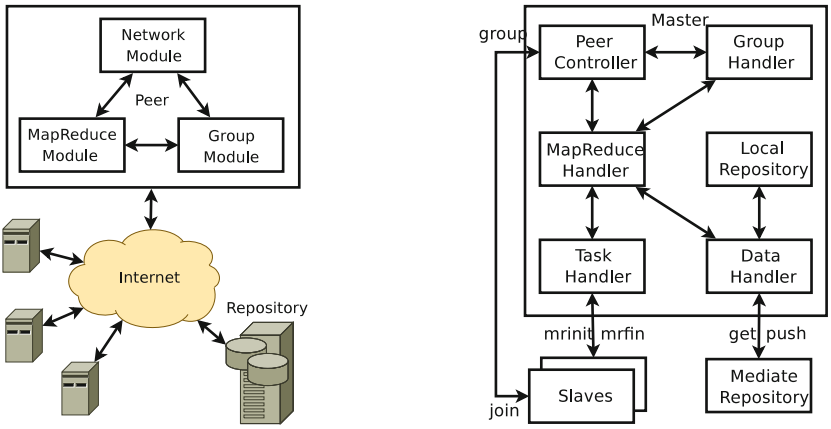


Fig. 1. MapReduce-P2P design architecture (left) and component implementation (right)

When finishing the tasks, the slaves send the results together with the task identifiers to the master in the mrfin messages.

A typical peer in the MapReduce-P2P system architecture as shown in Figure 1(left) comprises three key modules. *Network module* is responsible for communication between peers and modules . This module exchanges information to both peers and modules depending on various types of messages, e.g., a response of joining a group forwarded to the group module or a request of checking peer alive forwarded to the neighboring peers. *Group module* is responsible for group formation and management. This module manages groups to which the peer acting as either the master or the slave belongs. It cooperates with the network and MapReduce modules to maintain the stability of groups and provide the information of group members. *MapReduce module* controls the MapReduce operations on the groups. This core module interacts with the network and group modules to obtain the information of the operations and groups. It also involves executing and assigning tasks, retrieving and distributing data sets for the peer and groups respectively, and maintains the local repository. Besides, the system requires some mediate repositories for the master and the slaves to upload and download the input data sets. Note that the design description of the peers is based on the functions of both the master and the slave because the peers are symmetric in roles.

4 Implementation

We implement new messages to support the MapReduce operations for the Gnutella protocol. Each peer in the extended Gnutella network is capable of forming groups and executing the MapReduce operations. When a peer wants to solve a distributed problem, it sends the group messages to other peers to form a group. Peers receiving the requests can either reject or accept to join in the group by sending the join messages. The peer initiating the group becomes the master while the other peers of the group

become the slaves. The master manages the task assignment and data distribution. The slaves perform the assigned tasks on the retrieved data and send the processed data to the master. The implementation of the MapReduce-P2P system focuses on three main modules that contain several components as shown in Figure 1(right).

Network module contains a peer controller responsible for dispatching various types of messages to appropriate peers and handlers. The peer controller running on a separate process manages several threads for peer activities. One thread keeps track of the status of all connected peers and notifies if a peer fails to respond. Another thread maintains the stable number of the neighboring peers or the group members by connecting to a list of preference hosts recorded previously from the P2P network. When performing the MapReduce operations, the other two threads are created to control the message exchanges among this controller, the group and MapReduce handlers for group management and task execution.

Group module contains a group handler responsible for managing groups and providing group information to other handlers. When a peer deals with the MapReduce operations, the group handler running on a separate process starts a thread to initiate a group with a unique identifier. Combining with the peer controller, this thread requests other peers to join the group by spreading out the group messages. Whenever receiving the join messages, the thread adds the respondents to the group as the slaves. The group handler also uses another thread to exchange messages with the MapReduce handler. These messages aim to provide the information of the group for task assignment.

MapReduce module contains MapReduce, task and data handlers which control the MapReduce operations. Depending on the role of the peers, these handlers can function differently. For the master, the MapReduce handler invokes the data handler to split the input data stored in the local repository. The data handler splits the input data into fixed-size chunks of 64 MB and pushes the chunks to the mediate repository. The MapReduce handler obtains the slaves from the group handler and the data locations from the data handler. It then assigns each slave with a data location and a task. The task handler manages the execution of the slaves. It sends the mrinit messages including the task information and the data location to the slaves. When the task handler receives the mrfin messages, it forwards the results to the MapReduce handler to combine the final result. For the slaves, the MapReduce handler extracts the data location in the mrinit message and invokes the data handler to download the data chunk from the mediate repository. The data handler stores the data chunk in the local repository. The task extracted from the mrinit message is the pre-defined task in each slave, e.g., a word count task. The task handler performs the assigned task on the data chunk, and sends the result to the master through the mrfin message.

5 Evaluation

We use the word count problem as a distributed computing problem to evaluate the MapReduce P2P system. Algorithms 1 & 2 present the *mapper* and *reducer* operations respectively for this problem that counts the number of occurrences of every word in a text collection. The document and its unique identifier form a key-value pair, where the key is the document's identifier and the value is the document's content. The first

algorithm takes this key-value pair, tokenizes document, and produces an intermediate key-value pair for every word: the word is the key and number one serves as the value. All the intermediate key-value pairs are sorted and hashed into buckets. Note that the key-value pairs with the same key are placed in the same bucket. The second algorithm simply sums up all counts associated with each word and then emits the final key-value pairs with the word as key and the count as the value. The final result can be used as the input data for the subsequent MapReduce programs.

Algorithm 1: Mapper	**Algorithm 2:** Reducer
Input: id: document identifier Output: I: list of {term, count} pairs	Input: B: buckets of {term, [counts]} pairs Output: F: list of {term, count sum} pairs
1 $I \leftarrow \emptyset$ 2 **for each** term $t \in id$ **do** 3 $\quad I \leftarrow I \cup \{\{t, 1\}\}$ 4 **end for** 5 **return** I	1 $F \leftarrow \emptyset$ 2 **for each** element $\{t, [c_1, c_2, ...]\} \in B$ **do** 3 $\quad F \leftarrow F \cup \{\{t, sum(c_1, c_2, ...)\}\}$ 4 **end for** 5 **return** F

Several experiments focus on evaluating the feasibility and performance of this system. Peer groups including the master and several slaves are configured by 2 to 8 peers. Data sets are configured by 50 to 400 MB. We use the ftp servers for uploading and downloading data sets, and these servers can be installed on peers or separate servers. Peers connect each other within the computer laboratories of the university. Figure 2(left) depicts the network topology of a peer group for the experiments.

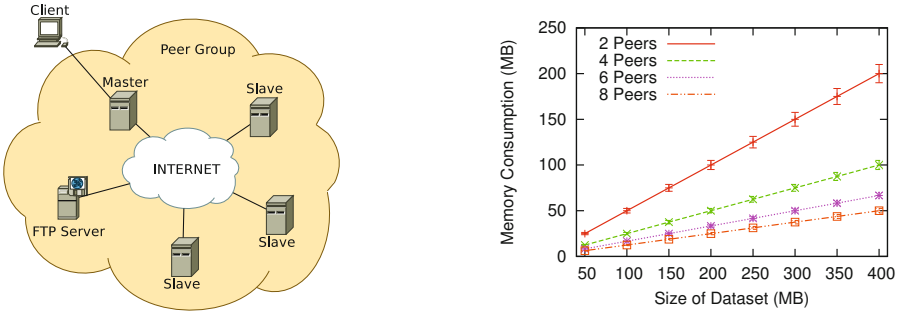


Fig. 2. Establishment of a peer group for the MapReduce operations (left). Comparison of memory usage of the peer groups (right).

The first experiment compares the memory usage of various peer groups. The memory usage is measured by the maximum memory usage of the slaves. Figure 2(right) shows that the peer groups with several peers use less memory as the size of data sets increases. With the data set of 400 MB, the 2-peer group allocates approximately 200 MB per each peer, while the 8-peer group only allocates approximately 50 MB per each peer. The approximation helps to assess the performance of the peer groups, such

as computation time, communication time and download time. The peer groups take advantage of the large number of peers to improve performance. However, multiple peers can increase communication cost and peer failure possibility that cause a profound impact on the performance.

The second experiment compares the execution time of various peer groups. The execution time is measured on the master. Figure 3(left) shows that except for the 2-peer group that performs poorly, the remaining groups tend to perform similarly when the size of data sets is either too small (<50 MB) or too big (>400 MB). With fewer peers in groups, each peer has to process larger data sets as the size of data sets increases, thus increasing download time and computation time significantly. However, groups with more peers increase communication time because the master has to wait for the slaves to return their results. It is more efficient to use the groups of several peers because the download time can be reduced on peers.

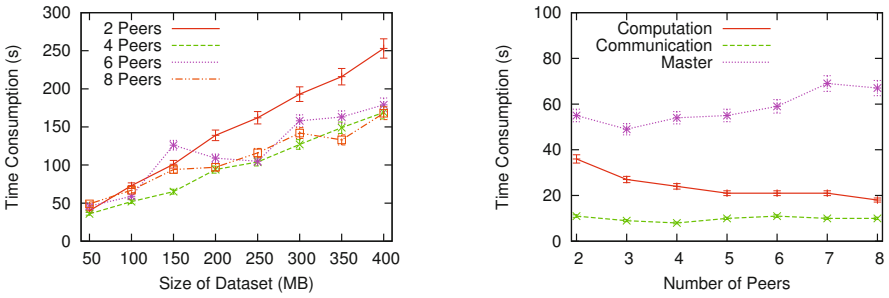


Fig. 3. Comparison of execution time of the peer groups (left). Comparison of execution, communication and computation times of the peer groups with the data set of 100 MB (right).

The third experiment investigates the communication time and computation time of various peer groups. The size of data set is chosen by 100 MB. The computation time is measured by the maximum computation time of the slaves, while the communication time is measured by the maximum download time and result sending time of the slaves. Figure 3(right) shows that the computation time reduces as the number of peers increases, while the communication time is stable and small. However, the execution time of the master contributes more significantly to the overall performance due to the waiting time of the results sequentially returned by the slaves.

The fourth experiment compares the execution time of the peer groups and the Hadoop groups. The number of peers is chosen by 2 and 8 peers. Figure 4(left) shows that the peer groups outperform the Hadoop groups. The Hadoop 2-node and 8-node groups spend 750s and 380s respectively to process the data set of 400 MB, while the 2-peer and 8-peer groups only spend 250s and 180s respectively to process the same data set. These groups are different in distributing data sets: the Hadoop groups using a distributed file system and the peer groups using a ftp server. The Hadoop groups are efficient with the large number of nodes, and the peer groups may encounter

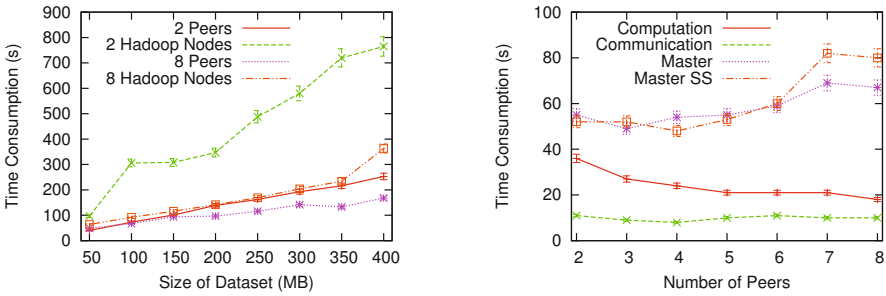


Fig. 4. Comparison of execution time of the peer groups and the Hadoop groups (left). Comparison of execution time of the peer groups on the separate ftp server (right).

performance reduction when running on the Internet setting because of the effect of the communication time.

The fifth experiment compares the execution time of the peer groups with the separate ftp server. This experiment is similar to the third experiment except the fact that the master and the ftp server are installed on separate computer nodes, thus the master needs to upload the data sets to the ftp server. Figure 4(right) shows that the execution time of the master on the separate node (Master SS) performs worse than the execution time of the master on the same node (Master), while the computation time and communication time remain unchanged on the slaves. Uploading the data sets to the ftp server causes some delay on the master, and thus affecting the performance of the peer groups.

6 Conclusions

We have proposed and implemented an approach of applying the MapReduce framework on P2P networks for distributed computing applications. The approach focuses on exploiting leisure resources including storage, bandwidth and processing power on peers to perform MapReduce operations. We have extended the Gnutella P2P protocol to enable peer group formation, MapReduce task assignment and data distribution. The peer groups can solve the word count problem in the distributed setting using ftp servers for data distribution. The experimental evaluation of the peer groups discloses several advantages. The master and slaves (peers) can complete the assigned tasks using reasonable peer resources. Using the separate ftp server to distribute data sets can obtain low time consumption. The peer groups outperforms the Hadoop groups with the same number of nodes (peers). However, the peers can fail when executing the tasks, causing the failure of the peer groups. Uploading data sets to the separate ftp server and waiting for the results from the slaves are time consuming. Future work focuses on evaluating the peer groups on large scale settings with many peers and realistic problems. The failure management model will be provided to solve the problem of peer failure.

Acknowledgement. This research activity is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2011.01.

References

1. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM 51, 107–113 (2008)
2. Cohen, B.: Incentives Build Robustness in BitTorrent. In: Proc. 1st Workshop on Economics of Peer-to-Peer Systems (2003)
3. Heckmann, O., Bock, A., Mauthe, A., Steinmetz, R.: The eDonkey File-Sharing Network. In: Proc. GI Jahrestagung (2), pp. 224–228 (2004)
4. Berkovsky, S., Kuflik, T., Ricci, F.: P2P Case Retrieval with an Unspecified Ontology. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 91–105. Springer, Heidelberg (2005)
5. Faroo, http://www.faroo.com/ (last access in January 2011)
6. Yacy, http://www.yacy.de/ (last access in January 2011)
7. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services. In: Proc. 2nd International Conference on Peer-to-Peer Computing, P2P 2002, p. 104. IEEE Computer Society, Washington, DC (2002)
8. Tatarinov, I., Ives, Z., Madhavan, J., Halevy, A., Suciu, D., Dalvi, N., Dong, X., Kadiyska, Y., Miklau, G., Mork, P.: The Piazza Peer Data Management Project. SIGMOD Rec. 32(3), 47–52 (2003)
9. Tran, H.M., Schönwälder, J.: Heuristic Search using a Feedback Scheme in Unstructured Peer-to-Peer Networks. In: Proc. 5th International Workshop on Databases, Information Systems and Peer-to-Peer Computing. Springer (2007)
10. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A Scalable Content Addressable Network. In: Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2001, pp. 161–172. ACM Press, New York (2001)
11. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In: Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2001, pp. 149–160. ACM Press, New York (2001)
12. Maymounkov, P., Mazières, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 53–65. Springer, Heidelberg (2002)
13. Gnutella Protocol Specification version 0.4 (2001), http://rfc-gnutella.sourceforge.net/developer/stable/index.html (last access in March 2012)
14. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: Federrath, H. (ed.) Anonymity 2000. LNCS, vol. 2009, pp. 46–66. Springer, Heidelberg (2001)
15. Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network. In: Proc. 19th International Conference on Data Engineering, ICDE 2003, p. 49. IEEE Computer Society, Los Alamitos (2003)
16. Marozzo, F., Talia, D., Trunfio, P.: A Framework for Managing MapReduce Applications in Dynamic Distributed Environments. In: Proc. 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing, pp. 149–158. IEEE Computer Society, Los Alamitos (2011)

Scalable Adaptation of Web Applications to Users' Behavior

Krzysztof Węcel, Tomasz Kaczmarek, and Agata Filipowska

Department of Information Systems, Faculty of Informatics and Electronic Economy, Poznań University of Economics
{k.wecel,t.kaczmarek,a.filipowska}@kie.ue.poznan.pl

Abstract. In this paper we present a comparative study of performance of an adaptive e-banking Web application supporting personalization either on a client or on a server side. Currently, modern applications being developed support various kinds of personalization. One of its types is changing behavior and appearance in response to actions taken by a user. Not only pre-defined rules but also new patterns discovered for different levels of events should be applied. Scaling such "interactive" applications to a large number of users is challenging. First, the stream of events generated by users' actions may be huge, and second, processing of the adaptation rules per single user requires computing resources that multiply with the number of users.

This paper reports on the efficiency of the method enabling a client-side adaptation after moving adaptation logics from a server to a client.

1 Introduction

The adaptability and in particular development of adaptable user interface pose a number of challenges for application developers [10,4] including, among the others, the issue of scalability.

The core of every adaptable system is the user model describing user preferences expressed explicitly or implicitly, derived from her behavior. This model should be updated, when new information is delivered, which is of particular importance when the user behavior is being traced and the new patterns are further transformed into adaptation rules. These rules have to be evaluated on the constant basis, as new events are caused by the user interacting with the application.

It is a challenge for most of the Web applications to provide many users with a personalization result instantly, because all processing is traditionally conducted on the server side, and the client (browser) is only responsible for rendering the final result. For large-scale applications, it is not feasible to evaluate dozens of rules for each user on the server side, even if efficient algorithms are applied. Therefore, the problem that arises is scalability of an adaptable Web application, which, as we are to show is achievable, if rule processing is moved entirely to the client side.

The example application on which we conducted our research is a modern e-banking application implemented in the Google Web Toolkit framework, with adaptability enhancements. We conducted several experiments that confirm the efficiency of rule based approach to adaptability on the client side and show the scalability of our solution to the rule execution problem.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 79–88, 2012.

2 Related Research

The problem of adaptability and in particular adaptable user interface (or intelligent user interface) has been studied for years in the context of regular applications [10] as well as hypertext [4]. A goal of an adaptable system is to deliver content and experience that match best user's preferences, knowledge and experience level. The user may express the preferences regarding the look and behavior of the interface explicitly or implicitly (through interaction with the system and the events that the users generate). These preferences are used in the adaptation process, which may take a form of personalization or customization [1,7]. Thanks to the development of Web technologies, the client-side scripting in particular, it is possible to capture detailed events generated by the user in the browser (such as mouse movement and individual keystrokes) [3]. This however, makes the stream of events denser and exerts greater pressure on the server to process it.

In order to mine the patterns of user behavior, it is necessary to apply the classic data mining methods (for example association rules [2]). The mined patterns are then converted to rules: this is the prevalent approach to date [9]. Depending on the approach, these rules may be event-based (series of events matching the rule head result in an action being executed) or state-based (rules are sensitive to the state change of the application) with several variants of how expressive the rule formalism is adopted [5,8]. More expressive formalisms, which are able to express sequences of events are said to handle a wider range of user requested adaptations [6]. Recently, semantic techniques are being adopted to modeling user preferences and adaptations [11].

3 Personalization in the Web Application

In this section we shortly describe main assumptions and ideas behind our personalization method as well as solutions that were tested in the experiments. The e-banking Web application that we conducted our experiments on allows to conduct standard tasks (checking accounts balances, history of transactions, place money transfer orders and standing orders) and is implemented using Google Web Toolkit (GWT) framework which uses AJAX technology for asynchronous communication with server side of the application and allows for relatively easy and powerful scripting on the client side.

3.1 Personalization Types

The system supports two kinds of personalization, technical and semantic, that incur different changes within the system. Technical personalization addresses issue of changing the graphical interface, e.g. adjusting placement of controls, changing their color, the font being displayed, adding the bounding or changing the size. The semantic analysis of user behavior on the other hand allows to suggest actions to the user, which are involved with application purpose rather than its technical side. The first group of semantic adaptations concerns providing content suggestions, which works like extended version of autocompletion. After entering data in one field other fields are filled in with appropriate (related) data, which is determined based on past behavior. For example,

after a user provided a name of an organization that she would like to transfer money to, the application may fill in other fields such as amount, address of this organization or transaction description.

We also adapt the functionality of application, i.e. user receives alerts and suggestions about potential actions. Alerts remind about actions that the user might be interested to perform, e.g. transferring money to a certain institution on a given day of the month, if the system discovered such a custom in the past. Suggestions propose to perform actions that are associated with other just performed action based on their co-occurrence in the past.

Guidelines regarding the adaptation of a user interface are expressed in a form of rules. The structure of a rule is independent from the type of personalization being performed. A rule consists of a body (antecedent) and a head (consequent). The body defines conditions required to fire an action defined in the head. The condition may be: a sequence of particular events (possibly interrupted by other events), a set of events (occurring without particular order) or a time-related event. The action may cause one of the effects (i.e. adaptations) in an application: filling given control with data, change of a style of a given control, display of tool tip for a given control, suggestion or alert for action, change of order of controls.

3.2 Rules' Lifecycle

Adaptation can take place in response to an event. The event is understood as an elementary manifestation of user behavior in the system. We distinguish three kinds of events, occurring on different levels: program events (fine grained mouse or keyboard events), logical events (change of contents in controls, e.g. typing in account number), semantic events (high level operations triggered by filling in forms, e.g. placing a transfer order).

The program and logical events are generated in the browser, while semantic events occur on the server. Program and logical events have to be transferred to the server for data mining purposes. As written previously, event stream is mined for patterns of user behaviour. However, as we learned, applying raw data mining algorithms bring a lot of noise and does not render useful user behaviour patterns. For example the most frequent associations between events are the following: "when the user **opens** transfer page, he **clicks** «send» button". Such associations are the side-effect of technical organization of the application and have to be filtered out to find the actual user behaviour patterns. The behaviour pattern are transformed into adaptation rules, which are recorded in the user's profile and updated with every run of the data mining subsystem. We were able to find from dozen to almost hundred such rules for a single user based on the exemplar data gathered from test user interaction with the e-banking application.

For the purpose of the experiment we used 87 state rules and 1380 event rules. This is more or less the number of rules expected in real life applications.

Since the rules are sensitive to program and logical events, which origin at the client, it is possible to move their enactment also to the client. The semantic events are also taken into account because they are associated with logical events that precede them (for example sending a transfer - a semantic event - is not possible without opening a transfer page - a logical event). The rules are transferred to the client upon application loading,

where they can be enacted thanks to the scripting capabilities of modern browsers, as described in the next section.

3.3 Client-Side Rule Evaluation

We implemented an efficient rule execution environment in the browser, which can be nowadays done conveniently thanks to modern application Web frameworks like GWT. The implementation is based on non-deterministic finite state automaton, which is a very efficient device for capturing multiple patterns at once, via a single pass over a stream of symbols (events in our case).

Each head of the rule passed to the client was added as a pattern to be recognized by the automaton. If the rule expected a sequence of particular events, possibly interleaved with other events, it was added as a sequential pattern. If the rule expected a set of events to occur, all the possible permutations of the event sequences were calculated on the client, and added as sequences with interleaving events to the automaton. Though it might seem inefficient, the system actually did not suffer from the explosion of permutations, since the maximal number of events in the rule body was 6, which gives 120 permutations of sequences only.

After creating the automaton, it was run over the stream of events, as they occurred. To handle the non-determinism of the automaton, a standard approach was taken, where the automaton was allowed to have several active states at once, corresponding to partially matched rules. Each of these active states was tracked independently, and new were activated as needed. Such approach proved to be efficient as expected with respect to computational time, and moderately heavy with respect to memory consumption. Total consumption of memory by the browser process did not exceed 500 MB, which was shared between all the components of the application and is comparable to opening several tabs with fairly standard Web pages.

4 Tests on Efficiency of Personalization

4.1 Research Methodology

In order to check the efficiency of methods we have prepared the scenario for navigating through the application and performing simple e-banking task. The scenario was scripted using iMacros tool, that automates execution of tasks in the client browser - this ensures repetitiveness of the experiment. To simulate the static Web pages we used HtmlUnit. The scenario was executed in several variants on a typical workstation supplied with Inter Core2 Duo, T8300 @ 2.4GHz and 3.5GB of physical memory. For remote tests, 100Mbit LAN was used. Our application implemented in Google Web Toolkit (GWT) ver. 2.0.3 was deployed on Glassfish server (v 3.0.1). As a database we used Postgresql ver. 8.4; application server used the following database library postgresql-8.4-701.jdbc3.jar. All tests were executed in Mozilla Firefox.

4.2 Centralized Execution of Personalization

In this variant of the testing scenario the personalization is prepared on server. We employ HtmlUnit to simulate execution of the business logics on server, while typically

it is run on client. The goal of the test was to estimate the mean time necessary to prepare a personalized view of the application on server side as a function of a number of concurrent users. HtmlUnit browser simulator was run on server in several threads; each thread simulated the behavior of one client.

Results. Each thread generated an independent instance of HtmlUnit and it influenced amount of required memory (up to 600MB for 15 concurrent threads). The time necessary to prepare the visualization increased as the number of threads grew. The table 1 presents the detailed timing.

Table 1. Duration of the scenario depending on the number of concurrent threads

No of threads	Scenario duration in sec.
1	26,090
3	36,863
5	51,145
10	88,686
15	132,241

The dependency is also presented in Figure 1.

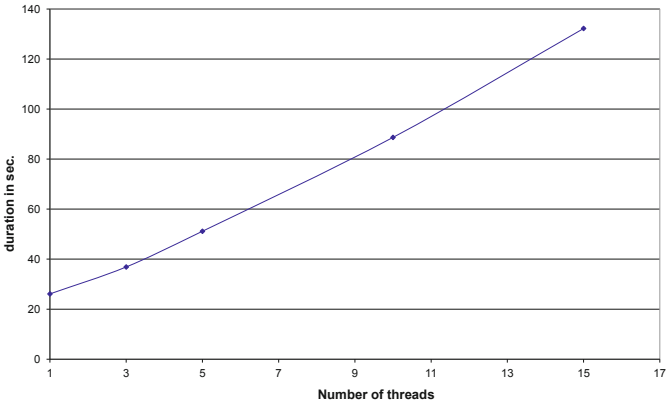


Fig. 1. Mean duration of the scenario execution with HtmlUnit depending on the number of concurrent local threads

Based on the figure we validated the hypothesis that the dependency is linear. Analysis of regression confirmed high dependence between variables. The coefficient of determination R^2, which measures proportion of variability in a data set that is accounted for by the statistical model, was 97,4%. We checked the statistical significance of the regression using Fisher-Snedecor (F-test). On the test machine, it took 19.51 sec. to prepare one personalized visualization from the scenario, and we needed 7.97 sec. more for each additional client. Each test run resulted in almost 100% CPU utilization.

4.3 Distributed Execution of Personalization

In this variant of the scenario, the personalization is prepared on clients. We employ the Web browser with appropriate scripting engine installed, so that execution can be started remotely and the server load is investigated. The goal of the test was again to estimate the mean time necessary to prepare a personalized view of the application, but this time on the client side, when several clients connect at the same time. To preserve comparability of the results we again used HtmlUnit to simulate a Web browser, but this time it was run on several machines connected in a local 100Mbit network at the same time, and each client executed the same scenario.

Results. Tests were run 8 times (8 sessions), using the following number of clients in respective sessions: 10, 10, 10, 5, 5, 5, 10, and 5. Mean execution times are within a narrow range (86 - 88 sec.), except for the first session (95 sec.) which can be attributed to necessary caching of files. Therefore, in further analysis we distinguish execution with first session (A) and without it (B). Client applications were run on identical machines, nevertheless, we have verified statistically that there is no statistically significant difference between duration of scenarios on different clients. We used univariate ANOVA analysis. For the first option (A) we have: F=0.45, p-value = 0.9 > 0.05; for the second option (B): F=1.19, p-value = 0.33 > 0.05. As p-value is substantially higher than 0.05, there is no ground to reject the hypothesis saying that all durations are equal.

The first step in analysis is the comparison of execution times depending on number of clients: 5 or 10 (two groups). Summary results are in table below.

Group	Count	Sum	Avg	Var
'5'	20	1726.2	86.309	0.451
'10'	40	3559.2	88.980	14.775

It may be noticed, that the average execution times for both groups are similar. The analysis of variance, however, proves that they cannot be assumed equal.

Source of variance	SS	df	MS	F	p-value	Test F
between groups	95.084	1	95.084	9.431	0.0032	4.007
within groups	584.780	58	10.082			
Total	679.864	59				

Moreover, the F statistics equals F_{obs}=9.431 and is higher than a critical value F=4.007. The p-value = 0.0032 < 0.05, therefore at 95% significance level the null hypothesis saying that average duration times in both groups are identical should be rejected.

To make sure that results were not biased by caching, we verified also the set with 7 sessions (option B).

Group	Count	Sum	Avg	Var
'5'	20	1726.186	86.309	0.451
'10'	30	2606.834	86.894	0.385

Paradoxically, although average times do not differ much, ANOVA again pointed out that samples could not have been obtained from the same population. This is because of the small variances, where the tolerance margin is really small.

Source of variance	SS	df	MS	F	p-value	Test F
between groups	4.109	1	4.109	10.001	0.0027	4.043
within groups	19.721	48	0.411			
Total	23.830	49				

This time, the F statistics equals F_{obs}=10.001 and is higher than a critical value F=4.007. The p-value = 0.0027 < 0.05, therefore at 95% significance level the null hypothesis saying that average duration times in both groups are identical should be rejected.

We therefore conducted analysis of regression in order to quantify dependency between number of machines and execution time of scenario.

The following equation was assumed:

$$y_2 = ax + b \tag{1}$$

where: y_2 – duration of the scenario in seconds (the dependent variable), x – number of clients (the independent variable).

The coefficient of determination R^2 shows that just a fraction of variability of dependent variable was explained by the independent variable - only 14%. Nevertheless, the regression is statistically significant (Fisher-Snedecor test F_{obs}=9.43). Values of t-Student statistics confirms that estimation of parameters a and b is also statistically significant. Therefore, we have:

$$y_2 = 0.534 * x + 83.639 \tag{2}$$

It should be compared to the parameters obtained in the previous experiment which presents execution times for local setting. Figure 2 presents such comparison.

An important benefit of the distributed variant, where visualization is prepared on the client side, over a centralized solution is just a slight increase in processing time with increasing number of clients. For hypothetical number of 10,000 clients, which is an expected number of clients using a production version of the e-banking system developed, the execution of the scenario on current hardware configuration would take respectively: 22 hours and 9 minutes for a centralized version and 1 hour and 30 minutes for the distributed version.

Conclusion: for the small number of clients the centralized solution is more efficient. With increasing number of clients a distributed approach is preferred, where the threshold in our experiment was estimated at 10 clients.

4.4 Execution of Personalization in Web Browsers

In the second experiment we used a browser, which is not efficient. In this experiment we will focus on real efficiency of the system using a typical Web browser instead of artificial interpreter like HtmlUnit. The goal of the test was to estimate the mean time necessary to prepare a personalized view of the application in Web browser, when

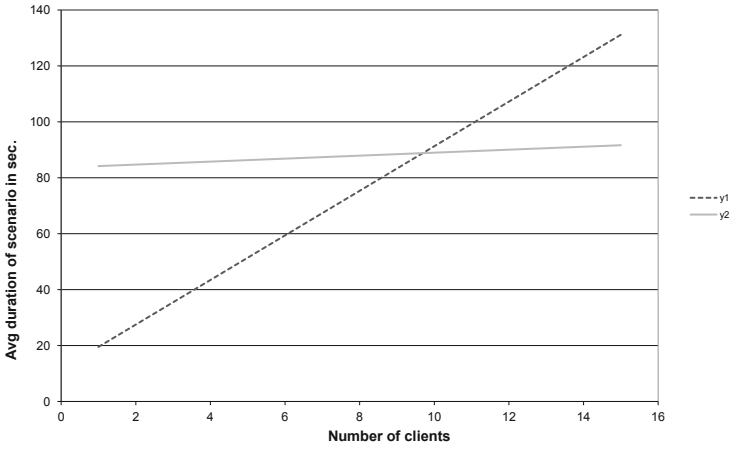


Fig. 2. Comparison of average execution times of scenario using HtmlUnit in local (y_1) and remote (y_2) setting

several clients connect to the application server at the same time. We used Mozilla Firefox with iMacros to execute the scenario. The clients were run on twenty machines on a local 100Mbit network.

Results. The conducted experiments confirmed the efficiency of JavaScript engines built into Web browsers. The execution times of scenario were significantly reduced: average time for a sample of size 72 was 3.66 sec. Shorter times can partly be attributed to a more efficient caching mechanism of Web browsers than of HtmlUnit. In order to measure the influence of parallelization on duration of scenarios, measurements were done in two variants:

- parallel – all clients connect to server at the same time,
- sequential – clients connect independently, with some delays between connections.

The null hypothesis is that execution times in these two variants do not differ. Again, ANOVA analysis was used to verify the hypothesis.

Group	Count	Sum	Avg	Var
'parallel'	17	61.22	3.601	0.547
'sequential'	55	202.11	3.675	5.376

It should be noted, that the average times seem to be identical.

Source of variance	SS	df	MS	F	p-value	Test F
between groups	0.0705	1	0.0705	0.0165	0.8981	3.9778
within groups	299.0323	70	4.2719			
Total	299.1029	71				

The observed statistics F_{obs}=0.0165 is smaller than the critical value F=3.9778 and at the same time p-value=0.898 > 0.05, therefore there is no reason to reject the hypothesis.

The **conclusion** of this experiment is as follows: when using Web browsers to connect simultaneously many clients in an experiment environment consisting of 20 machines, the concurrency (parallel vs. sequential) of connections does not affect the execution time of the scenario.

5 Conclusions and Discussion

For the implementation of the Web application we used a very efficient Google Web Toolkit engine. As a way to optimize personalization methods we proposed moving preparation of the visualization to clients. This, however, implicated moving information usually available on server (e.g. business logics) to clients as well. Thus, server is not responsible for tasks related to graphical user interface, and merely provides the clients with the data necessary to prepare visualizations locally.

In this paper we verified to which extent the distributed solution excels a centralized system. In the first variant, individual forms of corporate banking application and their adaptations were prepared in their entirety on the server. We utilized HtmlUnit, one of few possibilities to generate HTML pages on the server without modification of the Web application. In the second variant, the GUI was generated on the client side. We observed certain overheads attributed to the transfer over the network and just when a number of clients exceeded ten, the distributed solution was more efficient than the centralized one. In the third variant, the Web browsers were used as clients to generate the visualization and this solution was efficient as expected. The average time to execute the scenario was 3.6 sec. and was not significantly higher when 20 machines connected at the same time. The estimated time to execute the same scenario for 20 clients in the first variant is 171 sec. (47.5 times worse), and in the second variant – 94 sec. (26 times worse).

The obtained results are significant for developers who use Google Web Toolkit for application development. We have shown that in the case when application has to handle more than ca. 10 clients it is advantageous to invest time in developing a module that will be able to execute part of the business logics on client side. This is particularly true for computation intensive applications like one presented for personalization.

References

1. Adomavicius, G., Tuzhilin, A.: Personalization technologies: a process-oriented perspective. Communications of the ACM 48(10), 83–90 (2005)
2. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994), http://dl.acm.org/citation.cfm?id=645920.672836
3. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: Proceedings of the 15th International Conference on World Wide Web, WWW 2006, pp. 203–212. ACM, New York (2006), http://doi.acm.org/10.1145/1135777.1135811

4. Brusilovsky, P., Kobsa, A., Vassileva, J. (eds.): Adaptive Hypertext and Hypermedia. Springer (1998) ISBN 978-0-7923-4843-6
5. De Virgilio, R., Torlone, R., Houben, G.J.: A Rule-based Approach to Content Delivery Adaptation in Web Information Systems. In: Proceedings of the 7th International Conference on Mobile Data Management, MDM 2006, p. 21. IEEE Computer Society, Washington, DC (2006), http://dx.doi.org/10.1109/MDM.2006.16
6. Gao, C., Wei, J., Xu, C., Cheung, S.C.: Sequential event pattern based context-aware adaptation. In: Proceedings of the Second Asia-Pacific Symposium on Internetware, Internetware 2010, pp. 3:1–3:8. ACM, New York (2010), http://doi.acm.org/10.1145/2020723.2020726
7. Mueller, F., Lockerd, A.: Cheese: tracking mouse movement activity on websites, a tool for user modeling. In: CHI 2001 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2001, pp. 279–280. ACM, New York (2001), http://doi.acm.org/10.1145/634067.634233
8. Paskalev, P.: Rule based GUI modification and adaptation. In: Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, CompSysTech 2009, pp. 93:1–93:7. ACM, New York (2009), http://doi.acm.org/10.1145/1731740.1731841
9. Paskalev, P., Serafimova, I.: Rule based framework for intelligent GUI adaptation. In: Proceedings of the 12th International Conference on Computer Systems and Technologies, CompSysTech 2011, pp. 101–108. ACM, New York (2011), http://doi.acm.org/10.1145/2023607.2023626
10. Schneider-Hufschmidt, M., Kühme, T., Malinowski, U. (eds.): Adaptive User Interfaces: Principles and Practice. Human Factors in Information Technology. North Holland (1993) ISBN 978-0-444-81545-3
11. Wang, H., Mehta, R., Supakkul, S., Chung, L.: Rule-based context-aware adaptation using a goal-oriented ontology. In: Proceedings of the 2011 International Workshop on Situation Activity & Goal Awareness, SAGAware 2011, pp. 67–76. ACM, New York (2011), http://doi.acm.org/10.1145/2030045.2030061

OCE: An Online Colaborative Editor*

César Andrés[1], Rui Abreu[2], and Alberto Núñez[1]

[1] Universidad Complutense de Madrid
Departamento de Sistemas Informáticos y Computación
Madrid, Spain
c.andres@fdi.ucm.esand alberto.nunez@pdi.ucm.es
[2] University of Porto
Departament of Informatics Engineering
Porto, Portugal
rui@computer.org

Abstract. In this paper we present the development of an Online Collaborative Editor (OCE) software system. It allows several people, to edit and share computer files using different devices, such as mobiles, PDAs in an easy way.

We use formal methods in order to automatize and describe OCE. Its formalism is very suitable to specify time requirements (both time consumption due to the performance of tasks and timeouts) as well as to represent data communication among different components of the system.

This *exercise* convinced us that a formal approach to develop complex systems can facilitate some of the development phases. In particular, the testing and debugging phases, more precisely, how to chose those tests more suitable to be applied, is simplified since tests are automatically extracted from the specification.

Keywords: Cooperative Systems, Software Development, Collective Intelligence, Social Editing.

1 Introduction

Over the last 15 years, researchers within universities have been developing technologies for automated feedback in *Software Engineering* courses. Software Engineering can be considered as a systematic and disciplined approach to developing software. It concerns all the aspects of the production cycle of software systems and requires expertise, in particular, in data management, design and algorithm paradigms, programming languages, and human-computer interfaces. It also demands an understanding and appreciation for systematic design processes, non-functional properties, and large integrated systems. Thus, when developing

* Research partially supported by the European project TOCE: Testing Online Colaborative Editors, funded by the Bilateral Luso-Spanish Programme and by the the Spanish MCYT project TESIS project (TIN2009-14312-C02-01).

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 89–98, 2012.

complex systems, it is necessary to apply sound engineering principles in order to economically obtain reliable and efficient software.

Formal methods refer to techniques based on mathematics for the specification, development, and verification of both software and hardware systems. The use of formal methods is especially important in reliable systems where, due to safety and security reasons, it is important to ensure that errors are not included during the development process. Formal methods are particularly effective when these are used early in the development process, at the requirement and specification levels, but can be used for a completely formal development of a system. One of the advantages of using a formal representation of systems is that it allows to rigorously analyze their properties. In particular, it helps to establish the *correctness* of the system with respect to the specification or the fulfillment of a specific set of requirements, to check the semantic *equivalence* of two systems, to analyze the *preference* of a system to another one with respect to a given criterion, to predict the possibility of *incorrect behaviors*, to establish the *performance* level of a system, etc. In this line, formal testing and debugging techniques [10,1] can be used to test the correctness of a system with respect to a specification.

It has been argued both that formal methods are very appropriate to guide the development of systems and that, in practice, they are useless since software development teams are usually not very knowledgeable of formal methods in general, and have no knowledge at all of what academia is currently developing (see, for example, [16,11,8,3] among many others). In this paper is presented the development of an *Online Collaborative Editor*, coined OCE, using sound techniques of Software Engineering. This kind of software allows several users to share and edit simultaneously a computer file using different devices [13,12,4]. Nowadays, it is a hot topic in the Internet domain [9,7,6], and the possibility of editing and sharing documents have been recently incorporated in known text editing software such as Abiword, ACE, or Microsoft Office. Moreover, with the growing of Internet (see Figure 1), the development of these tools in a shared net area is also necessary.

In addition to the development of the tool itself, an additional contribution of this paper is that we collect the experiences obtained from the application of formal methods to the development of a complex software system. We applied a formal approach from the beginning of the project until the testing and debuging phase where we tried to establish the correctness of the developed system with respect to the specification constructed from the original requirements. Five independent groups were in charge of the different stages of the project: requirements, specification, implementation, and testing/debugging.

In total, around 50000 lines of code were generated. The first problem we found was that the formalisms we evaluated were not completely appropriate for our project. For example, the application has to be accessed via Internet. This fact implies, for instance, autonomous behaviors of the system, in order to provide a minimum level of security, or the execution of actions that do not require the participation of the user. These requirements are not easily represented by means

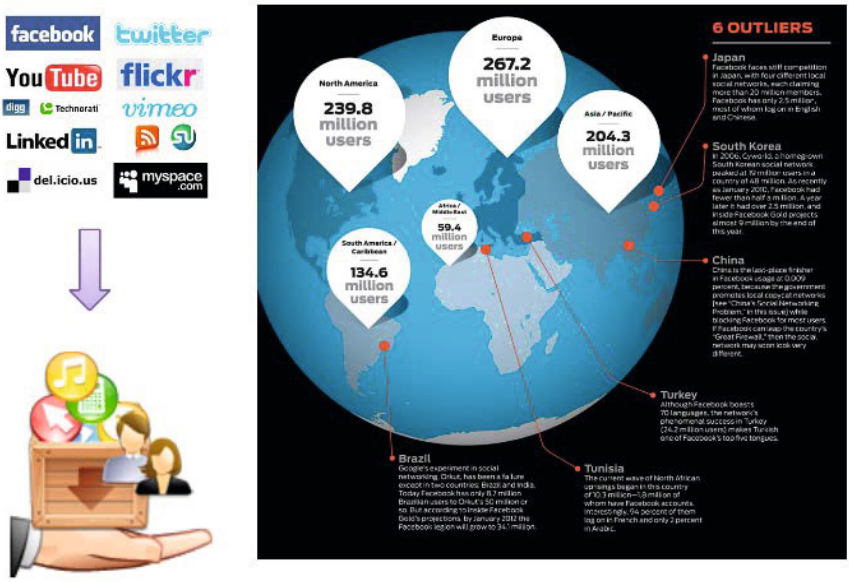


Fig. 1. Collaborative Editor Necessity

of the available formalisms, and even impossible in some cases. Actually, there does not exist a lot of work on formal approaches to specify and test either web-based systems or web services (see, for example, [2] for a recent formal approach based on EFSMs and [5] for an overview of the field). Moreover, other critical aspects must be taken into account when a web application is developed. For example, it is essential to consider the time that the system spends in producing a response to a request, while the amount of time that the system can wait in a state of inactivity can be considered critical in a system where security is essential. These considerations demanded an adaptation and extension of one of the existing formalism. Taking into account the previous considerations, we decided to use a formalism close enough to our needs. Thus, we chose as starting point the formalism proposed in [14] since it allows to represent the needed time requirements and incorporated a primitive way to exchange information among different components.

All in all, we think that the experience was positive. A formal specification allowed the implementation team to minimize the communication with the specification team. Certainly, the implementation team was composed by people, four undergraduate students plus and assistant teacher, who were familiar with formal methods. We understand that a different team could have more problems to implement from a specification, but we think that the effort to learn/remember formal methods is compensated in the implementation and testing phases. In particular, testing is strongly facilitated since tests are automatically derived from the specification. Of course, tests have to be adapted to deal with the real implementation, but this is a task that have to be done also if an informal approach is used. In fact, we found some errors in the testing phase that, we think,

could not be uncovered by either manually extracting tests from the specification or by entrusting the testers to generate tests.

The rest of the paper is structured as follows. Section 2 introduces a straightforward mechanism to specify our system and the techniques for testing and debugging that will be used in the development of the software. Next, in Section 3 we report some critical issues detected in the course of this project. Finally in Section 4 we present the conclusions and some lines of future work.

2 Formal Framework

In this section we review our formalism to model the collaborative editor. This formalism consists in defining the concept of *shared-points*. Intuitively, a shared-point is a software the user needs to download/upload his information. This definition is generic and does not take into account the hardware part presented in the different devices.

The internal behavior of a *shared point* is given by a finite state machine where, at each state, the machine can receive an input and produce an output, which corresponds to a new request, before moving to another state. Let us note that shared points send and receive messages. We assume that the communication between different shared points is asymmetric (the server of the application can be also seen as a shared point), that is, we should store the messages sent from a shared point to another one by using *lists*, presented in our framework as *input buffers*. We do not present all the formal details concerning the lists of these shared points.

Definition 1. In this paper, ID denotes the set of shared point identifiers, where – denotes the empty identifier. A *shared point* is a tuple

$$M = (id, \mathcal{S}, \mathcal{I}, \mathcal{O}, s_0, \rightarrow)$$

where $id \in ID$ is the identifier of the shared point, $\mathcal{S}$ is the set of states, $\mathcal{I}$ is the set of inputs, $\mathcal{O}$ is the set of outputs, $s_0 \in \mathcal{S}$ is the initial state and $\rightarrow \subseteq \mathcal{S} \times \mathcal{I} \times ID \times \mathcal{O} \times ID \times \mathcal{S}$ is the set of transitions. In our context, – denotes both the empty input and the empty output. □

A transition belonging to $\rightarrow$ is a tuple $tr = (s, i, snd, o, adr, s')$ where $s, s' \in \mathcal{S}$ are the initial and final states, respectively, $i \in \mathcal{I}$ is an input, $snd \in ID$ is the required sender of i, $o \in \mathcal{O}$ is an output, and $adr \in ID$ is the addressee of o. Intuitively, a transition in a shared point indicates that if the machine is in state s and receives the input i from snd, then the machine emits the output o to adr and moves to s'.

Example 1. Let us consider the shared points M_1 and M_2 depicted in Figure 2. These are two basic shared points. Client M_1 can start at any time because the first transition of the shared point M_1 is (–,–), meanwhile M_2 has to wait until it receives a message `a` from M_1, represented by the transition labeled (M_1,`a`).

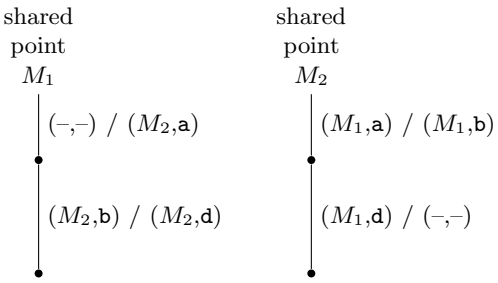


Fig. 2. Example of two shared points

A normal interaction between both shared points is described next. When M_1 starts, it sends to the M_2 the message a, and it moves from its initial state to the next step. M_2 is waiting this par (M_1,a) to trigger its initial transition. When M_2 performs its first transition, it sends to M_1 the message b, and it moves to its next state. This situation happens again with the message d. The first shared point waits for the message b from M_2 and, when it receives this message, it sends the message d to M_2, and M_1 moves to its final state. At the end, the last shared point consumes the message d and goes to its final state. □

Next, we formalize the notion of *supervisor*. This module allows us to represent the retrieval information process and how to check good and bad behaviors of the system. In particular, a supervisor focuses on representing the interaction of shared points as a whole. Thus a single machine, instead of the composition of several machines, is considered.

Let us remark that each transition of the supervisor denotes a *message action*, where some shared points sends a message to another one.

Definition 2. A *supervisor* is a tuple $\mathcal{C} = (\mathcal{S}, \mathcal{M}, ID, s_0, \mathcal{T})$ where $\mathcal{S}$ denotes the set of states, $\mathcal{M}$ is the set of messages, ID is the set of service identifiers, $s_0 \in \mathcal{S}$ is the initial state, and $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{M} \times ID \times ID \times \mathcal{S}$ is the set of transitions. □

Concerning supervisor machines, a transition $t \in \mathcal{T}$ is a tuple (s, m, snd, adr, s') where $s, s' \in \mathcal{S}$ are the initial and final states, respectively, $m \in \mathcal{M}$ is the message, and $snd, adr \in ID$ are the sender and the addressee of the message, respectively.

Example 2. In Figure 3 we represent a supervisor of a system. On the one hand, we have that if C_1 detects the exchange of message c from the shared point M_2 to the shared point M_3, then it waits for the message a from M_3 to M_2, followed by the message a from M_2 to M_3, and it finishes the interaction. On the other hand, if the supervisor observes the message a from the service M_1 to the service

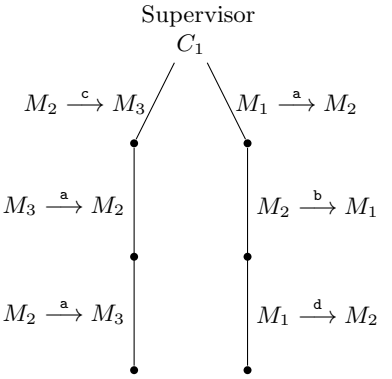


Fig. 3. Supervisor example

M_2, then it checks that the service M_2 sends the message `b` to the service M_1, and next it waits to see the message `d` from M_1 to M_2. □

The previous definitions only provides the syntax to write specifications. The rest of the theoretical machinery falls beyond the scope of this paper. In particular, the formal definition of test is quite similar to [14] while the derivation algorithm to extract tests is an adaption of the ones presented in [15,14].

3 Description of the System

In this section we present the main features of OCE and how we have used formal methods in order to increase the quality of the software. OCE runs over Apache while databases are running over MySQL. It has been implemented using Emacs Lisp. In order to illustrate the use of this language, in Figure 4 we present three functions implemented in OCE, that allow us to manage with text files in an easy and efficient way. The first function f_a, is used in OCE in order to select the word under cursor. The meaning of word here is considered as any alphanumeric sequence with "_" or "-", the second function f_b, deletes texts between any pair of delimiters. Finally, the third function, by means f_c allows the shared points to manage different buffers in order to refresh the information, and do it in a friendly way to the users. Finally, the application has been fully tested to work with as an e-macs plugging and we are quite confident that it properly works with other open-editors such as gedit.

To start using OCE we need to download at least a *client* and a *server*. The client is a set of software modules that manages the collaborative edition process integrated in a graphical editor as a plug-in. The server is a process running to serve the requests of the clients. Thus, the server performs some computational tasks on behalf of clients. In our environment, the clients either run on the same computer (identifier as localhost) or connect through Internet to the server.

f_a

```
(defun select-current-word ()
(interactive)
(let (pt)
(skip-chars-backward "-_A-Za-z0-9")
(setq pt (point))
(skip-chars-forward "-_A-Za-z0-9")
(set-mark pt)))
```

f_b

```
(defun delete-enclosed-text ()
(interactive)
(save-excursion
(let (p1 p2)
(skip-chars-backward "^([<>""")(setq p1 (point))
(skip-chars-forward "^)]<>""")(setq p2 (point))
(delete-region p1 p2))))
```

f_c

```
(defun next-user-buffer ()
(interactive)
(next-buffer)
(let ((i 0))
(while (and (string-match "^*" (buffer-name)) (< i 50))
(setq i (1+ i)) (next-buffer) )))

(defun previous-user-buffer ()
(interactive)
(previous-buffer)
(let ((i 0))
(while (and (string-match "^*" (buffer-name)) (< i 50))
(setq i (1+ i)) (previous-buffer) )))
```

Fig. 4. Some functions of OCE implemented in Emacs Lisp

Following we present a situation of the development of OCE and how we tested it. In Figure 5 the process of updating some information from one shared point to another shared point is presented. This is the scheme that we obtain from the supervisor (see Definition 2). There are three shared points. The first and the third one corresponds to different clients and the second one corresponds to the OCE server.

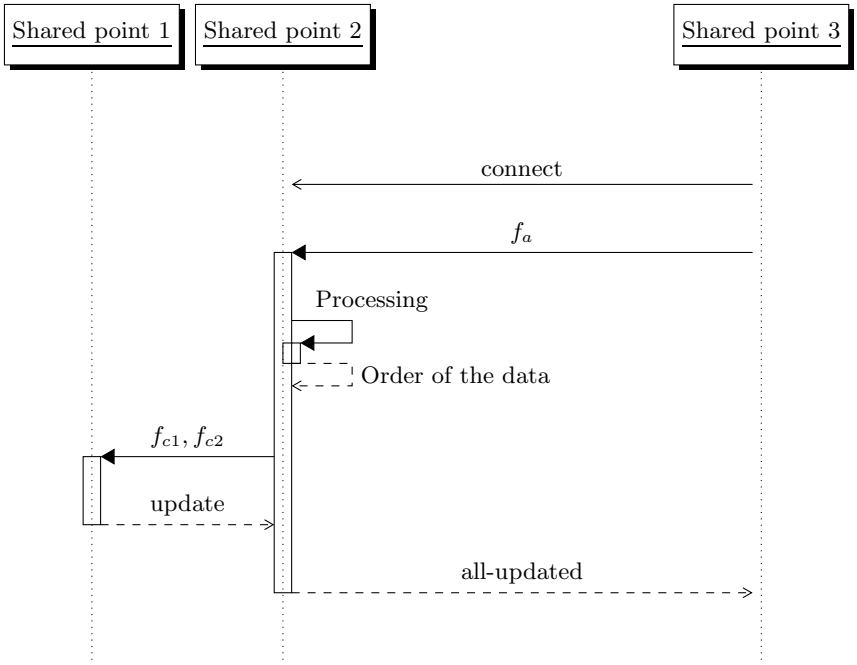


Fig. 5. Sequence diagram of the update/refresh process

The user in the third shared point `connects` to the server and introduces a sequence in his editor. This is automatically detected by OCE, and the function f_a is executed in order to obtain this word. After this, the word is sent to the server, (shared point 2); next it is processed and it is sorted inside the internal buffer of the server. Finally, this new word is sent to the shared point 1 (another user) and the functions f_{c1} and f_{c2}, that manage the buffers of the user are executed. Finally, the instance of OCE in the first shared point notifies to the second shared point that the update process was done correctly, and the server to the initial point.

Automatically we obtained a set of tests that checked the correctness of this situation. The inputs for the test where new words were introduced in the shared point 3, while the outputs were the messages that the shared point 1 was emitted. To monitor this outputs we used the wireshark tool. In Figure 6 there are presented those messages. Two errors were detected:

1. Problems of repetitive words. This situation happened when two words where sent from the shared point 3 to the shared point 1 and the user of the shared point 1 was writing in the same paragraph than the inserted words. Sometimes the text of the user of shared point 1 was duplicated, and sometimes the last information was deleted (that is, functions f_{c1} and f_{c2} were not working as expected). The problem was solved by implementing a list of received changes.

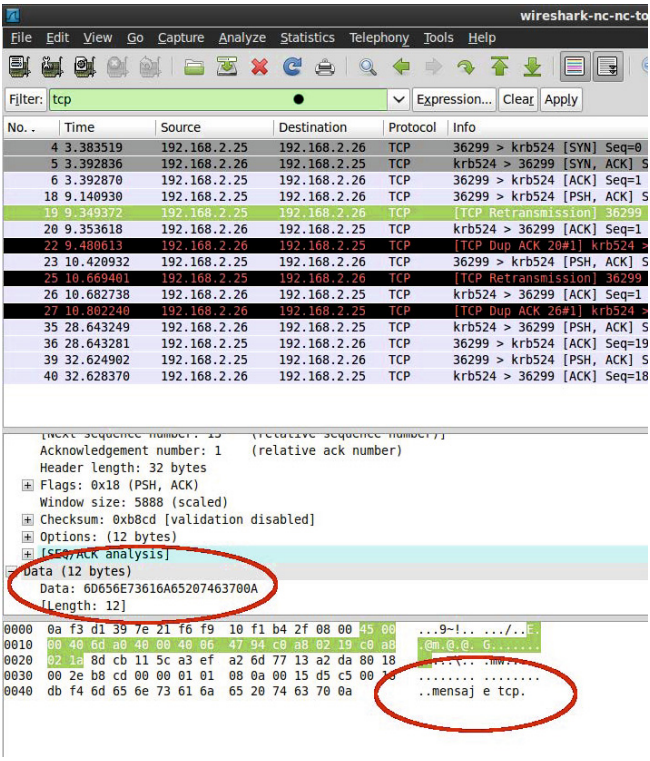


Fig. 6. Checking the pass of messages

2. A similar situation happened when instead of inserting a code it was deleted. With a test we detected that if both users deleted the same section before propagating the changes there were incoherences in the messages. The solution for this was to implement a list of sent changes in each shared point.

4 Conclusions and Future Work

In this paper we have described our experience while developing an online colaborative editor by using formal methods. We have presented our specification formalism and show how one of the main modules of the system was specified.

We have also introduced some of the tests that were automatically derived form the specification. In particular, we showed two tests that found errors in the implementation.

As future work, we are going to introduce new features in our colaborative editor such as online image editor, or video streaming online editor.

References

1. Abreu, R., Zoeteweij, P., van Gemund, A.J.C.: Simultaneous debugging of software faults. Journal of Systems and Software 84(4), 573–586 (2011)
2. Benharref, A., Dssouli, R., Serhani, M.A., En-Nouaary, A., Glitho, R.: New Approach for EFSM-Based Passive Testing of Web Services. In: Petrenko, A., Veanes, M., Tretmans, J., Grieskamp, W. (eds.) TestCom/FATES 2007. LNCS, vol. 4581, pp. 13–27. Springer, Heidelberg (2007)
3. Bowen, J.P., Hinchey, M.G.: Ten commandments of formal methods... Ten years later. Computer 39(1), 40–48 (2006)
4. Calvo, R.A., O'Rourke, S.T., Jones, J., Yacef, K., Reimann, P.: Collaborative writing support tools on the cloud. IEEE Transactions on Learning Technologies 4, 88–97 (2011)
5. di Lucca, G.A., Fasolino, A.R.: Testing web-based applications: The state of the art and future trends. Information & Software Technology 48(12), 1172–1186 (2006)
6. Gobby a collaborative text editor, `http://gobby.0x539.de/trac/wiki/WikiStart`
7. Goderbauer, M., Goetz, M., Grosskopf, A., Meyer, A., Weske, M.: Syncro - Concurrent Editing Library for Google Wave. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (eds.) ICWE 2010. LNCS, vol. 6189, pp. 510–513. Springer, Heidelberg (2010)
8. Gogolla, M.: Benefits and Problems of Formal Methods. In: Llamosí, A., Strohmeier, A. (eds.) Ada-Europe 2004. LNCS, vol. 3063, pp. 1–15. Springer, Heidelberg (2004)
9. Herrick, D.R.: Google this!: using google apps for collaboration and productivity. In: 37th Annual ACM SIGUCCS Fall Conference, SIGUCCS 2009, pp. 55–64. ACM (2009)
10. Hierons, R.M., Bogdanov, K., Bowen, J.P., Cleaveland, R., Derrick, J., Dick, J., Gheorghe, M., Harman, M., Kapoor, K., Krause, P., Luettgen, G., Simons, A.J.H., Vilkomir, S., Woodward, M.R., Zedan, H.: Using formal methods to support testing. ACM Computing Surveys 41(2) (2009)
11. Hinchey, M.G.: Confessions of a formal methodist. In: 7th Australian Workshop on Safety-Critical Systems and Software, SCS 2002, pp. 17–20. Australian Computer Society (2002)
12. Imine, A.: Coordination Model for Real-Time Collaborative Editors. In: Field, J., Vasconcelos, V.T. (eds.) COORDINATION 2009. LNCS, vol. 5521, pp. 225–246. Springer, Heidelberg (2009)
13. Lavinia Ignat, C., Norrie, M.C.: Customizable collaborative editor relying on treeopt algorithm. In: Proc. of the European Conf. of Computer-supported Cooperative Work, pp. 315–334. Kluwer Academic Publishers (2003)
14. Merayo, M.G., Núñez, M., Rodríguez, I.: Extending EFSMs to Specify and Test Timed Systems with Action Durations and Timeouts. In: Najm, E., Pradat-Peyre, J.-F., Donzeau-Gouge, V.V. (eds.) FORTE 2006. LNCS, vol. 4229, pp. 372–387. Springer, Heidelberg (2006)
15. Núñez, M., Rodríguez, I.: Conformance Testing Relations for Timed Systems. In: Grieskamp, W., Weise, C. (eds.) FATES 2005. LNCS, vol. 3997, pp. 103–117. Springer, Heidelberg (2006)
16. Rosenblum, D.S.: Formal methods and testing: why the state-of-the art is not the state-of-the practice. ACM SIGSOFT Software Engineering Notes 21(4), 64–66 (1996)

Construction of Semantic User Profile for Personalized Web Search

Mohammed Nazim Uddin, Trong Hai Duong, Visal Sean, and Geun-Sik Jo

School of Computer and Information Engineering, Inha University, Korea
tonazim@yahoo.com, haiduongtrong@gmail.com,
seanvisal@eslab.inha.ac.kr, gsjo@inha.ac.kr

Abstract. User profile is an essential component for accessing the personalized information from the Web. Efficiency of personalized accessed information highly depends on how to model the user details to construct user profile. Previously, user profile was constructed by collecting list of keywords to inferring user interests. These kinds of approaches are not sufficient for many applications. In this paper, we have proposed a new method for constructing semantic user profile for personalized information access. User's query is extended using ontological profile for generation of personalized search context. Experimental results show that our method of constructing semantic profile is effective for searching information with individual needs.

Keywords: User profile, Ontology, Semantic search, Personalization.

1 Introduction

Personalization is an important method of accessing web information related to user's intentions. Main component of personalization is modeling the user information which is also called the user profile. Due to rapidly increase of Web information it is very challenging to get right information to fulfill user intentions of search. Personalized information search with an accurate and efficient user profile can alleviate the information overload by providing the necessary information of user's needs. Construction of an accurate and efficient user profile is still an open challenge in personalized information retrieval. Effectiveness of personalization information search highly depends upon modeling the user's details. Thanks to Semantic Web technologies for providing an efficient way to describe the real world entities in more meaningful manner to human and computer. User's details can be represented in structural way to describe the conceptual semantic of user interest.

Several research efforts have been made to construct user profile for personalized information search. Survey of numerous techniques and methods for constructing user profile are outlined in [4]. Ontology based user profile based on a domain is constructed and learned with interest score for personalized Web search is described in [7]. A spreading activation algorithm is used to maintain update scores for re-rank the search

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 99–108, 2012.

results. Representation of objects by order partition is described in [10] which can be applied for representing concepts and relations in ontology for building profile considering user's dynamic complex information. Semantic user profile has been utilized for contextual search in [1], [6]. Some methods and applications for creating an ontological user profiles are described in [2]. Ontology for profile was created manually considering important general concepts of a user and then extended it to specific preferences. Ontological profile can be constructed automatically for various domains with the reference of pre-existing hierarchy.

2 Semantic User Profile for Personalization

2.1 Semantic User Profile

User profile basically contains the details information of a particular user such as name, address, phone number, etc, and also includes complex information such as individual preferences. User preferences are utilized to provide personalized information from the Web such as recommendation of items in commerce, filtering and rating systems for email and online newspapers etc. Previously, user profile was undertaken by collection of keywords to represent the user's preferences. In real world, user preferences cannot be represented by a list of keywords. Simple list of keywords generate lots of ambiguity while search the information in Web. Semantic Web techniques can be applied to construct the user profile which can provides semantics (meanings) of user preferences in conceptual manner. Unlike simple keywords semantic profile contains the concepts in structural way to represent the user's preferences or interests.

2.2 Ontological Approach for Semantic Profile

Ontology is the main component of Semantic Web technology. An ontology is defined as a formal, explicit specification of a shared conceptual understanding of a domain [9]. Ontologies are applied by number of researchers in various application for knowledge representation such as [11] and [12]. User details information can be modeled with ontological approach in hierarchical manner to construct a user profile for particular user.

A standard new ontology can be designed by ontology design engineer to model the user information. Classes and relations are defined based on a particular application and may extend through inheritance if necessary to describe the user details. On the other hand, existing domain ontology can be used as reference ontology to model the user detail information. The main benefit of using reference ontology is that there is no burden of designing new schema for a particular application, only need to create the instances of existing schema.

3 Construction of Semantic User Profile for Personalized Information Access

3.1 Personal Information Collection

User details can be collected explicitly by asking to user directly or implicitly through agent by monitoring user activities. Both the collection methods have their own limitations. In explicit method, users directly entered information which may lead to inconsistency or incomplete. Moreover, sometimes users do not want to give some information merely because they are not willing to provide the information. In implicit method, a software program should run in every client machine to monitor the user behavior. Cookies based information collection causes the profile inconsistency if user uses more than one computer (office and home) for browsing information. The main goal of building user profile is to identify the user interest to enhance the Web search information. So, information collection mainly focused on user's areas of interest.

In our approach, we use the combined technique to collect the information with minimum user intervention. User can provide their name and email address to enter the system and system automatically crawl the related information from the Web to build individual profile. The information collected from the Web may be raucous and with different formats. So, information should be preprocessed to construct the profile. Pre-processing basically includes remove HTML tags, remove stop words, and perform word stemming.

3.2 Document Representation

In information retrieval, the most common method of representing the documents is vector space model [3]. In vector space model, documents are represented as vectors of words or terms with weights. There are several methods of determining the weight of terms in documents. The most common method is tf/idf (term frequency and inverse document frequency). For each document *d* in a document collection *D*, a weighted vector is constructed as:

$$\vec{d} = (w_1, w_2 .... w_n) \qquad (1)$$

Where, w_i is the weight of term *i* in document d. Weights (w_i) are calculated as:

$$w_i = f_i * \log(N / n_i) \qquad (2)$$

Where, f_i is the frequency of terms *i* in the document *d*, *N* is the number of documents in collection *D* and n_i is the number of documents that contains term *i*.

3.3 Semantic Profile Construction

Reference Ontology. We have investigated domain ontology as reference ontology to model the use details. We use *ODP* (Open Directory Project) as reference ontology where topics are organized in hierarchical manner along with Web pages belongs to the

related topics. Topics are referred as concepts which are associated with related documents. Documents are nothing but related to Web pages representing the concepts. The textual information that can get extracting from the Web pages explains the semantics of concepts and is investigated to build the term vector for the concepts. Finally, a feature vector is constructed for every concept exists in the reference ontology.

Concept Feature Vector Generation. A Feature Vector is defined as weighted terms (index words) learned from a document or a set of document belonging to a particular concept. For example, Feature vector of concept /computers/ *internet = {web, 356.0; server, 273.0; data, 244.2;.. etc}.* For instance, term *web* has a representativeness of *365.0* to the concept *internet.* Length of feature vector of a concept is maintained with a given threshold.

There are two types of concepts in reference ontology, one is leaf concept and other is non-leaf concept. Leaf concepts are those which have no sub-concepts. For each leaf concept, the feature vector is calculated as an average vectors on the documents vectors that have already been assigned to related concept. Let F^c be the feature vector of concept C and let N be the number of documents assigned to the concept C. Then weight of each feature i of the concept vector is calculated as:

$$F_i^c = \frac{\sum_{N_j \in c} w_{ij}}{|N_j|} \tag{3}$$

For non-leaf concepts, the feature vectors are calculated by considering their direct sub concepts and concepts with subsumed relation. Let N_j be the number of documents belongs to concept C, N_k be the number documents under C's direct sub concepts and N_l be the other related concepts. The i^{th} feature of F^C is measured as:

$$F_i^C = \alpha * \frac{\sum_{N_j \in C} w_{ij}}{N_j} + \beta * \frac{\sum_{N_k \in C} w_{ik}}{N_k} + \gamma * \frac{\sum_{Nl \in C} w_{il}}{N_l} \tag{4}$$

Where α, β and γ are tuning parameters to adjust the contributions from the concept instances, sub concepts and related concepts respectively and $\alpha + \beta + \gamma = 1$.

Classification. Semantic user profile is constructed by classifying user's personal information to reference ontology. First, feature vectors are constructed for every concept and sub concepts in reference ontology (ODP). Then, feature vectors for every document are constructed from the personal information collection of a particular user. To classify user's information to the reference ontology, we calculate a similarity value for each user document and concepts of reference ontology. The similarity between user document and concept of reference ontology are directly calculated by the cosine measured of their representative feature vectors. Let feature vector of user document is a and feature vector of concept in reference ontology is b respectively, with same length n. Then the cosine similarity between a and b is measured by Equation (5).

$$sim(a,b) = sim(D^a, C^b) = \frac{\overrightarrow{D^a} * \overrightarrow{C^b}}{|D^a| * |C^b|} = \frac{\sum_{i=1}^{n} (D_i^a * C_i^b)}{\sqrt{\sum_{i=1}^{n} (D_i^a)^2} * \sqrt{\sum_{i=1}^{n} (C_i^b)^2}} \quad (5)$$

Where, D^a and C^b are feature vectors for document *a* and concept *b*, respectively. *n* is the length of feature vectors. And, $|D^a|$ and $|C^b|$ are lengths of the two vectors, respectively. While measuring similarities among user's documents and concepts in reference ontology a threshold value is maintained, similarity value fall under the threshold are discarded. After classify all documents a new ontology is obtained as shown in Figure 1and it is called semantic user profile. Finally, semantic user profile is an ontology includes user's interests in a hierarchical structure. Ontological user profile for a particular user is depicted in Figure 1. Concepts in profile hold three attributes such as interest scores, feature vectors and related documents respectively.

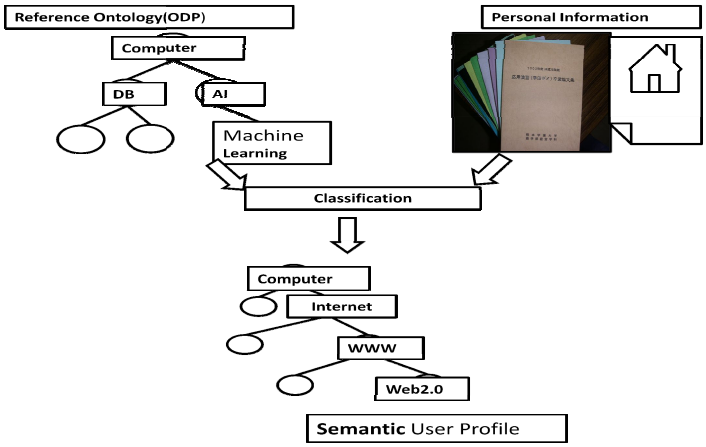


Fig. 1. Ontological user profile

3.4 Personalized Information Access

The improvement of personalized information access depends upon processing the query using user details. Query processing is a method for improving the initial query formulation using the information that is related to the query intent. The process of query processing accomplish by explicit feedback, where the users explicitly provide information on relevant documents to a query for reformulation and implicit feedback, in which the information for query expansion is implicitly derived by the system.

A traditional query consists of one or more keywords for searching the information on the Web. Normal keywords do not provide any semantic for search the information in traditional search engine like Google or yahoo. Hidden concepts behind these

keywords can be explored implicitly with the help information from external sources such as thesaurus or from term relations extracted from the document collection relevant to the given query or using some background knowledge to identify the user interest. In this research, a query is implicitly expanded each time using ontological user profile for generating personalized semantic search context. These semantic contexts are utilized while searching the information.

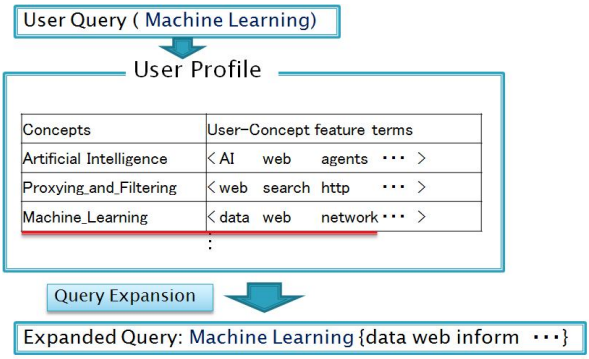


Fig. 2. Query Expansion Process

A query generator takes the search keyword(s) and expands it with concepts and relations explored by the user profile ontology. Figure 2 describes the query expansion process for a query "Machine Learning".

4 Evaluations

In order to construct the ontological user profile for personalized search services the ODP concept hierarchy has been investigated. The RFD representation of ODP is downloaded for the website (http://www.dmoz.org/). Only top "Computer" concept of ODP is considered as a root concept for the experimental purpose and used a branching factor of ten with a depth of six levels of root concept. The main goal of using ODP in this experiment is to construct a reference ontology which is learned with the users' details to build personal profile. The model proposed in this research is targeted to a specific domain of scientific research in computer science field that the top computer concept of ODP has been chosen for the experiment. However, ODP concepts include many sub concepts and documents which are not related to this research domain. Only suitable concepts, sub concepts and documents are included after filtering the RDF version of ODP. Finally, experimental data sets contained 650 concepts in the hierarchy and 15,326 documents that were indexed under various concepts. The indexed documents were pre-processed as described in Section 3 and built reference ontology accordingly.

4.1 Evaluation Metrics

Two widely used statistical methods in information retrieval; Recall and Precision are used to evaluate the accuracy of user profile. The Precision measures the probability that the system mapped the user details to the reference ontology to build profile will be relevant to users whereas recall measures the probability that the classifier will select entire set of relevant documents. Precision and Recall can be defined as:

$$Precision = \frac{\#\ of\ relevant\ documents\ mapped}{\#\ of\ documents\ mapped} \tag{6}$$

$$Recall = \frac{\#\ relevant\ documents\ mapped}{Total\ \#\ of\ relevant\ documents} \tag{7}$$

Finally, the F-measure is calculated by combining precision and recall as follows.

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

4.2 Profile Accuracy

The goal of the experiment of user profile accuracy is to demonstrate that constructed ontological user profile represents user interests and preferences accurately. To construct the ontological user profile, fifty users' details are collected from Google scholar and social network site Facebook by querying their name and e-mail address. A super document is created by combing the collections from these two sites for each user. After pre-processing the documents each user details are mapped with reference ontology to construct the ontological user profile which represents the interests and preferences of a particular user. Mapping between reference ontology and user's details are accomplished by measuring similarities among corresponding feature vectors of respective concepts and document. Ontological user profile contains the weighted concept hierarchy to represents the user interests. Concepts' weights are calculated by summation of its feature vectors along with similarity scores measured by mapping process. Five graduate students were assigned for manual judgments of profiles were relevant or not to the users with necessary information of whose profile have been constructed. According to their opinion precision and recalled were measured. Comparison of precision and recall are presented in Figure 3.

4.3 Query Expansion

The aim of this experiment is to evaluate expanded query concept using ontological profile are relevant to user context or not. For conducting test, twenty users are divided into five groups consisting of four users in each group. Ten queries are submitted by the user to expend using profile to test the context. Figure 4 shows the evaluation results by different groups of users with different queries.

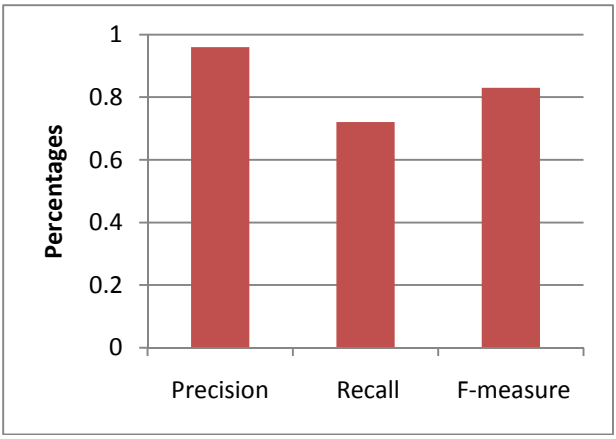


Fig. 3. Profile Accuracy

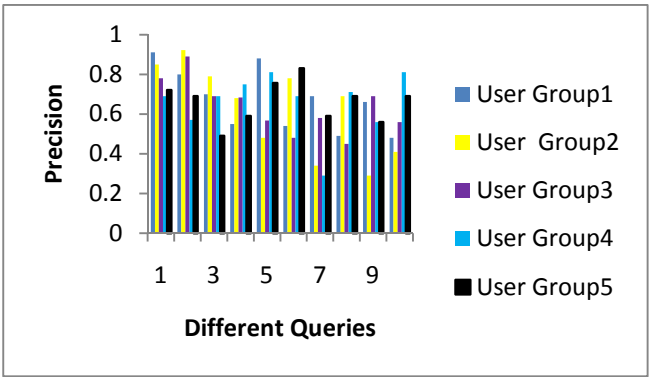


Fig. 4. Query Expansion with different Users

4.4 Personalized Search Results

To evaluate the search results we have investigated the method of pooled relevance judgments with the human judgment. Initially, for a given query top 50 results are collected from Google. Then same query is extended based on semantic profile constructed for 10 users and send to Google for collecting results. All the results were given to some researchers including research faculty members, post doctoral researchers, doctoral and master students to assess the search results. To help the researchers in evaluation process we have provided the necessary information about users whose profile was constructed. According to the assessment personalized search results returned based on semantic profile out performed over non personalized results. For comparing the results of our proposed method we set a baseline by analyzing [1], [6] and [8]. Figure 5 shows the evaluation of Precision at top *N* returned results. Moreover,

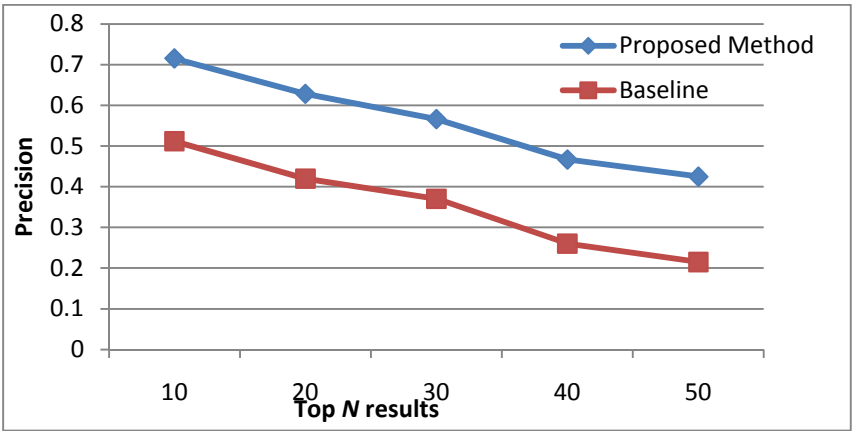


Fig. 5. Personalized search results

based on their analysis the personalized results are closely related to user's background information. To sum up, Precision results on Figure 5 shows that our proposed semantic user profile acceptably improved the personalized search results over baseline method.

In order to search and ranking information, matching of query concept to the Web resources is accomplished. However, most of the exiting methods of searching and ranking information based on Ontology are utilized information filtering approach. The baseline method in the area of searching and re-ranking information using ontological profile [8] based on filtering the information matching with Ontology concept. Additionally, collecting information for building profile [1] and [8] are based on usage histories which contain lot of unnecessary information which do not provides users interests and preferences accurately. However, information collection in our approach is based on social web which specify user interest and preference more accurately.

5 Conclusion

In this paper, we have proposed a model for constructing semantic user profile. Semantic user profile is constructed in semi-automatic approach with minimum user intervention. User's background information is learned with domain knowledge in the form of ontology. Finally, the model represents user interest in hierarchical approach which can be utilized for personal information search. Case studies and evaluations show that our model considerably improved the personalized information search. Nevertheless, there still has room in various dimensions to improve the model. The main backbone structure of our profile is based on ODP. There are many irrelevant concepts and documents (Web pages) exist in ODP. For constructing semantic profile more meaningfully ODP concepts and documents should be filtered effectively.

Acknowledgment. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2011-0015484).

References

1. Mohammed, N.U., Duong, T.H., Jo, G.S.: Contextual Information Search Based on Ontological User Profile. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010, Part II. LNCS (LNAI), vol. 6422, pp. 490–500. Springer, Heidelberg (2010)
2. Golemati, M., Katifori, A., Vassilakis, C., Lepouras, G., Halatsis, C.: Creating an Ontology for the User Profile: Method and Applications. In: RCIS, Morocco (2007)
3. Salton, G., McGill, M.J.: An Introduction to Modern Information Retrieval. McGraw-Hill (1983)
4. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
5. Tang, J., Yao, L., Zhang, D., Zhang, J.: A combination Approach to Web User Profiling. ACM Transaction on Knowledge Discovery from Data V(N), 1–38 (2010)
6. Duong, T.H., Uddin, M.N., Li, D., Jo, G.S.: A Collaborative Ontology-Based User Profiles System. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 540–552. Springer, Heidelberg (2009)
7. Mylonas, P., Vallet, D., Castells, P., Fernandez, M., Avrithis, Y.: Personalized information retrieval based on context and ontological knowledge. The Knowledge Engineering Review, 1–24 (2004)
8. Sieg, A., Mobasher, B., Burke, R.: Learning Ontology-Based User Profiles: A semantic Approach to Personalized Web Search. IEEE Intelligent Informatics Bulletin 8(1) (November 2007)
9. Gruber, T.: A translation Approach to portable Ontology specifications. Knowledge Acquisition 5, 199–220 (1993)
10. Danilowicz, C., Nguyen, N.T.: Consensus-based partitions in the space of ordered partitions. Pattern Recognition 21(3), 269–273 (1988)
11. Nguyen, N.T.: Inconsistency of Knowledge and Collective Intelligence. Cybernetics and Systems 39(6), 542–562 (2008)
12. Nguyen, N.T.: A Method for Ontology Conflict Resolution and Integration on Relation Level. Cybernetics and Systems 38(8) (2007)

Link Prediction in Dynamic Networks of Services Emerging during Deployment and Execution of Web Services

Adam Grzech, Krzysztof Juszczyszyn, Paweł Stelmach, and Łukasz Falas

Institute of Computer Science, Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{adam.grzech,krzysztof.juszczyszyn,pawel.stelmach,
lukasz.falas}@pwr.wroc.pl

Abstract. We propose an approach, according to which the Web services interoperability and resulting composition schemes may be used to create the network structures reflecting the patterns according to which the services interact during execution of composition and execution queries. We show how to create so-called networks of Web services which allow to effectively use the network structural analysis and optimization techniques to solve the network composition problems. The service network is created on the basis of the semantic bindings between the services in the repository joined with the actual patterns of the service usage resulting from composition queries. Next we show how available techniques of dynamic network structure prediction and analysis may help to assess the future service usage and resource consumption of the service execution layer. Our approach is illustrated by the real data gathered from the *PlaTel* platform, dedicated to the complex service planning, management, provision, composition, execution and validation.

1 Introduction

The rapid development of contemporary service systems, built in accordance with the SOA (Service-Oriented Architectures) paradigm triggers the development of various methods and algorithms devoted to the analysis of user activity, service usage and overall description of complex service systems [9][10]. Among them the first approach to graph based description of service repositories was proposed in [8]. In this work we extend this approach by demonstrating the application of graph structural analysis [5] to the networks of services and introducing a model of dynamic network of services. This model can be used to build effective tools to manage access to system services, infrastructure management, system load forecasting and evaluation of quality of service.

This allows us to apply and evaluate the existing link prediction methods to the evolving networks of services. A broad survey of link prediction methods is presented

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 109–120, 2012.

in [6]. It should be noted that most methods of the link prediction give rather poor results – the best predictors discussed in [2] can identify < 10% of emerging links.

Basing on our previous experience, which shows that the distribution of sub-graphs in complex networks is statistically stable and typical for the considered network [4], we claim that it is possible to characterize the network structural changes by statistical data about the evolution of its sub-graphs and show that this approach leads to especially good results in the case of service networks.

The proposed and discussed approach is illustrated by quantitative analysis based by the real data gathered from the *PlaTel* platform, dedicated to the complex service planning, management, provision, composition, execution and validation.

In the following sections we propose a method for the description of service repositories within a graph based model, then we show an example of structural analysis of such networks, which allows to infer the roles, importance and possible risk level of the services. We also present an example of dynamic network of Web services and propose an application of link prediction methods to infer the future service usage and evolution of service networks.

2 Networks of Web Services

A Web service s_i typically has two sets of parameters: $in(s_i)$ for SOAP request (as input) and $out(s_i)$ for SOAP response (as output). When s_i is invoked with all input parameters $in(s_i)$, it returns the output parameters, $out(s_i)$. We assume that in order to invoke s_i, all input parameters in $in(s_i)$ must be provided ($in(s_i)$ is mandatory). In the case of composite services the input parameters for each of its atomic services are provided from two sources. First is the user input (which takes place when the user invokes a composite service, providing initial parameters) and we assume that these parameters are complete and adequately described. The second are the outputs of other atomic services taking part in the same execution plan of the composite service. This requires semantic compatibility between inputs and outputs of atomic services.

The information contained in the SSDL descriptions of Web services is sufficient to create the Network of Services (*NoS*) - a graph model representing all the semantic bindings between services within a given repository. The same concerns the standard approach – WSDL language [8].

The Network of Services is a tuple: $NoS = (S, E)$ where:

- $S = \{s_1, s_2, \dots, s_n\}$ is a set of services (stored and described in a repository),
- $E = \{e_{ij} = e(s_i, s_j) | i, j \in \{1,2, \dots, n\}, i \neq j\}$ is a set of directed edges (relations) between the services from S, where:

$$e_{ij} = e(s_i, s_j) \in E \quad iff \quad in(s_j) \subseteq out(s_i)$$

The existence of a directed edge $e_{ij} = e(s_i, s_j)$ may be interpreted as a fact that the execution of s_i provides a full set of input parameters needed to invoke s_j.

Having defined the *NoS* we may propose a general approach to the characterization of dynamic patterns of interaction between the Web services, which result from service composition and execution. Note that, the *NoS* represents all possible parameter transfers between semantically compatible services in a repository. As the composite services are being composed and executed, only a subset of them may be observed in a system within a given timeframe.

If we decide to observe all the parameter transfers between services in a time window of arbitrary length we may use the *NoS* approach and define the networks representing the services (from given set of services) activity within this time window. Itn leads to the definition of the Dynamic Network of Services (*DNoS*):

$$DNoS_k = f(\{NoS(1), NoS(2), \dots, NoS(k)\})$$

where $NoS(l) = (S, E_l)$, and $e_{ij} = e(s_i, s_j) \in E_l$ $(l = 1,2, \dots, k, \dots)$ iff s_i was executed and provided input parameters for s_j during time window number l with equal or different lengths t_l and $f(\{\dots\})$ is a transformation according which the collected patterns are processed in gain to predict future services patterns.

Dynamic Network of Services obtained after k time windows is denoted by $DNoS_{ks}(S, E(k,s))$, where $E(k,s)$ is a transform of all or selected subsets of edges E_l $(l = 1,2, \dots, k)$, i.e., $E(k,s) = g(E_{k-s}, E_{k-s-1}, \dots, E_k)$ and s $(0 \le s \le k-1)$ is a number of time windows took into account. In the simplest cases the $g(\dots)$ may be sum of all or selected edges patterns E_l $(l = 1,2, \dots, k)$, i.e., $E(k,s) = E_{k-s} \cup E_{k-s-1} \cup \dots \cup E_k$ ($E(k,s)$ is a binary matrix) or may be weighted sum of all or selected edges patterns E_l $(l = 1,2, \dots, k)$ ($(E(k,s)$ is a real numbers matrix).

If the $NoS(l) = (S, E_l)$ and – consequently - $DNoS_{ks}(S, E(k,s))$ are known for assumed $f(\{\dots\})$ and $g(\dots)$ transformations, well-known prediction methods may be applied among other in gain to:

- formulate access control strategies for services requests,
- negotiate Service Level Agreement (SLA) parameters' values,
- evaluate and establish billing strategies,
- split and/or merge services available in the services repositories,
- resize virtualized services execution environment,
- accommodate resources provisioning strategies to changes in users requirements,
- personalise requests processing,
- forecast quality of the delivered services in given execution environment.

User queries trigger composition of complex services, which are then executed by the service engine. Thus, *DNoS* stores information about parameter interchange in a service system, which is time-dependent and has a graph representation.

We may notice that this approach is analogous to the representation of dynamic social networks, where the interactions between humans are stored as graphs

and analysed on time-window basis [3][7]. In this case we may describe the *DNoS* model as a social network of Web services. We utilise this analogy to propose a Web service usage analysis methodology, which assumes the following steps:

1. For given service repository and associated service composition framework create the NoS model representing semantic services' compatibility.
2. Apply structural graph analysis methods to infer the properties of services.
3. Build $DNoS_k$ – a series of networks of services representing the previous and actual composite services usage.
4. Use link prediction methods to infer the future composite service usage and parameter flows.
5. Relate predictions to measurable consumption of system resources.

In order to illustrate the above concepts, in the next section we present the first experimental results obtained with the *PlaTel* (Platform for ICT solutions planning and monitoring) service management framework.

3 PlaTel Framework and the Experimental Setup

The illustrative example of the creation and analysis of the *NoS* model will be presented on the basis of repository of services belonging to the *PlaTel* framework, supporting business processes in distributed ICT (Information and Communication Technologies) environment based on Service Oriented Architecture (*SOA*) paradigm [11]. The framework scope of functionalities is divided into applications that cover the whole life cycle of business oriented ICT applications.

The *PlaTel* approach to service description problem assumes the use of the native service description language, SSDL (Smart Service Description Language) which is proposed as a solution allowing simple description of composite service execution schemes, supporting functional and non-functional description of services. Its functionality includes that of the Web Service Description Language (WSDL), but offers important extensions. SSDL is dedicated for service execution support, including the guarantees of service QoS parameters and dynamic service composition at runtime.

The main difference between WSDL and SSDL is not only the range of description (atomic service vs. composite service) but also the fact that WSDL is a complete instruction on how to call a Web Service via http (namely using SOAP protocol), while SSDL is an execution plan needing an engine to execute (interpret) it.

A definition of SSDL node types contains all basic data types which allow for the functional and non-functional description of a service, its execution requirements and the description of complex services with conditional execution of their atomic components.

Each of them is associated with the number of sub-nodes allowing for precise description of a service. An important part of the functional description of a SSDL node is class attribute, which contains semantic labels describing the input parameters of

the service. The labels are taken from domain ontology and used during service composition and the construction of data flow inside a composite service. Thus, the service repositories in *PlaTel* store all information needed to create the *NoS* and *DNoS* models.

So far, we can find many approaches to the service composition problem [18]. In work [14] it is mentioned that, rather than starting with a complete business process definition, the composition system could start with a basic set of requirements and in the first step build the whole process, whereas many approaches [19] require a well-defined business process to compose a complex service. Current work often raises the topic of business process analysis, semantic analysis of user requirements, service discovery (meeting the functional requirements), selection of specific services against non-functional requirements (i.e. execution time, cost, security) and computational and communication resources provisioning. However, the presented solutions have some disadvantages, i.e. these methods have not yet been successfully combined to jointly and comprehensively solve the problem of composition of complex services that satisfy both functional and non-functional requirements. In many cases only one aspect is considered. For example the work [21] focuses on services selection based only on one functional requirement at a time. Other works [17][20][22] show that non-functional requirements are considered to be of a key importance, however many approaches ignore the aspect of building a proper structure of a complex service which is key to optimization of i.e. execution time.

To this date researchers have approached the service composition from different perspectives. Some have presented specialized methods for services selection or composite service QoS-based optimization [15]. However, despite the importance of their contribution, those solutions are not widely used by other researchers. Some propose complete end-to-end composition tools introducing a concept of two-staged composition: [13] logical composition stage to prune the set of candidate services and then composing an abstract workflow. METEOR-S [14] presents a likewise concept of binding web services to an abstract process and selecting services fulfilling the QoS requirements. Notions of building complete composition frameworks are also clear in SWORD, [16] which was one of the initial attempts to use planning to compose web services. However, the proposed approaches are closed and do not support implementation of other methods and, because of this, it is difficult to call them frameworks. And a framework-based approach is what is currently needed in SOA field in order to create composition approaches that are fitted to different domains characteristic for them.

Our framework provides multi-criteria (functional and non-functional) composition of Web services and QoS assurance for utilized ICT services that allows for building of various service selection and composition scenarios. It also utilizes configurable composite services to provide its functionality. In order to ensure the interoperability with external service and ontology repositories we have created specialized services – Mediators – responsible for database access and providing

standardized interfaces for internal *PlaTel* services. In result we can integrate external resources within our framework (which was equipped with access control functions). All these features taken together offer flexible and extensible environment for communication and composition of the Web services. To our best knowledge it is the first framework for service composition that implements this kind of flexible approach.

- All the key components of the *PlaTe*l (namely: service composer, communication mapper and execution engine) are composite services themselves, which implies that their execution schemes may be easily reconfigured or extended – and a specialized graphical user interface is provided to support such actions. This approach allows creating repositories of dedicated composite services for composition, mapping and retrieval of the Web services, which may be used according to current needs. They may be stored and conditionally chosen from the repository.
- We address the non-functional requirements of Web services and propose an extensible description language (SSDL – Smart Service Description Language), which allows expressing non-functional parameters of a composite service.
- The execution engine interpreting the SSDL definitions of composite services and maintaining the non-functional parameters. It has two modes of execution – interpretation (the components of a composite service are explicitly executed) and composite service emulation (in this case the execution engine reads the SSDL and communicates like a service described in the SSDL definition, thus supporting automatic deployment of composite services). The framework also allows the use of external execution engines (in this scenario the *PlaTel* engine serves as an SSDL-driven interface to them).
- We also deal with the issue of communication between Web services by providing communication mapper – a dedicated service which supports service composer and substitutes each node of the composite service scenario graph with a sub-graph representing the order of execution of atomic ICT services (communication and computational) necessary to connect domain atomic services (communication services provide an appropriate medium for transmission of data between computational services responsible for data processing tasks such as: encryption, encoding, signal merging/splitting etc.).

The *PlaTel* framework and its repository of services for the composition of service-based applications for property monitoring domain will serve as a demonstrative example of how the network of services can be created and analyzed.

4 Exemplary Network of Services

For the first experiments on *PleTel* framework, the repository of services used to build applications for monitoring and property security domain was chosen.

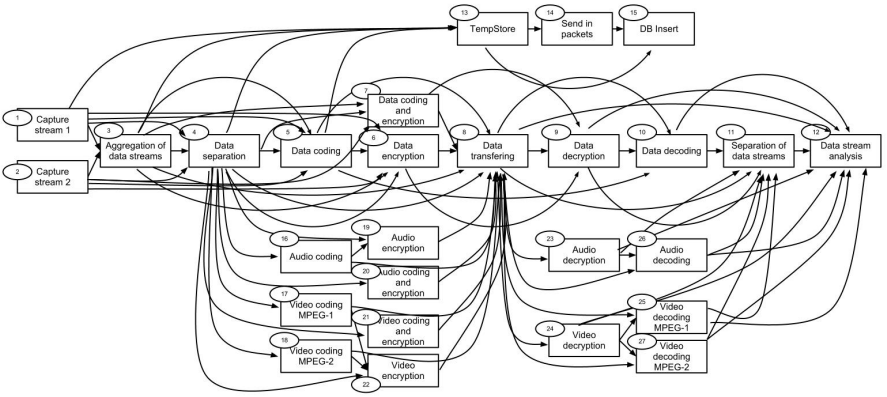


Fig. 1. Network of Services for exemplary service repository

The repository is relatively small and consists of 27 services, for which the *NoS* model was created, according to the definition given in the preceding section. Fig. 1 presents the visualisation of the *NoS*, viewed as a directed graph with labelled nodes representing services. The *NoS* was analysed using standard structural network analysis techniques which returned interesting results.

First of all the node degree distribution was checked – most of the complex networks existing in nature, from social to biological, economic and technology-based show scale free node degree distribution, following the power law [1], the same was confirmed for the *NoS* of *PlaTel* service repository. Fig. 2 presents the node degree distribution for *PlaTel* service repository.

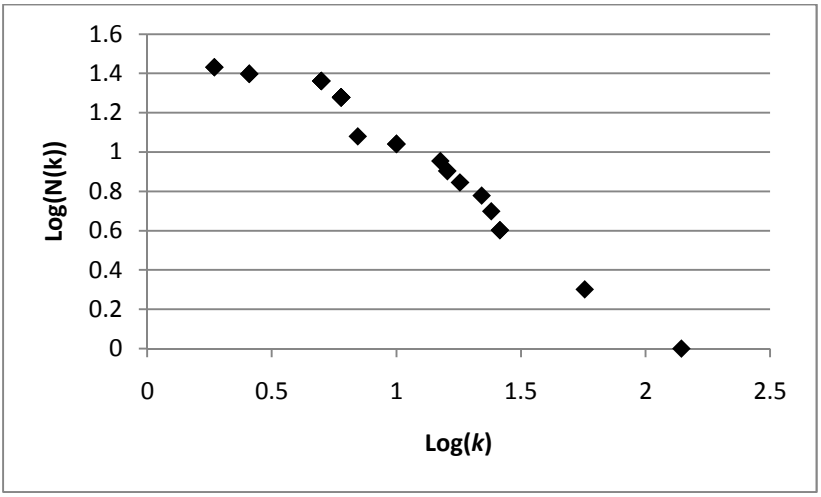


Fig. 2. Node degree (k) distribution for PlaTel service repository

This results confirm observations and conclusions presented in [8] for Web service repositories. The next step was the structural analysis of the network – inferring the node types from their connection patterns, calculating betweenness centralities (which correspond to the relative importance of the node in a graph) and node group analysis.

Fig. 3 presents the *PlaTel NoS* graph created in NetMiner 4.0 network analysis software, with three node groups detected by the standard CNM (Clauset, Newman and Moore) algorithm. We may note that the groups, however detected only on the basis of the graph structure contain the services with corresponding functionalities (coding and encryption – G1, decoding and decryption – G2, storing and stream processing – G3). This may suggest an effective strategy for categorization of Web services in large heterogeneous repositories.

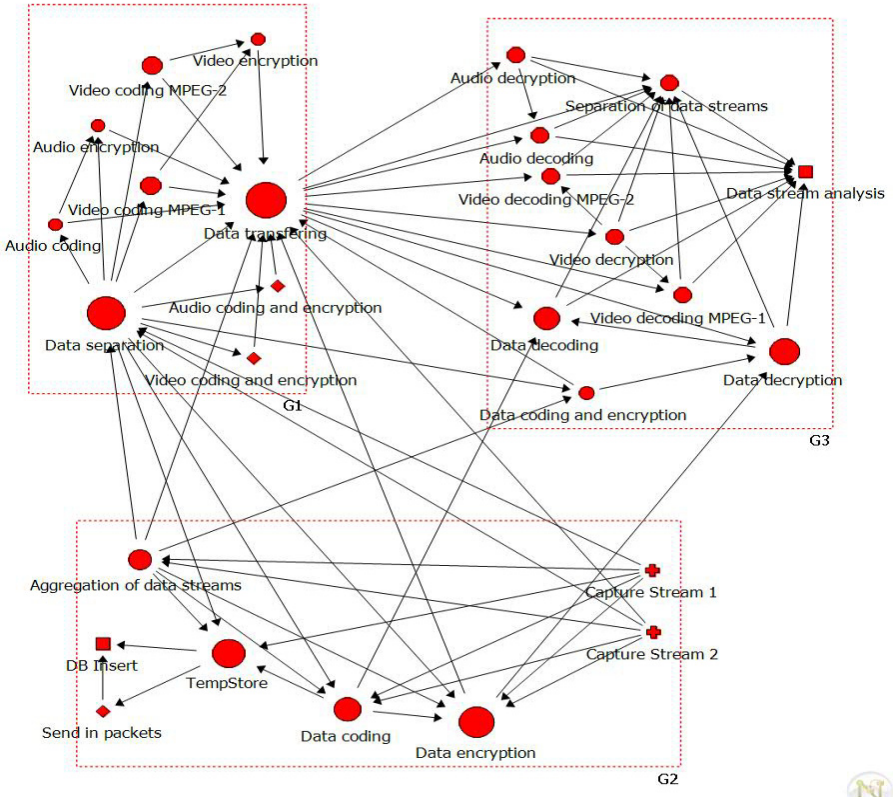


Fig. 3. Network size for time windows of different size for WUT dataset

The node types were detected using approach defined in [5]: *Isolate* nodes do not have any links. *Transmitter* (crosses on Fig. 3) has only out links and no in links. *Receiver* (cross) node has only in links, while *Carrier* node (rhombus) has exactly one incoming and one outgoing link. *Ordinary* node (circle) does not fall in any of the

above categories. The types allow to classify the nodes (services) in the context of their roles and the importance for the functionality of the service repository.

The detailed results concerning network nodes of the investigated *NoS* are presented in Table 1. The services *Data transferring* and *Data separation* were assigned the highest betweenness centrality (visualized as the node size on Fig. 3) which suggests their key role in the communication between the services in the repository which corresponds to the domain knowledge and typical usage of the *PlaTel* services.

Table 1. The node characteristics for the PlaTel *NoS*

	In Degree	Out Degree	Betweenn. Centrality	Node Type
Capture Stream 1	0	5	0	Transmitter
Capture Stream 2	0	5	0	Transmitter
Aggregation of data streams	2	6	0,006462	Ordinary
Data separation	3	11	0,038769	Ordinary
Data coding	4	3	0,006895	Ordinary
Data encryption	5	2	0,018998	Ordinary
Data coding and encryption	2	2	0,001305	Ordinary
Data transfering	12	8	0,166956	Ordinary
Data decryption	3	3	0,009916	Ordinary
Data decoding	3	2	0,006559	Ordinary
Separation of data streams	8	1	0,001972	Ordinary
Data stream analysis	8	0	0	Receiver
TempStore	4	2	0,015385	Ordinary
Send in packets	1	1	0	Carrier
DB Insert	2	0	0	Receiver
Audio coding	1	2	0	Ordinary
Video coding MPEG-1	1	2	0,003077	Ordinary
Video coding MPEG-2	1	2	0,003077	Ordinary
Audio encryption	2	1	0	Ordinary
Audio coding and encryption	1	1	0	Carrier
Video coding and encryption	1	1	0	Carrier
Video encryption	2	1	0	Ordinary
Audio decryption	1	3	0,001972	Ordinary
Video decryption	1	4	0,001972	Ordinary
Video decoding MPEG-1	2	2	0,001972	Ordinary
Audio decoding	2	2	0,001972	Ordinary
Video decoding MPEG-2	2	2	0,001972	Ordinary

We argue that the results of such analysis may be effectively used to select services which are important for given domain, and the structural *NoS* analysis may contribute to the risk and resource management in the service systems.

In the next section the dynamic properties of the networks of Web services will be analyzed, with special attention to the network link prediction problem and its application to the assessment of the future service and resource consumption.

5 Networks of Web Services and the Link Prediction [Problem

For the experiments with the *DNoS* a record of the actual service usage was needed. The dynamic network representing the actual service usage was created, then the link prediction methods were applied in order to assess the future service usage and the structure of the resulting *DNoS*. The experiments were carried on the PlaTel framework, with the following assumptions:

- 5 users took part in the experiment, and 9 types of queries (requirement graphs, representing the user demands for composite services) were invoked ~200 times.
- Queries were served by the *PlaTel* composer module, with *exact match* semantic filter (assuming exact correspondence between semantic description of requirements and the selected services).

The resulting dataset (from here denoted as *PlaTel* dataset) was divided into 80 time windows, corresponding to the 80 *DNoS*s. First 30 were used to train the link prediction algorithms, the remaining 50 were used for verification.

Prediction evaluation procedure was performed according to the scheme proposed in [2].

The *DNoS*s created were highly dynamic. The number of links emerging and disappearing in the consecutive time windows varied frequently, which was quite different from the situation met in the case of dynamic social networks, where, in most cases, the number of new links in the network may be predicted by means of time series analysis [12].

For the link prediction problem three algorithms were used: Preferential Attachment (*PA*), Common Neighbours (*CN*) and Triad Transition Matrix (*TTM*). First two are standard link predictors which assume the social-driven behaviour of network nodes: PA assumes the tendency of new links to be adjacent to network hubs, CA tries to connect nodes which have numerous common neighbours.

This approaches have strong grounding in social science and were proven to be effective in the case of social networks. TTM is a novel, domain-independent method, first introduced in [12]. It is based on statistical description of changes in elementary network sub-graphs – the triads of nodes. Despite the link prediction problem being hard (prediction accuracy for real-life complex networks are rarely better than 5%) it was shown to be very effective, especially for networks analysed in short time scales [12]. The average prediction accuracy was 1.3% for CA, 2.7% for PA and 18.7% for TTM. The TTM decisively outperforms the other predictors which leads to the conclusion, that the evolution of *DNoS* is not driven by social evolutionary schemes. The relative performance of CA and PA also confirms this conclusion – in most social network datasets CA outperforms PA. This also suggests, that we may expect similar phenomena for the majority of other link predictors available – most are derived from the observations of social phenomena applied to the complex networks.

Good results for TTM imply also that predictors using time series analysis, sub-graph structure mining and network statistics will perform better in the case of dynamic networks of services. We may also note that for some windows all the predictors have zero

accuracy. This is caused by the lack of user activity (queries) during these windows and suggests that a methodology for choosing window timespan is needed.

An important fact is also the significant reduction in the computational cost (for all predictors). This is caused by the reduction of possible link space in contrary to social networks, where one can expect n^2 possible links in a n-node network, in the case of *NoS* the complete link space is equal to the number of its links (note that only some of them occur in the *DNoS*). This, however had no influence on the performance of the predictors.

6 Conclusions and Future Work

The presented approach is quite novel – the only one work suggesting the network approach to the description of service repositories is [8], however only the static approach to the service networks was presented there and no structural analysis or network evolution scenarios have followed. The concepts of *NoS* and *DNoS* open vast possibilities of applying various graph and network analysis techniques for the management and evolution discovery of complex service systems. The most attractive and practically important areas of future research are:

- Utilizing all the information stored in service description records (in WSDL and SSDL alike) for the creation of complex service networks.
- Broad analysis of link prediction methods in order to choose appropriate approaches to dynamic service networks.
- Establishing connections between structure prediction of service networks and resource consumption and allocation in service systems.
- Utilizing information about users (who submit composite service queries) during creation and analysis of service networks' models.

References

1. Barabasi, A.: The origin of bursts and heavy tails in humans dynamics. Nature 435, 207 (2005)
2. Lieben-Novell, D., Kleinberg, J.M.: The link-prediction problem for social networks. JASIST (JASIS) 58(7), 1019–1031 (2007)
3. Braha, D., Bar-Yam, Y.: From Centrality to Temporary Fame: Dynamic Centrality in Complex Networks. Complexity 12(2), 59–63 (2006)
4. Juszczyszyn, K., Musial, K., Kazienko, P., Gabrys, B.: Temporal Changes in Local Topology of an Email-Based Social Network. Computing and Informatics 28(6), 763–779 (2009)
5. Wasserman, K., Faust, L.: Social network analysis: Methods and applications. Cambridge University Press, New York (1994)
6. Getoor, L., Diehl, C.P.: Link mining: a survey. ACM SIGKDD Explorations Newsletter 7, 3–12 (2005)
7. Huang, Z., Lin, J.: The Time-Series Link Prediction Problem with Applications in Communication Surveillance. INFORMS Journal on Computing 21(2), 286–303 (2009)

8. Oh, S., Lee, D., Kumara, S.: Effective Web Service Composition in Diverse and Large-Scale Service Networks. IEEE Transactions on Services Computing 1(1) (2008)
9. Wang, Y., Zhang, J., Vassileva, J.: Effective Web Service Selection via Communities, Formed by Super-Agents. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 549–556 (2010)
10. Stelmach, P., Grzech, A., Juszczyszyn, K.: A Model for Automated Service Composition System in SOA Environment. In: Camarinha-Matos, L.M. (ed.) DoCEIS 2011. IFIP AICT, vol. 349, pp. 75–82. Springer, Heidelberg (2011)
11. Stelmach, P., Juszczyszyn, K., Prusiewicz, A., Świątek, P.: Service Composition in Knowledge-based SOA Systems. New Generation Computing 30(2&3) (2012)
12. Juszczyszyn, K., Musial, K., Budka, M.: Link Prediction Based on Subgraph Evolution in Dynamic Social Networks. In: SocialCom/PASSAT 2011, pp. 27–34 (2011)
13. Agarwal, V., Chafle, G., Dasgupta, K., Karnik, N., Kumar, A., Mittal, S., Srivastava, B.: Synthy: A system for end to end composition of web services. Web Semantics: Science, Services and Agents on the World Wide Web In World Wide Web Conference 2005, Semantic Web Track 3(4), 311–339 (2005)
14. Aggarwal, R., Verma, K., Miller, J., Milnor, W.: Constraint Driven Web Service Composition in METEOR-S. In: Proceedings of the 2004 IEEE International Conference on Services Computing, pp. 23–30 (2004)
15. Jong Myoung, K., Kim, C.O., Kwon, I.H.: Quality-of-service oriented web service composition algorithm and planning architecture. The Journal of Systems and Software 81, 2079–2090 (2008)
16. Ponnekanti, S.R., Fox, A.: SWORD: A developer toolkit for Web service composition. In: Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA (2002)
17. Cena, F., Furnari, R.: Discovering and Exchanging Information about Users in a SOA Environment. Communication of SIWN - Systemics and Informatics World Net 4(3), 34–38 (2008)
18. Rao, J., Su, X.: A Survey of Automated Web Service Composition Methods. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 43–54. Springer, Heidelberg (2005)
19. Jong Myoung, K., Kim, C.O., Kwon, I.H.: Quality-of-service oriented web service composition algorithm and planning architecture. The Journal of Systems and Software 81, 2079–2090 (2008)
20. Karakoc, E., Senkul, P.: Composing semantic Web services under constraints. Expert Systems with Applications 36(8), 11021–11029 (2009)
21. Klusch, M., Fries, B., Sycara, K.: OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services. Web Semantics: Science, Services and Agents on the World Wide Web 7, 121–133 (2009)
22. Zeng, L., Kalagnanam, J.: Quality Driven Web Services Composition. In: 12th International Conference on the World Wide Web, pp. 411–421 (2003)

Towards a Model of Context Awareness Using Web Services

Mahran Al-Zyoud, Imad Salah, and Nadim Obeid

Department of Computer Information Systems,
King Abdullah II School for Information Technology,
The University of Jordan,
obein@ju.edu.jo

Abstract. Recent years have witnessed the movement of many applications from the traditional closed environments into open ones. These applications, which are being accessed via web browsers, usually offer a great amount of information and services. Open environments and content explosion may affect the usability of web applications, where usability measures the degree of usage satisfaction of the application provider and the application user. If both sides of a communication (the web application and the device accessing it) collaborate to manage the various issues of context, usability could be improved. This paper focuses on modeling context awareness. We propose two models that organize knowledge in layers, and complement each other, in order to give the web applications' developers the adequate knowledge and a visualization of what should be performed to develop a context aware application. Some of the major issues that need to be considered are: context representation and the heterogeneity that characterizes the open environment of web applications. We shall employ the object oriented approach to represent context and we shall utilize the web services to make a first step toward developing a notion of universal interoperability that aims to facilitate the communication between the server hosting the web application and the devices used to access it. We aim to enable each device to be responsible for its own context without the need of the web application to know the details of how the device is managing the context. As a case study, we present an implementation of a prototype of a university portal.

Keywords: Context awareness, Context representation, Web services.

1 Introduction

Institutions create web applications that aim to serve as an entrance to their information systems or to announce their existence to the public. These web applications usually include huge amounts of important information and distinguished services that are targeted to users according to their preferences, interests, roles in the business, geographical location, time of access, etc. Open environments and content explosion may affect the usability of web applications, where usability measures the degree of usage satisfaction of the application provider and the application user.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 121–131, 2012.

If both sides of a communication (the web application and the device accessing it) collaborate to manage the various issues of context, usability could be improved.

One way to enhance the usability of web applications is to make them adaptable using context. In this paper, we focus on modeling context awareness. We propose two models that organize knowledge in layers, and complement each other, in order to give the web applications' developers the adequate knowledge and a visualization of what should be performed to develop a context aware application. Some of the major issues that need to be considered are: context representation and the heterogeneity that characterizes the open environment of web applications. We shall employ the object oriented approach to represent Contextual knowledge (e.g., role, time, location, and device) which will be embedded into layered models. We shall utilize the web services to make a first step toward developing a notion of universal interoperability that aims to facilitate the communication between the server hosting the web application and the devices used to access it. We aim to enable each device to be responsible for its own context without the need of the web application to know the details of how the device is managing the context. However, there is a need for a common ground to reliably exchange messages between the web application and the device without error or misunderstanding; i.e. to make them interoperable regardless of hardware architecture or software platform differences.

In Section 2, we give the background and related work that include the notion of context, Context representation, context aware applications and web services. Section 3 will be concerned with context awareness modeling. We present two models: (1) a model of the process performed at the node initiating the communication with the web application and a model of what happens at the web application side. In section 4, we present an implementation of a prototype of a university portal as a case study.

2 Background and Related Work

There are many definitions of context and many dimensions along which representations of contextual knowledge may vary [3]. However, most researchers in the different disciplines seem to agree that (1) a context is partial as it describes only a subset of a more comprehensive state of affairs, (2) it is approximate as contextual information which is not required by the user must be abstracted away and (3) a given state of affairs can be considered from several independent perspectives such as a spatial perspective (e.g., a user's location), temporal perspective and logical perspective.

Across a variety of disciplines, specific formulations of context tend to emphasize different aspects. Dey [5] defines context as: "*Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*".

Context awareness may be defined as the ability of a computer system to know about the surrounding environment of the user (the context); i.e. to be aware of the user's context. There many context aware applications. In [7], a system that is aware

of the patient's condition was created. It is accompanied by a repository containing condition's history of the patient in order to enhance the medical care provided. A kind of applications that serves as a friend finder is described in [9]. It works by taking the user type and location as criteria to suggest and find friends. Facebook now supports this type of service. In [8] a discussion of how the presentation of different context sources would improve the user's interaction with calendar and appointment applications. Other context aware applications could be found in [2] and [1].

The main approaches of context modeling include Key-Value Modeling, Markup Scheme Modeling, Graphical Modeling, Object Oriented Modeling, Logic Based Modeling, and Ontology Based Modeling (cf. [10]). Different approaches have been presented for contextual information acquisition such as direct sensor access, Context server and Middleware infrastructure (cf. [4]). In this paper we adopt the object oriented modeling approach where the concepts of encapsulation and inheritance are used to hide the context processing and to give a hierarchical shape of the system. We make the node, accessing the web application, responsible for its context by allowing it to directly access its own sensors while the web application performs the querying for these contextual aspects collected and controlled by the node itself.

Web Services are loosely coupled software components that can be reached using open protocols such as Hyper Text Transfer Protocol (HTTP) and eXtensible Markup Language (XML). They can be considered as an effort to create a universal interoperability by facilitating the communication between electronic devices on the network. In this regards, they can be considered as a universal client/server architecture that allows systems to communicate with each other without the need to use any proprietary client libraries.

The service interface information is disclosed to the client via a configuration file encoded in a standard format (WSDL) and the UDDI registry is used as a repository of these services information as described in their WSDL documents.

An explanation of the process that must be performed in order to design and build a context-aware application is presented in [6]. The design process consists of:

(1) Specification: aims at specifying the behaviors that need be implemented and the contextual information that must be present at run.
(2) Acquisition: aims at defining the mechanisms needed to provide the context such as installing physical sensors, utilizing virtual sensors, knowing the type of data the sensors provide, knowing how to communicate with theses sensors, communicating with sensors, store context and interpret context.
(3) Provisioning: aims at providing methods to make the context information accessible to applications.
(4) Reception: aims at acquiring context. It may involve locating the needed sensors, figuring out how to communicate with them, interpreting the received context to make it more useful for the application.
(5) Action: aims at adapting the application according to received context by analyzing the context or combining it with other information collected momentarily or from past communication, selecting the most appropriate context-aware behavior and finally performing the behavior.

3 Context Awareness Modeling

Our aim is to enable the web application, which is created cooperatively by the analyst and the programmer, to understand the activities (situations) and understand the preferences of the user when he/she is engaged in various activities. Based on the design process discussed above, we provide the following two models:

(1) A model of the process to be performed at the node initiating the communication with the web application to which we shall refer as Node-Model. This includes the phases of specification, acquisition, and provisioning.
(2) A model of what happens at the web application side to which we shall refer as Web-Model. This includes reception and action phases.

The idea of the two models is due to the need to separate the core application logic from the logic of the various processes that aim to detect, refine, and reason about context. This separation of base behavior of the application from context related processes should improve modularity. In our view, separation means that each node (e.g., communicating party) that accesses the application has to deal with a number of issues such as: (1) what context data to collect via the available mechanisms, (2) what analytical processes must be performed to benefit the purpose of usage at the node, (3) what context information to give to requesting applications and at what level (regardless of the level of context requested) and (4) how to utilize the resources of the node efficiently to the benefit of the context management process and to other routine purposes.

The node that represents the application has to handle a set of issues which may include: (1) what services/information the application provides (the core application logic), (2) how to take the context of usage into consideration while providing these services and (3) in what ways and in what parts of the application the context may play a role in helping the user.

3.1 Node-Model

Figure 1 shows the layers of the Node-Mode.

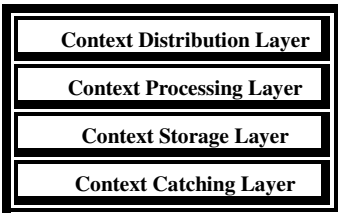


Fig. 1. Node-Model

Context Catching Layer: This is the layer that mostly deals with hardware. Here, hardware devices are heavily employed to collect (acquire) various contextual information of the communicating node's environment. This type of sensing where

hardware devices are used to collect context information is called hard sensing. A quick instance of these devices is sensors. It is also possible to employ software mechanisms to collect data (soft sensing) concerning the user interaction with the device's programs and his/her interaction with the Internet. It is apparent then that context information can be gathered from a variety of sources using different technologies. This should enhance the requirement of making the context loosely-coupled from the sensing mechanisms used.

We propose a Universalization Sub-Layer which transmits (provides) the context data in a universal way. We can connect this layer to the layer above through well-known interfaces. This should enhance the modularity of the architecture by allowing the use of any kind of technology in the next layer as long as it can figure out how to deal with the interfaces of context providers.

Context Storage Layer: This layer represents the usage of a permanent memory mechanism that influences the processing done by the above layer, especially those concerning statistical analysis and historical references where context history may be used to establish trends, predict future context values, and to implement intelligent learning algorithms that help to make applications more able to adapt intelligently; to mention a few examples of context history usage. We consider this layer to be of higher importance and we claim that its introduction in the model is also becoming more and more practically feasible due to rapid advancements in storage technology. One of the properties of this layer is the introduction of it at this position, not at a higher position like most models do.

All contextual data that are being collected (the previous layer) are fed into this layer. In addition, domain knowledge and inference rules are stored in the database. We view this layer as the kitchen that will always be used to prepare the recipes of the following layer. Querying context data stored in the database could be done through the Structured Query Language (SQL) that provides the ability to read from and write to the database at a high level of abstraction. Responsibilities of this layer encompass in addition to storing context data, maintaining the integrity of data, and the efficient utilization of storage.

Context Processing Layer: This is the layer that gives the taste of the whole process. It pulls instant and old contextual data from the layer below (we like to call it cupel) and does some processing (cupellation) in order to provide the required knowledge that the application will depend on to adapt its behavior. Note that some processing activities may also write some results back into the Context Storage Layer. Context here is being prepared into a set of levels determined according to the context type; i.e. levels help to provide coarse grained and fine grained context. Since most consumers are interested in already interpreted and aggregated information than raw data; technical data gathered and stored in the previous layer are being analyzed and reasoned about in a way that makes context more readable and beneficial to context-aware applications. Levels mentioned above include raw data and processed context. We recommend the approach of having a context knowledge base and an inference engine that uses various inference rules and information gathered from sensors that are saved in the database to deduce the leveled context. The inference engine follows rules in the database and applies the well known forward chaining technique, for

example, in order to create the specified level of abstraction needed and to help the application make the proper decision of adaptation according to the application's conditional rules.

Context Distribution Layer: This layer has the responsibility of delivering context to the interested requestors. At this layer there is a kind of protecting agent that will decide whether it is safe to give correct information to the requestor. Also, we put a requirement in this layer which states the necessity of providing context in a universal format that should be understood regardless of the communicating parties' (providers and consumers) hardware and software specifications. This requirement's importance is clearly apparent; it will increase the usability of the model and the applications conforming to the model, since it overcomes the heterogeneous context data sources issue.

This layer contains a Context Protection Sub-Layer, Context Publisher Sub-Layer, and Context Universalization Sub-Layer.

(1) Context Protection Sub-Layer: The protective shield is strongly needed when we are suggesting that there will be many distributed web-enabled devices having user profiles and other sensitive information and which may deal with malicious applications. It will be the firewall that must be dealt with at first to take the contextual information requested. This firewall would consult a set of rules that may include user ownership rules to determine the safest action to the node. User ownership rules state for example that this kind of context information could be given to the application satisfying some criteria. So, the context protection sub-layer resembles the function of a filter that queries the database and allows only a subset of context information to be sent to each type of application according to predefined rules, (i.e. every node defines a set of constraints to determine when it is allowed to give certain aspects of context to the requesting application).

(2) Context Universalization Sub-Layer: Context Universalization Sub-Layer has the responsibility of converting the contextual information required at the specific level into a format that can be interpreted by all parts of the communication running whatever software on whatever hardware architecture. Having each framework present today with its own format to describe context and with its own communications mechanisms makes it difficult, if not impossible, to look forward an open context framework. Standardized formats and protocols, on the other hand, make it more realizable and possible to achieve this goal and bring the advantage of concentrating more on the product or service being developed rather than on the communication and the type of node being communicated. According to current software technologies, Web Services could achieve the task and make context aware services more interoperable. Interoperability is important since it enables developers to use Web services without thinking about which programming language or operating system the services are hosted on.

(3) Context Publisher Sub-Layer: This layer also contains a sub-layer which we called a Context Publisher. It has the responsibility of advertising the services of the node, i.e. which contexts are provided by the node. It does this work after consulting the firewall.

Node-Model Discussion: Some of the important features in this model are: the position of the storage layer, the multiple layers that mainly aim to give multiple levels of abstraction to the same contextual data and the introduction of the firewall at the top of the hierarchy. The model asserts the requirement of providing context aspects in a standard format to overcome the heterogeneity difficulties. The model should be realized and implemented by each web-enabled (has an IP) node. Therefore, we have a distributed system where sensing, context storage, and context processing are all performed remotely at the node. The application will communicate with the context distribution layer to obtain the needed contextual information. In other words, there is a separation between detecting and processing context on one side and using context on the other side. Requiring each node to be responsible for the control and management of its own context is beneficial as small (mobile) devices or PCs will be capable of handling the processing level. Furthermore, we may avoid the single point of failure problem associated with a central location of storage and processing. This feature should improve the modularity and reusability of systems.

The publisher object is there to allow the application to know which sensors are attached to the node and therefore to determine which parts of the application may be adapted.

3.2 Web-Model

From the application viewpoint, we will model the process that the application should perform in order to complete the context-awareness cycle. The aim is to make the web application, which is created cooperatively by the analyst and the programmer, able to understand the activities (situations) and understand the preferences of the user when he/she is engaged in various activities. Figure 2 shows the suggested model and a discussion of its components follows.

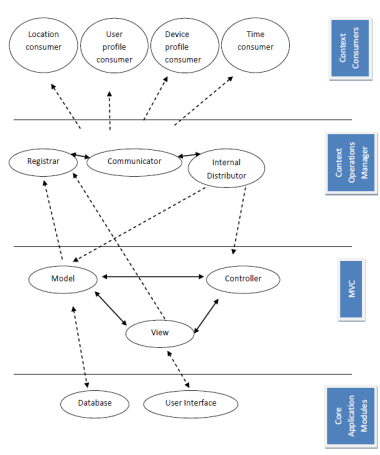


Fig. 2. The model at the web application

Core Application Modules: these include the primary functionalities presented by the web application which are represented by the database, user interface, and the variety of services dealing with the database and interacting with (e.g. sending output to or taking input from) the user interface.

MVC: on top of the database and the user interface come the MVC objects which act as a mediator that communicate with the upper layer on behalf of the lower layer(s).

Context Operation Manager: moving bottom up, we are going more and more into being context aware. We take this class to have the main tasks of contextualizing the content and behavior of the web application. We divided its responsibilities into the following:

(1) Registrar. The Registrar object is will be contacted by the model and the view on behalf of database and user interface. Its job is simply to record that these database items are interested in this contextual information (data – content – adaptation) and that these user interface items have the potential to be adapted based on the following contextual information, etc.
(2) Communicator. The Communicator, having channel with the upper layer, will be told by the Registrar object that this web application (database and interface) is demanding to have what types of context from the node at the other end. Armed with that information about the required types, the Communicator contacts the specified consumers – according to a list specifying what type of context could be brought by what consumer – and waiting for their quick reply before timing out after a predetermined time in which case it will return a default value for each context type not being brought on time.
(3) Internal Distributor. When the Communicator finishes its job, the contextual information would be available to be delivered to those parts of the database and user interface that registered their sensitivity to some context.

Context Consumers: Context Consumers at the web application server node will call their counterparts and Context Providers, at the Context Distribution Layer. Negotiations take place to bring the context required at all available levels that could be agreed upon between the web application and the client node. Some types of context information could be cached to overcome any performance issues emerging from continuous demanding of the same context information by various database and user interface items.

4 Case Study: University Portal Prototype

Aiming at improving the usability of a previously created website, we have redesigned it to be context-aware to some aspects based on our second model. In addition, to measure the impact of this improvement; we have provided a PC with the mechanisms needed to disseminate some context.

We have used Java programming language to develop the classes, interface, and the web service that reside at the node and to develop the JSP pages of the portal. We have created three classes as shown in Figure 3 below.

The Universalizer class is the one that is converted to a web service in order to constitute the interface to various web applications asking for the node's context. Our interests, when implementing the prototype, were limited to the context types: Location, Time, Role and Device.

The scenario is as follows: the web application (the portal) is developed and deployed at Server A. There are other nodes (PCs) in the network that wish to access the application. For the purpose of simulation, we have used PCs (Desktops and Laptops), and we have deployed the web application and the web service on a standalone WebLogic server installed on each device (the server holding the web application and the PCs holding the web services).

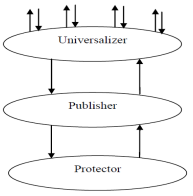


Fig. 3. Interaction of suggested implementation classes

WebLogic is an application server that has to be configured appropriately and will work as the container for the web service and the application. Each PC will have its own database (could be the light version of a database). When PC_1, for instance, accesses the portal, the servlet (a type of java class) which is programmed to receive the request will extract PC_1 IP address in order to use it in further communications. Then the web application will check if the node is committed to give the context types of interest to the web application, so it will use the IP address of PC_1 and try to access the web service that should be there in the compatible nodes. If there is no such web service, the application could use the mechanisms of exception handling to deal with such cases and will put a default values in the variables representing the context types of interest to it. If the service is there, then further communication will take place and the chain of classes mentioned at PC_1 will do the job and will disseminate just those context types/levels that are allowed to be disseminated to such an application. Therefore, we have eliminated the discovery phase mentioned in the service oriented architecture as it does not make sense to discover what is already known. A web application has to check if the web service exists on the node and if the answer is positive then it can ask for the contextual aspects it needs.

On the web application side, the sections of the page could, for instance, be adapted according to the PCs' locations. For example, some section may be interested in advertising to different places in the university, where several advertisements are being addressed to distinct locations (faculties, centers, etc…). The benefits should be apparent since contacting PC_1 will say to the web application at which location it sets exactly and the application will adapt accordingly by prioritizing those advertisements pertaining to the location of the node.

Adaptation of the portal could be accomplished for the following aspects: Content, Presentation, and Navigation. For the content aspect, the same page requested may display different content according to the user's role, location, or time of access. For example, if the user is accessing the page from inside the university, then the page may display advertisements of current place of access. However, if the access is from outside, then more general advertisements may appear. Adaptation of presentation may mean to adapt the appearance of the application's user interface according to the device's display size. Adaptation of navigation is performed in the portal by suggesting to go to Acad (academic system used to enter students' grades) system or employees' email if the user is one of the faculty members, or to go to Reg (students' registration page) or students' email if the user is a student.

The models gave us, as developers, a clear map of what we have to do utilizing a technology that is available to every developer. Distributing the requirements of context awareness and forcing every part to share a task; this hybrid approach follows a simple and logical manner for dividing these tasks and adopts a practical way to implement the final product where the emphasis is on the technology not on the framework used for implementation.

5 Conclusions and Future Work

In this paper we have proposed two models that organize knowledge in layers, and complement each other, in order to give the web applications' developers the adequate knowledge and a visualization of what should be performed to develop a context aware application. We have emphasized the model which resides at the node side. It would be beneficial to investigate ways to elaborate the model which resides at the application server. The investigation may involve the ability to enhance the context awareness potential of legacy systems. This may require the introduction of a new layer at the top of model which pertains to the web application.

Some of the layers of the first model need more analysis, especially the context processing layer. We believe that the improvement could address some questions such as: how exactly we can benefit from the database at the lower layer and what kind of analysis, to the context, could be performed so that the required results could be given in a reasonable time.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-Aware Systems. Int. J. Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
2. Beigl, M., Krohn, A., Zimmer, T., Decker, C., Robinson, P.: AwareCon: Situation Aware Context Communication. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 132–139. Springer, Heidelberg (2003)
3. Bradley, N.A., Dunlop, M.D.: Towards a Multidisciplinary Model of Context to Support Context-Aware Computing. Human–Computer Interaction 20, 403–446 (2005)

4. Chen, H., Finin, T., Joshi, A.: Using OWL in a Pervasive Computing Broker. In: Proc. Workshop on Ontologies in Open Agent Systems (OAS), pp. 9–16 (2003)
5. Dey, A.: Understanding and Using Context. Personal and Ubiquitous Computing 5(1), 4–7 (2001)
6. Dey, A., Abowd, G.D., Wood, A.: CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services. Knowledge-Based Systems 11, 3–13 (1998)
7. Jahnke, J., Bychkov, Y., Dahlem, D., Kawasme, L.: Context-Aware Information Services for Health care. In: Proc. KI 2004 Int. Workshop on Modelling and Retrieval of Context, pp. 73–84 (2004)
8. Louis, S.J., Shankar, A.: Context learning can improve user interaction. In: Zhang, D., Gregoire, E., DeGroot, D. (eds.) IEEE Systems, Man and Cybernetics Society, pp. 115–120 (2004)
9. Schilit, B., LaMarca, A., Borriello, G., Griswold, W.G., McDonald, D., Lazowska, E., Balachandran, A., Hong, J., Iverson, V.: Challenge: Ubiquitous Location-Aware Computing and the "Place Lab" Initiative. In: 1st ACM Int. Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, New York, pp. 29–35 (2003)
10. Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. In: Sixth Int. Conf. on Ubiquitous Computing, pp. 33–40 (2004)

Short-Term Spatio-temporal Forecasts of Web Performance by Means of Turning Bands Method

Leszek Borzemski, Michal Danielak, and Anna Kaminska-Chuchmala

Institute of Informatics, Wroclaw University of Technology,
ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{leszek.borzemski,michal.danielak,anna.kaminska-chuchmala}@pwr.wroc.pl
http://www.ii.pwr.wroc.pl/index.php/en/institute

Abstract. This work presents Turning Bands simulation method (TB) as a geostatistical approach for making spatio-temporal forecasts of Web performance. The most significant advantage of this method is requirement for the minimum amount of input data to make accurate and detailed forecasts. For this paper, necessary data were obtained with the Multiagent Internet Measuring System (MWING); however, only those measurements of European servers that were collected by the MWING's agent in Gdansk were used. The aforementioned agent performed measurements (i.e. download times of the same given resource from the evaluated servers) three times every day, between 07.02.2009 and 28.02.2009, at 06:00 am, 12:00 pm and 06.00 pm. First, the preliminary and structural analyses of the measurement data were performed. Then short-term spatio-temporal forecasts of total downloading times for a four days ahead were made. And finally, thorough analysis of the obtained results was carried out and further research directions were proposed.

Keywords: Web performance, spatio-temporal forecasts, geostatistics, Turning Bands method.

1 Introduction

The amount of data traffic in the Internet is constantly and rapidly growing. According to the forecast presented in [1], in 2016 traffic generated only by mobile devices will be six time greater than the one that is generated now. This is partially caused by the new phenomenon - the Internet of Things. The Internet is gradually becoming the place where tiny devices of everyday life such as tablet computers, tv receivers or even mere clocks or refrigerators, being uniquely identifiable objects are able to communicate with one another. What is more, web users have always been expecting high efficiency of Internet services, especially when modern technology allows the same services to be present on many servers. In this case, a subjective quality indicator, seen from the perspective of a web user, may be introduced. Forecasts of how good particular servers (seen from a given user) may operate, can help to achieve an efficient Web performance.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 132–141, 2012.

1.1 Related Works

Web (or the Internet) performance forecast is a very important issue. Therefore, in the literature various approaches may be found: some research, for instance, deal with the Internet performance using transmission delay (Round Trip Time, RTT), other solve it using data throughput (TCP throughput). Moreover, Web performance can be evaluated by measuring either loading responses of Web pages or download times of Web resources (see: [2]-[6]).

Two methods of Web performance forecast can be distinguished: formula-based and history-based. First method use mathematical formulae to express particular performance measure as a function of essential independent variables, which characterise a studied phenomenon. The latter method, which was used in this paper, analyses series of observations which were obtained by making measurements with a certain interval over time. Another example of the latter method may be found in [7]. So far, the geostatistical methods have been used mainly in areas such as geology, mining, oceanography and hydrology. Nevertheless, in recent times these methods were used in completely different fields of science such as geodesy and cartography [8] or economic analysis [9]. Moreover, geostatistical methods have been also used to study the problem of spatial distribution of floating car speed [10] that seems to be similar to the problem of traffic data packets on the Internet. Currently, to the best of the authors' knowledge the geospatial approach to Web performance prediction presented in this paper is unique as developed in our papers, leaving no similar problem statement in the literature. More information about our approach may be found in [11], [12].

This paper presents the Turning Bands method as a geostatistical approach to predict user-perceived performance of the Web. We have chosen TB for a few reasons. First, it requires the minimum amount of input data to perform a spatio-temporal forecast. Second, this forecast is carried out not only for the studied servers, but for the whole examined area. And finally, this method has successfully proven itself in forecasting load in electrical networks [15], [16].

2 Turning Bands Method

Turning Bands method, originally initiated by Matheron, is stereologic tool that allows to reduce multidimensional simulation to one-dimensional [13], [14].

Let us consider a stationary Gaussian random function with mean equal to 0, variance equal to 1 and covariance C that is continuous in $D \in R^d$. According to Bochner's theorem, covariance C can be defined as the Fourier transform of positive Borel measure, for instance χ:

$$C(h) = \int_{R^d} e^{i<h,u>} d\chi(u) \; . \tag{1}$$

Also $C(0) = 1$, so χ is a measure of the probability. After the introduction of the polar coordinate system $u = (\theta, \rho)$, where θ is the directional parameter of the hemisphere S_d^+ i ρ is the location parameter $(-\infty < \rho < \infty)$, spectral measure

$d\chi(u)$ can be expressed as the product of decomposition $d\varpi(\theta)$ and conditional distribution $d_{\chi\theta}(\rho)$ for a given θ. After using this distribution to develop the spectral covariance C and the introduction of one-dimensional function $C_\theta(r)$ Bochner's theorem was used, so that the covariance function $C(h)$ can be expressed as:

$$C(h) = \int\limits_{S_d^+} C_\theta(< h, \theta >) d\varpi(\theta) \ , \tag{2}$$

where C_θ is also a covariance. Therefore TB consists in reducing the simulation of a Gaussian function with covariance C to the simulation of an independent stochastic process with covariance $C(h)$.

Let $(\theta_n, n \in \mathbf{N})$ be a sequence of directions S_d^+ and let $(X_n, n \in \mathbf{N})$ be a sequence of independent stochastic processes of covariance C_{θ_n}, Random function:

$$Y^n(x) = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} X_k(< x, \theta_k >), x \in \mathbf{R} \tag{3}$$

takes covariance equal to:

$$C^n(h) = \frac{1}{n} \sum_{k=1}^{n} C_{\theta_k}(< h, \theta_k >) \ . \tag{4}$$

The central limit theorem shows that for very large n, $Y(n)$ tends to Gaussian distribution with variance $\lim_{n\to\infty} C^{(n)}$. When series $\frac{1}{n}\sum_{k=1}^{n} \delta_{\theta_k}$ converges weakly to ϖ this limit is exactly C.

Turning Bands[1] algorithm may be presented in the following way:

1. Transform input data using Gaussian anamorphosis.
2. Select directions $\theta_1, ..., \theta_n$ so that $\frac{1}{n}\sum_{k=1}^{n} \delta_{\theta_k} \approx \varpi$.
3. Generate standard, independent stochastic processes $X_1, ..., X_n$ with covariance functions $C_{\theta_1}, ..., C_{\theta_n}$.
4. Calculate $\frac{1}{\sqrt{n}}\sum_{k=1}^{n} X_k(< x, \theta_k >)$ for every $x \in D$.
5. Make kriged estimate $y^*(x) = \sum_c \lambda_c(x) y(c)$ for each $x \in D$.
6. Simulate a Gaussian random function with mean 0, covariance C in domain D on condition points. Let $(z(c), c \in C)$ and $(z(x), x \in D)$ be the obtained results.
7. Make kriged estimate $z^*(x) = \sum_c \lambda_c(x) z(c)$ for each $x \in D$.
8. Obtain the random function $W(x) = (y^*(x) + z(x) - z^*(x), x \in D)$ as the result of conditional simulation.
9. Perform a Gaussian back transformation to return to the original data.

[1] TB and conditional simulations are discussed in more detail in [17], [18].

3 Preliminary Data Analysis

To create a database containing input data, necessary to make spatio-temporal forecasts using TB, we used measurements obtained with multiagent system MWING [19], [20] and [21]. This system consists of agents that are distributed throughout the world. One of their tasks is to measure times needed to download a copy of the same file (which is a Request for Comments text document, rfc1945) from many web servers.

In this paper, the measurements of European servers, collected by the agent in Gdansk were used. These measurements were taken three times a day, between 07.02.2009 and 28.02.2009, at 06:00 am, 12:00 pm and 06.00 pm. In the next step, these data were used to create the aforementioned database. This database contained information such as servers' locations (their latitudes and longitudes), downloading times, and timestamps for each measurement; all this information was necessary to make spatio-temporal forecasts using TB.

Table 1 presents basic statistics of Web performance for considered servers. The largest span of data occurs for 06:00 am where the difference between minimum and maximum value is 28.95 seconds. Not only does high value of kurtosis (more than 3) indicate the variability of the examined process, but it also shows big right-side asymmetry of the examined phenomenon. Taking into account both high skewness and the fact that the whole idea is to achieve a distribution as close as possible to a symmetric distribution, logarithmic values were calculated for data obtained for every hour.

Table 1. Basic statistics of download times from evaluated web servers, taken between 07.02.2009 and 28.02.2009

Statistical parameter	**6:00 AM**	**12:00 PM**	**6:00 PM**
Minimum value X_{min} [s]	0.11	0.12	0.12
Maximum value X_{max} [s]	29.06	12.15	7.93
Average value X [s]	0.60	0.62	0.60
Standard deviation S [s]	1.59	1.08	0.77
Variability coefficient V [%]	266	174	129
Skewness coefficient G	15.35	7.25	4.99
Kurtosis coefficient K	265.65	64.16	34.61

Figure 1 presents Q-Q plots (Q stands for quantile) before and after the calculation of logarithms for 12:00 pm. The points follow a strongly nonlinear pattern, suggesting that the data are not distributed as a standard normal.

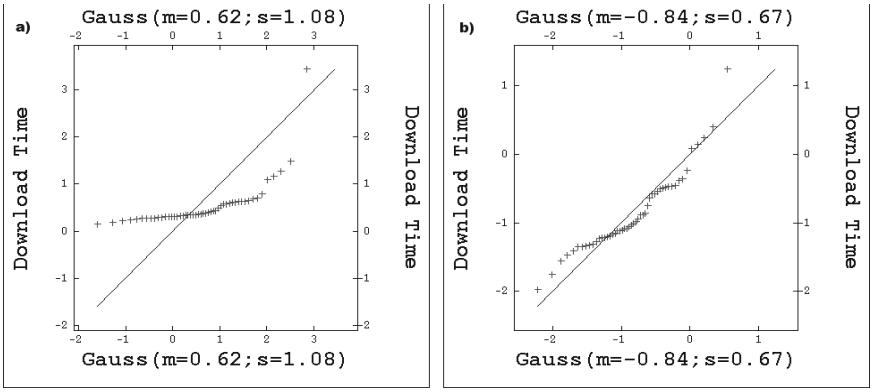


Fig. 1. Q-Q plots for the measured download times at 12:00 pm, before (a) and after the calculation of logarithms (b)

However the points in the second Q-Q plot follow pattern that is slightly similar to normal distribution.

4 Structural Data Analysis

Calculation of Gaussian anamorphosis is the first step after making the preliminary data analysis. To calculate Gaussian transformation frequency, the inversion model was used and the number of adopted Hermite polynomials was equal to 100.

The next step in structural data analysis is modeling of a theoretical variogram function. For 12:00 pm and 06:00 pm hours, approximated theoretical variograms consisted of nuggets effect and K-Bessel. In case of 06:00 am, approximated theoretical variogram consisted of nuggets effect and J-Bessel; the variograms of Web performance for 12:00 pm and 06:00 pm were approximated by the model of nuggets effect and K-Bessel. Directional variograms were calculated along the time axis (for 90^o direction). Figure 2 presents an example of theoretical variogram. The distance classes equal 5.69, 7.76 and 4.34 for 06:00 am, 12:00 pm and 06:00 pm respectively. All the variograms indicate a gentle rising trend.

5 Spatio-temporal Web Performance Using the Turning Bands Method

To make spatio-temporal forecasts of Web performance for 06:00 am, 12:00 pm and 06:00 pm, we created three types of forecasting models. These models consist of aforementioned in the previous section variogram models with their respective

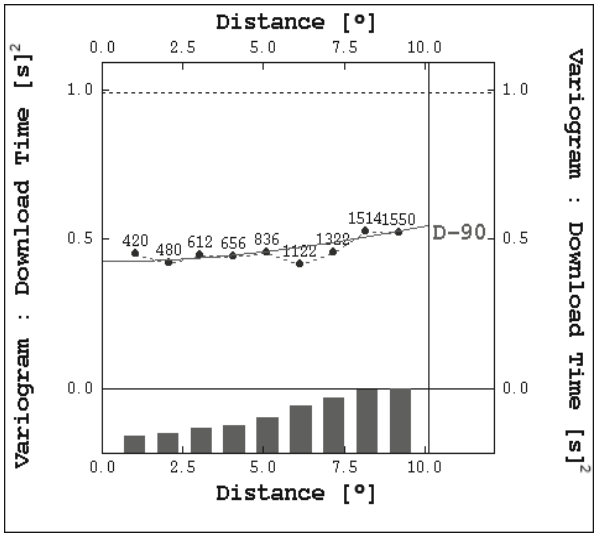


Fig. 2. Directional variogram along the time axis for download times for 06:00 am, approximated with the theoretical model of nuggets effect and J-Bessel

Gaussian anamorphosis. Additionally, for every forecast made by means of these models, the moving neighbourhood and the block type of simulation were selected. What is more, these forecasts of download times were determined on the basis of realisation of one hundred simulations.

Three-dimensional forecasts were made four days ahead, for the period between 01.03.2009 and 04.03.2009. Global statistics of these forecasts may be found in Table 2. The forecasts conducted for 06:00 pm are characterised by the highest variance coefficient and standard deviation. For other hours, both standard deviation and variance coefficient are similar; therefore it can be assumed that in these cases forecasts may be performed with better accuracy. The mean forecasts errors are 26.91%, 20.00% and 17.55% for 06:00 am, 12:00 pm and 06:00 pm respectively.

Table 2. Global statistics for the four-day forecasts of Web performance

Geostatistical Parameter	Minimum value $Z_{min}[s]$	Maximum value $Z_{max}[s]$	Average value $Z[s]$	Variance S^2 $[s]^2$	Standard deviation $[s]$	Variance coefficient $V[\%]$
Mean Forecasted value Z, for 06:00 am	0.15	1.04	0.45	0.03	0.16	36
Mean Forecasted value Z, for 12:00 pm	0.15	1.28	0.46	0.03	0.16	35
Mean Forecasted value Z, for 06:00 pm	0.14	1.61	0.45	0.04	0.20	45

The best results of forecast were obtained for the server in Rome and presented in Figure 3. The mean forecasts errors for this particular server equal to 21.43, 2.39 and 6.16 for 06:00 am, 12:00 pm and 06:00 pm respectively. The high value of the mean forecast error for 06:00 am could be caused by the high values of kurtosis coefficient and skewness for historical data (Table 1).

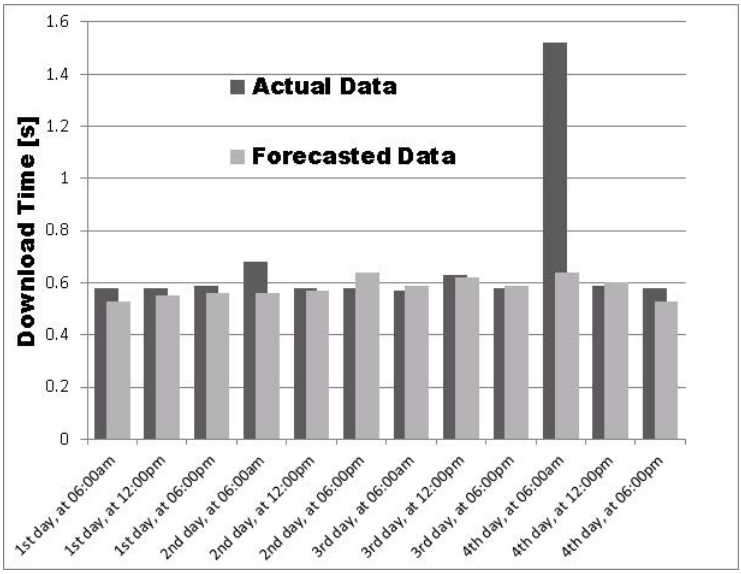


Fig. 3. Forecasts results of Web performance for the server in Rome, Italy

Figures 4 and 5 present historical, original and forecasted download times for two selected servers: the first is located in Rome and the other in Warsaw; data for both servers concern 12:00 pm. The forecasted errors for these servers are 2.39% and 37.46% for Rome and Warsaw respectively. Based on a thorough analysis of the obtained data, it can be stated that this difference in forecast error was caused due to differentiation in historical data. Namely, the historical data for the first server had been relatively moderate with only one peak on 21st day; historical data for the server in Warsaw, however, contain three unpredictable peaks which make an accurate forecast almost impossible to achieve.

Sample forecasts results for the whole considered area for 06:00 am are presented as a raster map in figure 6. Crosses shown on the map represent examined servers and the size of these crosses corresponds to the real times needed to obtain a file from a given server. The server with the largest download time is located in Frederikshavn, Denmark.

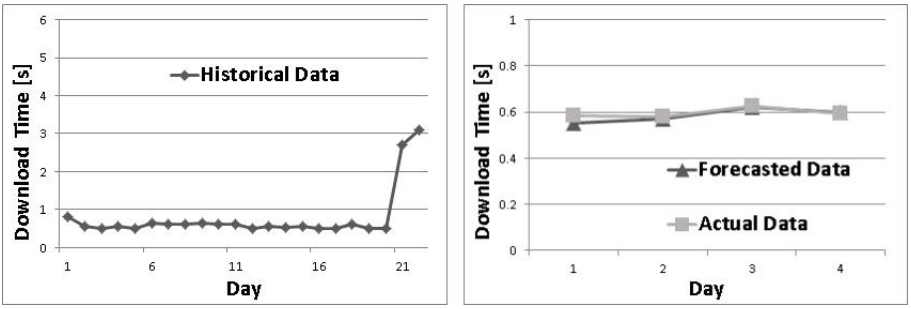


Fig. 4. Historical, original and forecasted download times for the web server in Rome, Italy, at 12:00 pm

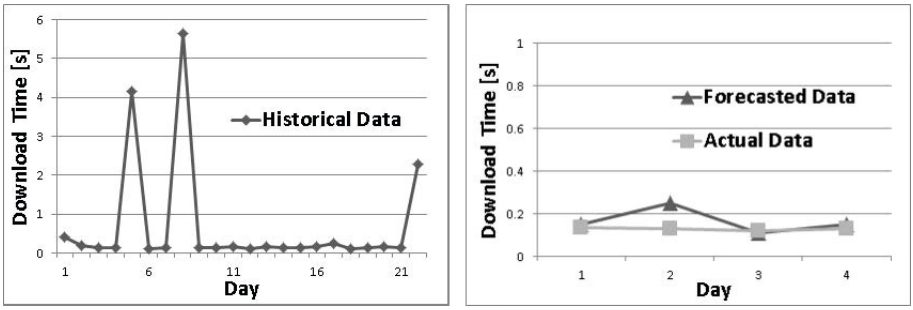


Fig. 5. Historical, original and forecasted download times for the web server in Warsaw, Denmark, at 06:00 am

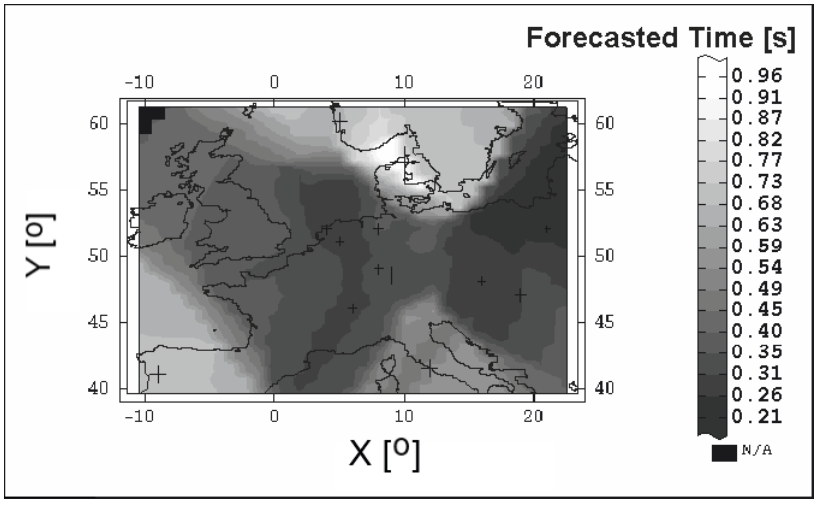


Fig. 6. Sample raster map showing forecasted download times on March 3, 2009, at 06:00 am

6 Summary

This paper presented TB as a geostatistical approach for making spatio-temporal forecasts of web servers performance. Such research may be helpful in analysing both network traffic and web servers performance in a particular area. What is more, the obtained results justify the usage of TB in making spatio-temporal forecasts of Web performance. These forecasts can be an essential and key element in building Internet service providers' knowledge and thus contribute to improve the quality of their services, especially when technology is developing rapidly and users are becoming increasingly demanding.

Nevertheless, based on the obtained results, it can be stated that there is still a need to improve the accuracy of forecasts. This could be achieved by making them for different scenarios, varying in the type of measured values, their timestamps, and the length of time horizons. Moreover, conducting forecasts only for the servers that form the backbone of the Internet could also produce interesting results. However due to difficulties in collecting necessary data, this task is extremely difficult or even impossible to achieve at this moment in time.

References

1. Cisco Visual Networking Index (VNI) Global Mobile Data Traffic Forecast for 2011 to 2016, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf
2. CAIDA (Cooperative Association for Internet Data Analysis), http://caida.org
3. Mirza, M., Sommers, J., Barford, P., Zhu, X.: A machine learning approach to TCP throughput prediction. IEEE ACM T. Network 18(4), 1026–1039 (2010)
4. Karrer, R.: TCP prediction for adaptive applications. In: Proc. 32nd IEEE Conference on Local Computer Networks, pp. 989–996 (2007)
5. He, Q., Dovrolis, C., Ammar, M.: On the predictability of large transfer TCP throughput. Comput. Netw. 51(14), 3959–3977 (2007)
6. Yin, D., Yildirim, E., Kulasekaran, S., Ross, B., Kosar, T.: A data throughput prediction and optimization service for widely distributed many-task computing. IEEE Trans. Parall. Distr. 22(6), 899–909 (2011)
7. Borzemski, L.: Internet path behavior prediction via data mining: Conceptual framework and case study. J. Univers. Comput. Sci. 13(2), 287–316 (2007)
8. Sunila, R., Kollo, K.: A comparison of geostatistics and fuzzy application for digital elevation model. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI-2/C43 (2007)
9. Amiri, A., Gerdtham, U.: Relationship between exports, imports, and economic growth in France: evidence from cointegration analysis and Granger causality with using geostatistical models. Munich Personal RePEc Archive Paper No. 34190 (2011)
10. Wang, Y., Zhuang, D., Liu, H.: Spatial Distribution of Floating Car Speed. Journal of Transportation Systems Engineering and Information Technology 12(2), 36–41 (2012)
11. Borzemski, L., Kaminska-Chuchmala, A.: Client-Perceived Web Performance Knowledge Discovery through Turning Bands Method. Cybern. Syst. 43(4), 354–368 (2012)

12. Borzemski, L., Kaminska-Chuchmala, A.: Distributed Web Systems Performance Forecasting Using Turning Bands Method. IEEE. Trans. Ind. Inform. PP(99), 1, doi:10.1109/TII.2012.2198644, ISSN=1551-3203
13. Matheron, G.: Quelques aspects de la montée. Internal Report N-271, Centre de Morphologie Mathematique, Fontainebleau (1972)
14. Matheron, G.: The intrinsic random functions and their applications. JSTOR Advances in Applied Probability 5, 439–468 (1973)
15. Kaminska-Chuchmala, A., Wilczynski, A.: 3D electric load forecasting using geostatistical simulation method Turning Bands. The works of Wroclaw Scientific Society, series B, XVI(215), 41–48 (2009)
16. Kaminska-Chuchmala, A., Wilczynski, A.: Analysis of different methodological factors on accuracy of spatial electric load forecast performed with Turning Bands method. Rynek Energii 2(87), 54–59 (2010)
17. Lantuejoul, C.: Geostatistical Simulation: Models and Algorithms. Springer (2002)
18. Wackernagel, H.: Multivariate Geostatistics: an Introduction with Applications. Springer, Berlin (2003)
19. Borzemski, L., Cichocki, Ł., Fraś, M., Kliber, M., Nowak, Z.: MWING: A Multiagent System for Web Site Measurements. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 278–287. Springer, Heidelberg (2007)
20. Borzemski, L., Cichocki, Ł., Kliber, M.: Architecture of Multiagent Internet Measurement System MWING Release 2. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS, vol. 5559, pp. 410–419. Springer, Heidelberg (2009)
21. Borzemski, L.: The experimental design for data mining to discover web performance issues in a Wide Area Network. Cybern. Syst. 41(1), 31–45 (2010)

Extreme Propagation in an Ad-Hoc Radio Network - Revisited*

Przemysław Błaśkiewicz, Mirosław Kutyłowski, Wojciech Wodo, and Kamil Wolny

Wrocław University of Technology, Poland
firstname.secondname@pwr.wroc.pl

Abstract. We consider the algorithm by Baquero, Almeida and Menezes that computes extreme values observed by nodes of an ad hoc network. We adapt it to meet specific technical features of communication in wireless networks with a single channel based on time multiplexing. Our approach leads to substantial reduction of the number of messages transmitted as well as execution speed-up.

1 Introduction

One of fundamental problems of data aggregation in wireless sensor networks is to compute an extreme value of the values observed. This could be for instance maximum temperature observed in a forest – indicating probability of fire outbreak. In most sensor networks monitoring environment, the most crucial functionality is warning about extraordinary conditions (like exceeding admitted level of concentration of toxic substances). In such a case the highest value should be propagated as fast as possible. Due to technical limitations the aggregation scheme should fulfill the following conditions:

- the network nodes use a narrow radio channel,
- the energy usage should be minimized in order to save batteries of the nodes/sensors,
- the nodes are not synchronized, the nodes may leave and join the network; network architecture may be unknown to the network nodes.

Due to energy constraints, the nodes should transmit low energy signal – this limits perimeter in which the signal can be received. So in many cases we have to deal with multi-hop networks. The number of messages sent and their volume should be minimized in order to leave place for other channel usage.

Network Model. The network is modelled as a graph, in which each vertex corresponds to a node in the network. Nodes A and B are connected by an undirected edge, if the message sent by node A can be received by node B and vice versa (for simplicity we assume that the links are bidirectional and that a node changes neither the location nor the strength of the signal). The algorithm should work without knowledge of the graph architecture - the only information accessible for a node are the messages received from its intermediate neighbors.

* The paper was partially supported by EU Operational Programme Innovative Economy 2007-2013 POIG.01.03.01-02-002/08-00.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 142–151, 2012.

Time Model. We assume that the time is continuous. Each node holds a clock but the clocks are only loosely synchronized. If a node decides to send a message at time t, then it chooses a time moment $t' \in [t, \leq t + \Delta]$, when it starts the transmission. As each transmission has some length (say δ), it may happen that difference between two starting points of the transmissions of two nodes is lower than δ and a collision occurs. We assume that Δ is large enough so that the probability of collisions is practically negligible. This corresponds to the model from [2] with carrier detection (for further details see [3]).

Maximum Propagation Problem. Let a network consist of nodes $C_1, \ldots, C_n$ holding, respectively, values $x_1 \ldots, x_n$ from $\mathbb{R}$. *Maximum propagation* is a task such that upon its termination each node of the network is aware of the value $\max(x_1, \ldots, x_n)$.

Optimization targets of *maximum propagation* are the number of messages sent and termination time.

Protocol Extrema Propagation. In this paper we consider a variant of the following simple protocol from papers [1], [6] and [4] solving the maximum propagation problem. It executes periodically the following round until all nodes are aware of the maximum. The variable ξ is the current candidate for the maximum hold by the node (ξ is initialized with the own value of the node):

Algorithm 1. a round of Extrema Propagation

1: **if** the maximum value c received from neighbors in the previous round exceeds ξ **then**
2: $\quad \xi \leftarrow c$
3: $\quad$ broadcast ξ to all neighbors

When we attempt to implement Algorithm 1 we have to deal with the problem of simultaneous transmission by all nodes of the network – direct implementation would lead to signal interferences and inability to receive correctly the messages sent.

One can fix this problem in a couple of ways. The first method is to use a large number of communication channels so that there is no interference between neighboring nodes. This reduces to the problem of coloring nodes of the network graph, so that no neighboring nodes get the same color (a color corresponds to a brodcast channel used by such a node). However, graph coloring with a minimal number of colors is hard, if the network architecture is unknown, the decisions must be taken locally. Instead of creating multiple channels one may use a single channel. Now the time is split into slots and the slots are assigned to nodes in a periodic way. However, this again assumes some preconfiguration of the network and knowledge of its architecture.

A fairly practical approach (see [2]) is to choose a moment of transmission at random within an interval $[t, t + \Delta]$ with the hope that there will be no interference. In this case, sometimes a node receives a value which is higher than the value which the node intended to send. Obviously, the node should transmit the new maximum, but the

main design decision is *when to transmit*. Should we finish the current "round" of Algorithm 1 or restart in some way? In this paper we consider the second option and show that it leads to quite promising performance.

2 Asynchronous Maximum Propagation Algorithm

In our wireless communication model, Algorithm 1 can be reformulated as follows. Let Δ be the length of a single round. Then each node executes the following code during round i:

Algorithm 2. round i of Extrema Propagation with a single radio channel

```
1: t ← Random(iΔ, (i + 1)Δ)
2: if the maximum value c received from neighbors in the previous round exceeds ξ then
3:     ξ ← c
4: if time t elapsed then
5:     broadcast ξ to all neighbors
6:     t ← ∞
```

We propose a modification of this algorithm – see the pseudo-code given in Algorithm 3. The main differences are the following: there is no strict splitting into rounds (so strict synchronization is not required anymore and implementation is much easier). Second, when a node obtains a new maximum at time t', then it resets its intended transmission time to a time t chosen uniformly at random from the interval $[t', t' + \Delta]$.

Algorithm 3. Asynchronous Max Propagation

```
1: ξ ← x_i
2: t ← Random(0, Δ)
3: loop
4:     wait until time t or message received
5:     if message received at time t' and the value c received is > ξ then
6:         ξ ← c
7:         t ← Random(t', t' + Δ)
8:     else if time t elapsed then
9:         broadcast ξ to all neighbors
10:        t ← ∞
```

For the above algorithm we have to address the following questions:

- as the protocol redefines transmission periods, we have to make sure that it does not introduce time periods where too many nodes attempt to transmit,
- what is the execution time compared to the time of Algorithm 2,
- what is its message complexity compared to Algorithm 2.

We address these questions, respectively, in Sect. 3, 4, and 5.

3 Congestion of Transmissions

For Algorithm 2 time period Δ is long enough, so that n stations transmit at random time moments with practically negligible chance of a collision. Let us consider the situation for Algorithm 3. For simplicity, we consider here the complete graph consisting of n nodes. Initially, the nodes choose their transmission times uniformly at random from $[0, \Delta]$. However, if at a time t some node transmits a value v, then the following nodes choose their transmission times anew in the interval $[t, t + \Delta]$:

- all nodes that have transmitted up to time t,
- all nodes that have not transmitted so far but hold values smaller than v.

The remaining nodes keep previously chosen transmission times. Thereby we change probability of using time moments in the interval $[t, \Delta]$: out of n stations some choose uniformly at random within $[t, \Delta]$ and some use interval $[t, t+\Delta]$. So probability density for transmission within interval $[t, \Delta]$ becomes higher.

Below we argue why this "hot spot" effect is in practice negligible. Let $D = D_t$ be a function of density transmissions. That is, given an interval J, then $\int_J D(x)\ dx$ is the expected number of transmissions in the interval J, conditioned by the protocol execution up to time t. Initially, D is a constant function with value n/Δ on $[0, \Delta]$ and equal to 0 elsewhere, and each node contributes value $1/\Delta$ on this interval.

If $t = p \cdot \Delta$, and $p \ll 1$, then from the point of view of points from $[t, \Delta]$ it is not substantial if a node has transmission time distributed uniformly in $[t, \Delta]$ or in $[t, t+\Delta]$. If p is not very small, then the value transmitted so far come from about $n \cdot p$ nodes. Among them, there are quite many large values: so at average only $\frac{n}{n \cdot p} = \frac{1}{p}$ out of n nodes hold bigger values than the values heard up to time t. So we have the following contribution for function D at time t:

- from $\approx n \cdot p$ nodes that have transmitted before time t, all have now transmission time chosen uniformly at random in $[t, t + \Delta]$, so these nodes contribute value $\approx n \cdot p/\Delta$ on the interval $[t, t + \Delta]$ and 0 elsewhere,
- out of $\approx (1 - p) \cdot n$ nodes that have not transmitted up to step t, only a fraction of $\approx \frac{1}{np}$ hold values bigger than the one transmitted at step t. So the number of these nodes is $\approx \frac{1-p}{p}$ (note that the factor n cancels out!). Each of these nodes contributes a uniform density over the interval $[t, \Delta]$, hence the value $\frac{1}{\Delta - t}$ over the interval $[t, \Delta]$ and zero elsewhere. So together they contribute $\approx \frac{1-p}{p} \cdot \frac{1}{\Delta - t} = \frac{1}{p\Delta}$ to function D on the interval $[t, \Delta]$, and 0 elsewhere.
- the remaining $\approx (1 - p) \cdot n - \frac{1-p}{p}$ nodes contribute the same value $\approx [(1 - p) \cdot n - \frac{1-p}{p}]/\Delta$ to D on the interval $[t, t + \Delta]$.

So we see that on interval $[t, \Delta]$ function D has the constant value that equals approximately

$$[p \cdot n + \tfrac{1}{p} + (1 - p) \cdot n - \tfrac{1-p}{p}]/\Delta = (n + 1)/\Delta$$

As a corollary we may conclude with the following rule of thumb which indicates that the length of the period Δ has to be increased only slightly:

Rule 1. *Let Δ' be time period such that if n nodes transmit at random moments in the interval $[0, \Delta']$, then probability of collision is practically negligible. Then for executing Algorithm 3 it is enough to take $\Delta = \Delta' \cdot \frac{n+1}{n}$ in order to achieve similar resilience to transmission collisions.*

4 Time Complexity

Let us consider a situation in which a node A has the maximum value that has to be transmitted to node B and then retransmitted to node C. We compare Algorithm 3 and, to be fair, a modification of Algorithm 1 in which the value ξ is updated to the new maximum whenever a new value arrives from the neighbors of the node.

In case of Algorithm 3, the expected time needed by the maximum to reach C is $\frac{1}{2}\Delta + \frac{1}{2}\Delta = \Delta$ (the first term is the expected time needed to reach B, the second one is the expected time required to transmit from B to C). For the second algorithm this expected time equals:

$$\int_0^1 \left((1-p)\cdot(p+0.5(1-p)\Delta) + p\cdot 1.5\Delta\right) dp = \tfrac{13}{12}\Delta$$

In the above expression the variable p comes from transmission time $t = p \cdot \Delta$. For a time $p \cdot \Delta$ there are two cases: if B has not transmitted so far (it occurs with probability $1-p$), the transmission time to C is uniformly distributed in the interval $[p\Delta, \Delta]$, so the expected time is $p\Delta + 0.5(\Delta - p\Delta)$. If B has already transmitted, then B transmits the maximum in the second round and the expected time of transmission is $\Delta + 0.5\Delta$.

We conclude that Algorithm 3 reduces the execution time by half compared to Algorithm 1, and about 8% compared to its modification described above.

5 Message Complexity

Algorithm 3 defines a highly complex stochastic process. In order to explain some of its properties we consider a couple of specific graphs and provide some experimental results.

5.1 Star Graph S_n

First type of graph considered in this paper is star S_n consisting of $n+1$ nodes, where the only edges are between a distinguished node called *center* and n other nodes. Even though in typical wireless setting with a constant transmission range a star-like network with more than 5 nodes in its "arms" might enable communication between these nodes bypassing the central node, we consider this topology for arbitrary n in order to catch some universal phenomena considering communication between a node and its neighbors in a dense network.

First we consider the retransmissions from the center node. This is a potential danger for the algorithm complexity, since in the worst case the center node retransmits all incoming values: it happens if the values are transmitted in the increasing order and the center node decides to retransmit before the next value arrives. However, we show the following fact:

Fact 1. *Let n be the number of nodes and M be a random variable denoting the number of retransmissions of the central node. Then* $\mathbf{E}[M] \leq \frac{3}{2} - \frac{1}{n}$.

Proof. For simplicity of notation let us assume that $\Delta = 1$. We also assume that the center node has the smallest value - as it is the worst case where it cannot stop retransmissions from the center node. Let us consider the time interval $[0, 1]$ and assume that each node has chosen its transmission time in the interval $[0, 1]$. Let $T = t_0, \ldots, t_k$ be the longest increasing subsequence of transmission times such that the corresponding values to be transmitted at times $t_0, \ldots, t_k$, say $x_0, \ldots, x_k$ are increasing. That is, without retransmissions this would be the sequence of executed transmissions.

The probability that x_i is retransmitted by the center node equals $t_{i+1} - t_i$ (afterwards, x_i is "killed" by x_{i+1}). Obviously, the maximal value x_k is certainly going to be transmitted. Note that the expected length of the interval $[t_k, 1]$ equals $\frac{1}{2}$, as the transmission time for the maximum value x_k is chosen uniformly at random in $[0, 1]$. On the other hand, the expected length of the interval $[0, t_0]$ is the expected value of the minimum m of n independent random variables X with uniform distribution on the interval $[0, 1]$. We have:

$$\mathbf{E}[t_0] = \int_0^1 (\Pr(X > m))^{n-1}\, dm = \int_0^1 \big(1 - m\big)^{n-1}\, dm = \int_0^1 m^{n-1}\, dm = \tfrac{1}{n}.$$

So the expected number M of transmissions of the central node in S_n satisfies:

$$\mathbf{E}[M] \leq 1 + \mathbf{E}\big(\textstyle\sum_{i=0}^{k-1}(t_{i+1} - t_i)\big) = \frac{3}{2} - \frac{1}{n}\,. \qquad \square$$

Obviously the expected number of transmissions of the outside node is less than 2. In Table 1 we present some experimental results for Algorithm 3. We see an improvement compared to Algorithm 2, where the number of messages per node is 2.

Table 1. Simulations for star graph S_n: Avg.Msg.Sent is the average of the number of messages sent per node over all tests, Max.Avg.Msg.Sent concerns average value taken over the maximum number of messages sent in a single test, Max.Msg.Sent concerns the maximum number of messages sent observed over all trials and all nodes

Network Size n	Number of Tests	Avg.Msg.Sent	Max.Avg.Msg.Sent	Max.Msg.Sent
50	500	1.89	2.98	8
100	100	1.95	2.76	5
250	100	1.92	2.85	5
500	100	1.94	2.67	6
1000	100	1.89	2.86	5

5.2 Complete Graph K_n

A complete graph K_n has n nodes, where each pair of nodes are connected by an edge. For K_n, Algorithm 3 can be modified as follows:

- a node does not retransmit at all,
- a node transmits its value only if it is greater than any value transmitted so far.

Fact 2. *For complete graph K_n, the expected total number of messages sent during execution of Algorithm 3 modified as described above is* $\approx \ln n + 0.577$.

Proof. Let $t_1, \ldots, t_n$ be a sorted sequence of transmissions times chosen by the nodes and M be a random variable denoting the number of messages actually sent. The transmission at time t_i occurs if and only if all of $(i-1)$ previous values are smaller. Let p_i denote the probability of transmission at time t_i. Then $p_i = \frac{1}{i}$. So the expected number of transmissions of the i-th node is also $\frac{1}{i}$. The expected number of the total number of messages sent can be computed by summing up the expected numbers per node. Thus,

$$\mathbf{E}[M] = \sum_{i=1}^{n} \frac{1}{i} = H_n,$$

where H_n is the n-th harmonic number. Let us recall that $H_n = \ln n + \gamma + \frac{1}{2n} + \ldots$ and $\gamma = 0.57721\ldots$ is the Euler-Mascheroni constant. □

If the above modification is not implemented (i.e., nodes cannot assume the topology of a complete graph), calculations of the expected number of messages for a single node are nearly the same as for the center node of S_n with its own value. We present results of simulations for this case in Table 2. Better results for average number of messages in comparison to star graph can be noted.

Table 2. Simulations for complete graph K_n (see Table 1 for description of the table contents)

Network Size	Number of Tests	Avg.Msg.Sent	Max.Avg.Msg.Sent	Max.Msg.Sent
50	100	1.55	2.38	4
100	100	1.61	2.23	4
250	100	1.61	2.27	4
500	100	1.63	2.27	4
1000	100	1.59	2.26	5

5.3 Linear Graph L_n

L_n consists of n nodes, where node i is connected to node $i-1$ (if $i > 2$) and to node $i+1$ (if $i < n$). L_n is an important case since in many practical situations the nodes are deployed in this way (e.g. sensors along a river or a road). Let us recall the following fact that follows directly from [4]:

Fact 3. *Let M be a random variable denoting the number of all messages transmitted during execution of Algorithm 1 for L_n, where the values for the nodes are chosen uniformly at random from the interval $[0, 1]$. Then $\mathbf{E}[M] = 2 \cdot \sum_{i=1}^{n} H_n$, where H_n is the n-th harmonic number. Hence $\mathbf{E}[M] \approx 2 \ln n + 1.154$.*

Proof. The proof follows directly from the results included in [4]. □

We performed some experiments for Algorithm 3 in order to compare with Algorithm 1. The results are presented by Table 3. Fig. 1 shows that the average number of messages per node for Algorithm 3 also follows the logarithmic pattern. Moreover one can see that Algorithm 3 reduces the average number of messages per node by a factor at least 2.

Table 3. Simulations for linear graph L_n

Network Size	Number of Tests	Avg.Msg.Sent	Max.Avg.Msg.Sent	Max.Msg.Sent
20	1000	3.05	4.7	8
100	100	4.75	5.82	14
250	100	5.62	6.82	15
500	100	6.42	7.93	17
1000	60	7.09	8.60	18
2000	20	7.78	9.13	21

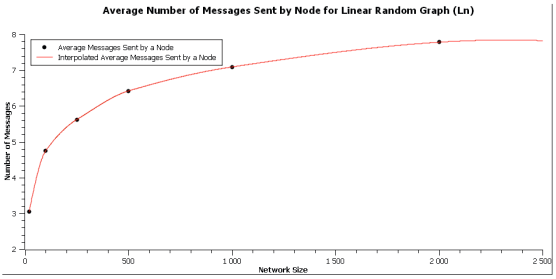


Fig. 1. Average number of messages sent per node for linear graphs

5.4 Burst Values

So far we have disregarded the fact that the values observed at a node might be correlated with its location in the network. This concerns in particular sensor networks and measurement of environment data, where gradient of changes is usually limited and the values tend to increase or decrease along each direction. Below we examine such a situation for graph L_n.

For simplicity we consider a network consisting of nodes $C_1, \ldots, C_n$, where node C_i is connected to nodes C_{i-1} (unless $i = 1$) and C_{i+1} (unless $i = n$). We assume that for $i \leq n$ node C_i stores x_i and that $x_1 > \cdots > x_n$ – which is apparently a bad case.

For this setting Algorithm 1 makes each value x_i to travel the whole way from C_i up to C_n being broadcasted $n - i + 1$ times (we assume that C_n also broadcasts, as he does not know that there is no C_{n+1}). The total number of messages transmitted is $\sum_{i=1}^{n}(n - i + 1) = \frac{n(n+1)}{2}$; the execution time is n.

The situation is very different for Algorithm 3. Let us consider the starting moment of the algorithm. Then a node C_i chooses $r_i \in [0, 1]$ at random. It broadcasts x_i provided that it does not receive x_{i-1} (or even x_{i-j} for some $j > 1$) before moment t_i. So x_i gets "killed" by x_{i-1} if $t_i > t_{i-1}$ and C_{i-1} does not receive a message from C_{i-2} before moment t_{i-1}. We compute the number of "killed" values from the set $\{x_{i-1}, x_i\}$ in this scenario. It might be 0 if $r_i < r_{i-1}$ and is at least 1 in the opposite case. As $\Pr[r_i > r_{i-1}] = r_i$ for each given r_i, the expected number of killed values from the pair x_{i-1} and x_i is at least $\int_{r=0}^{1} r \, dr = \frac{1}{2}$. By linearity of expectation we see that the expected number of killed values is at least $n/4$ already in the time period $[0, 1]$.

(In reality, the situation is better since there is an interaction between each x_i and x_{i-1}, which we have disregarded by taking $n/2$ pairs only.)

Now let us consider the case that x_i succeeded to be transmitted to C_{i+1}. We consider the chances that x_i will be killed by x_{i-1} at C_{i+1} before it gets transmitted to C_{i+1}. For simplicity of argumentation we assume that x_{i-1} has not be killed by any value x_{i-j} during the first two moves of x_{i-1}. Let x denote the time difference between transmitting x_i to C_{i+1} and of transmitting x_{i-1} to C_i. First let us observe that probability density of x is $(1-|x|)$ for $x \in [-1,1]$. So if we condition by the event that $x \geq 0$, then the density function is $2(1-x)$. Indeed, the cumulative distribution function $F(x)$ for $x \geq 0$, satisfies $F(x) = \frac{1}{2} - \frac{(1-x)(1-x)}{2} = x(2-x)$, then density is obviously the derivative of $F(x)$ and $F'(x) = 2(1-x)$. For a given x the probability that x_{i-1} "kills" x_i at C_{i+1} is the probability that $x + r'_{i+1} < r'_{i+2}$, where r'_j is the delay chosen for x_j at the second move. So have

$$\Pr[x + r'_{i+1} < r'_{i+2}] = \int_{r=0}^{1-x} (1-x+r)\, dr = \frac{(1-x)^2}{2}\,,$$

$$\Pr[x_i \text{ gets killed by } x_{i-1} \text{ at } C_{i+1}] = \int_0^1 \frac{(1-x)^2}{2} \cdot 2(1-x)\, dx = \int_0^1 x^3\, dx = \frac{1}{4}\,.$$

We see that again a substantial fractions of values are killed and will not go towards C_n.

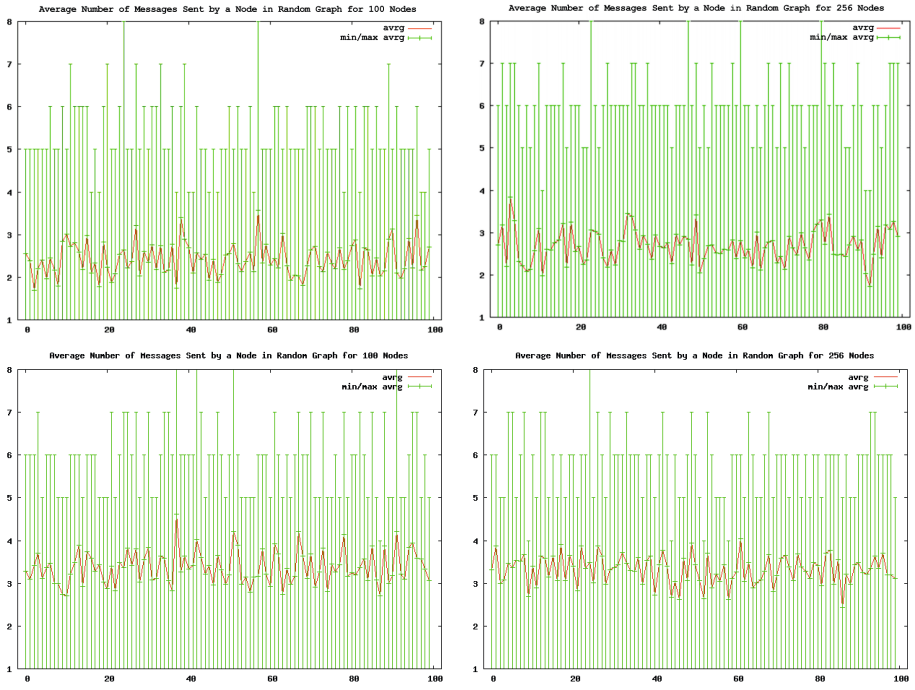


Fig. 2. Average number of messages sent by node (broken line) and minimal and maximal numbers of sent messages (vertical bars) for random graph; data gathered for 100 runs of the algorithm for Algorithm 2 (above) and Algorithm 1 (below). 100 nodes (left) and 256 nodes (right).

5.5 Simulations for Random Graphs

Now we present some results of simulations run for random graphs with Shawn simulation platform [5]. To illustrate the behavior of Algorithm 3, we run it on 100 different instances of random graphs, with the condition that they be connected i.e., there are no secluded clusters of nodes in the simulated networks. For each run, we count the number of messages sent by all nodes and derive an average number of messages per node as well as determine minimal and maximal numbers of messages sent.

The results are shown in upper part of Fig. 2. As we see, for 100 nodes the maximal number of messages varies from 4 to 8, and the average number of messages varies from 1.5 to 3.5. There is a slight increase in the number of transmissions for 256 nodes (right): the average number of messages lies in the interval $[1.6, 4]$. Moreover, the maximal number of transmissions for a single node is less or equal to 8. For comparison, we provide simulation results for Algorithm 1 run for the same settings in the bottom of Fig. 2. Our proposition shows improvement in performance seen as smaller average number of messages sent.

Conclusions and Further Research

We have shown some analytic arguments and experimental data showing that Algorithm 3 provides a substantial improvement over the straightforward implementation of Algorithm 1. Nevertheless, the stochastic process triggered by Algorithm 3 is highly complex and deserves further investigations. One of challenging questions is to determine precisely execution time even for such simple graphs as trees or graphs of small bandwidth.

References

1. Baquero, C., Almeida, P.S., Menezes, R.: Fast estimation of aggregates in unstructured networks. In: Calinescu, R., Liberal, F., Marín, M., Herrero, L.P., Turro, C., Popescu, M. (eds.) ICAS, pp. 88–93. IEEE Computer Society (2009)
2. Cai, Z., Lu, M., Wang, X.: Distributed initialization algorithms for single-hop ad hoc networks with minislotted carrier sensing. IEEE Trans. Parallel Distrib. Syst. 14(5), 516–528 (2003)
3. Cichoń, J., Kutyłowski, M., Zawada, M.: Power of discrete nonuniformity - optimizing access to shared radio channel in ad hoc networks. In: MSN, pp. 9–15. IEEE Computer Society (2008)
4. Cichoń, J., Lemiesz, J., Zawada, M.: On Message Complexity of Extrema Propagation Techniques. In: Li, X.-Y., Papavassiliou, S., Ruehrup, S. (eds.) ADHOC-NOW 2012. LNCS, vol. 7363, pp. 1–13. Springer, Heidelberg (2012)
5. Kroller, A., Pfisterer, D., Buschmann, C., Fekete, S.P., Fischer, S.: Shawn: A new approach to simulating wireless sensor networks (2005)
6. Mosk-Aoyama, D., Shah, D.: Computing separable functions via gossip. In: Ruppert, E., Malkhi, D. (eds.) PODC, pp. 113–122. ACM (2006)

A Model for the Performance Analysis of SPL-OBS Core Nodes with Deflection Routing

Dang Thanh Chuong[1], Vu Duy Loi[2], and Vo Viet Minh Nhat[1]

[1] HueUniversity
[2] Center of Information Technology, National Office of Communist Party - Viet Nam
{Dtchuong,vominhnhat}@gmail.com, vdloi@vptw.dcs.vn

Abstract. In optical burst switching networks, techniques like wavelength conversion, optical buffer or deflection routing are often applied to resolve a contention problem that may cause data loss. Instead of the planned output port, the contended burst is sent to a new output port, on a new path to its destination by the deflection routing. In the case of wavelength conversion, the arriving burst is conveyed on a new available wavelength, but only partial wavelength conversion is available due to the technology constraint. This article considers a model for the performance evaluation of OBS core nodes with the Share-Per-Link(SPL) architecture where partial wavelength converters are distributed at each output port. A continuous-time Markov chain model is proposed to analyze the performance of OBS core nodes operated with the deflection routing rule.

Keywords: OBS node, Blocking Probability, Deflection Probability, continuous-time Markov chain, state transition rate matrix.

1 Introduction

Optical Burst Switching (OBS) in Wavelength Division Multiplexing (WDM) networks has been seen as a promising solution for the next generation of Internet because of huge potential bandwidth [1]. An important issue in OBS networks is to reduce the loss of bursts [2]. A contention will occur when two arriving bursts in an OBS core node simultaneously require the same wavelength on the same output port. The burst contention can be solved by wavelength conversion, optical buffer (i.e. Fiber Delay Lines) or routing deflection.

The wavelength conversion, which allows the change of wavelength of an arriving burst on after OBS nodes, can significantly improve the performance of OBS networks. However, the commercial production cost of wavelength converters is still very expensive due to the technology reason. As a consequence, the partial wavelength conversion or limited-range wavelength conversion are considered as a realizable contention resolution.

This article considers a model of OBS core nodes with the Share-Per-Link (SPL) architecture where which wavelength converters are partially (instead of completely) distributed at each output port, and supporting deflection routing that send the

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 152–161, 2012.

contended burst to a new output port, on a new path to its destination. We proposed the application of a continuous-time Markov chain to evaluate the performance of OBS core nodes. We use our model to study the dependency of the blocking probability on the traffic load density, the amount of used wavelengths and wavelength converters; and the variant abilities of deflection.

The rest of this paper is organized as follows: Section 2 gives an overview concerning the SPL architecture of OBS core nodes that support deflection routing; Section Part 3, a model of performance analysis of OBS core nodes with the SPL architecture (in which wavelength converters are partially distributed at each output port) and supporting deflection routing will be considered; The method of calculating the steady-state probability, the factor that determines the contention probability, based on state transition rate matrix will be presented in part 4; the results and analysis will be mentioned in Section 5; Finally, the paper is concluded in Section 6.

2 The SPL Architecture of OBS Cores Nodes Supporting Deflection Routing

There are two principal ways to arrange wavelength converters in an OBS core node as follows.

- The architecture of Share-Per-Node (SPN): wavelength converters are shared among all input/output ports;
- The architecture of Share-Per-Link (SPL): an output port is equipped with some wavelength converters.

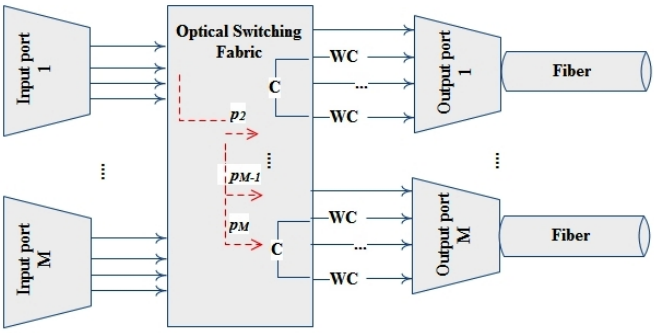


Fig. 1. An SPL-OBS core node supporting deflection routing

In a partial wavelength conversion applied in the SPL architecture only a limited amount of wavelength converters share among the channels on an output port [3]. Although it is not as effective as fully equipped, the partial wavelength conversion reduces the contention significantly [3] and also decreases the cost of equipping wavelength converters. An analysis for a blocking probability in an OBS core node with

partial wavelength conversion based on continuous-time Markov chain has been performed in [2]. In this article, we consider the case of partial wavelength conversion in the SPL architecture with deflection routing (i.e. considering the case of multi-output ports).

The SPL architecture of an OBS core node with multi-output ports is described in Figure 1. When a contention happens at the planned port, an arriving burst is deflected to a new output port.

3 A Model of Performance Analysis of SPL-OBS Core Nodes Supporting Deflection Routing

Notations are introduced as follows.

i. An OBS node has multi-input/output ports; each corresponds to an optical fiber carrying ω wavelengths;

ii. There are $C(C < \omega)$ full wavelength converters equipped at an output port.

iii. Inter-arrival time of bursts is exponentially distribution with the parameter γ and the process of burst service is exponentially distributed with the average rate of $1/\mu$ (μ is the average length of burst); the traffic load is $\rho = \gamma/\mu$.

iv. The wavelength of an arriving burst can or cannot be converted depending on the availability of the expected wavelength and/or of wavelength converters. A burst will be dropped only if all wavelengths are busy or no wavelength converter is available.

v. The bursts which arrive at an OBS core node are deflected to all its output ports with the same probability (p_k). Therefore, as described in Figure1, we need consider the deflection probability of one of $M-1$ remaining output ports, because $p_2 = \ldots = p_{M-1} = p_M$ and $\sum_{k=1}^{M}\{p_k | k \neq i\} = p$, where M is the number of output ports and p is the total of deflection probabilities.

In [3][6], a model for computing the blocking probability at an OBS core node of the SPL architecture with only one output port (which does not support deflection routing) was considered. Each state of this model corresponds to the pair (w, c), where $0 \leq c \leq w \leq \omega$ and $c \leq C < \omega$ are the number of used wavelengths and wavelength converters, respectively. The number of states of this model is:

$$n_S = \frac{(2\omega - C + 2)(C + 1)}{2} \tag{1}$$

This article will analyze the performance of SPL-OBS core nodes that support deflection routing. We consider SPL-OBS core nodes with two output ports (called port1 and port 2), each has ω wavelengths and $C(C \leq \omega)$ wavelength converters.

In this case, the state transition diagram does not only depend on the number of wavelengths, the number of wavelength converters at each output port, but also the number of output ports that the bursts can deflect to. The state transition diagram of a 4-dimentional Markov chain [7] $(w_1, c_1; w_2, c_2)$ is illustrated in Figure 2.

Considering the system in state$(w_1, c_1; w_2, c_2)$ at time t and an arriving burst in interval $(t, t + \delta)$, a state transition may be as follows:

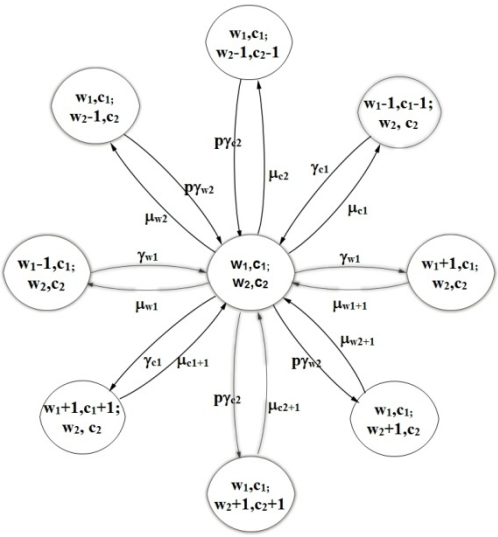


Fig. 2. State transition diagram of quart (w_1,c_1;w_2,c_2)

(1). The wavelength required by the arriving burst is available on the output port (port 1 or port 2), the arriving burst does not need to use any wavelength converter; the new state will be $(w_1 + 1, c_1; w_2, c_2)$ where $w_1 + 1 \leq \omega$, or $(w_1, c_1; w_2 + 1, c_2)$, where $w_2 + 1 \leq \omega$ and the state transition rate will be $\gamma_{w1} = (\omega - w_1) \cdot \gamma/\omega$ for port 1 or $\gamma_{w_2} = (\omega - w_2) \cdot \gamma/\omega$ for port 2.

(2). The required wavelength was used at the output port (port 1 or port 2), a wavelength converter is used to schedule the arriving burst on a new available wavelength (chosen randomly); the new state will be $(w_1 + 1, c_1 + 1; w_2, c_2)$ where $w_1 + 1 \leq \omega$ and $c_1 + 1 \leq C$, or $(w_1, c_1; w_2 + 1, c_2 + 1)$ where $w_2 + 1 \leq \omega$ and $c_2 + 1 \leq C$ and state transition rate will be $\gamma'_{w1} = w_1 \cdot \gamma/\omega$ for port 1 or $\gamma'_{w_2} = w_2 \cdot \gamma/\omega$ for port 2.

(3). The arriving burst is deflected to a new output port (with the probability p) in the following cases:

- All of the wavelengths at the output port (port 1 or port 2) are busy (i.e. $w_1 = \omega$ or $w_2 = \omega$), at this time the burst will be deflected from port 1 to port 2 with the state transition rate of $(p \cdot \gamma_{w1} + \gamma_{w2})$ or from port 2 to port 1 with the state transition rate of $(p \cdot \gamma_{w_2} + \gamma_{w_1})$.

- All of the wavelengths at the output port (port 1 or port 2) are busy and the required wavelength at the deflected port is busy, a wavelength converter at the deflected port is used to schedule the arriving burst on a new available wavelength (chosen randomly); the state transition rate will be $(p \cdot \gamma'_{w_1} + \gamma'_{w_2})$ with the deflection from port 1 to port 2, or$(p \cdot \gamma'_{w_2} + \gamma'_{w_1})$ with the deflection from port 2 to port 1.

(4). State transition diagram will not change if:

- all wavelengths are busy at 2 ports, i.e. in state $(\omega, c_1; \omega, c_2)$, where $c_1 \leq C$ and $c_2 \leq C$; or

- the required wavelength is busy at 2 ports and no wavelength converter is available, i.e. $c_1 = c_2 = C$. This means the current state is $(w_1, C; w_2, C)$.

Similarly, if the system is in the state $(w_1, c_1; w_2, c_2)$ and a burst is served in interval $(t, t + \delta)$, state transition may be as follows:

(5). $(w_1 - 1, c_1; w_2, c_2)$ where $w_1 - 1 \geq 0$ and $w_1 - 1 \geq c_1$; or $(w_1, c_1; w_2 - 1, c_2)$, where $w_2 - 1 \geq 0$ and $w_2 - 1 \geq c_2$ if no wavelength converter is used for this burst before. The state transition rate will be $\mu_{w_1} = (w_1 - c_1)\mu$ or $\mu_{w_2} = (w_2 - c_2)\mu$.

(6). $(w_1 - 1, c_1 - 1; w_2, c_2)$ where $w_1 - 1 \geq 0, c_1 - 1 \geq 0$ and $w_1 \geq c_1$, or $(w_1, c_1; w_2 - 1, c_2 - 1)$ where $w_2 - 1 \geq 0, c_2 - 1 \geq 0$ and $w_2 \geq c_2$, if a wavelength converter is used for this burst before. The state transition rate will be $\mu'_{w_1} = c_1\mu$ or $\mu'_{w_2} = c_2\mu$.

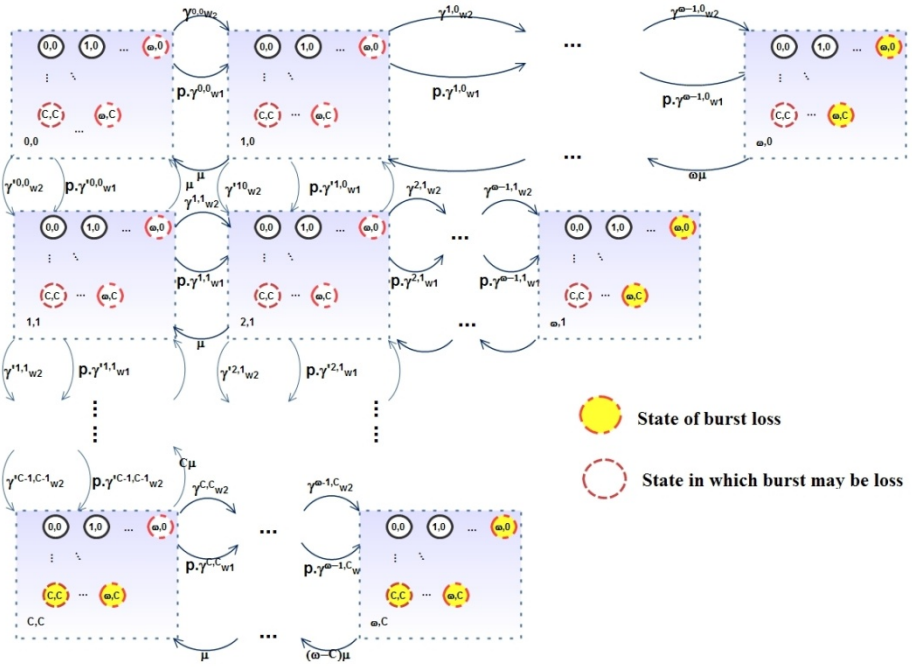


Fig. 3. The diagram of all state transitions

Figure 3 plots the overall of state transition diagram, where states are grouped. Note that the diagram of a group of states (w_2, c_2) in Figure 3 is equivalent to a 2-dimensional diagram in [3].

In addition to the transitions inside of each group, there are the transitions between groups. For example, there are n_S transitions from group (0,0) to group (1,0): $(0,0;0,0) \rightarrow (0,0;1,0)$; $(1,0;0,0) \rightarrow (1,0;1,0)$; …; $(w_1, c_1; 0,0) \rightarrow (w_1, c_1; 1,0)$ and $(\omega, C; 0,0) \rightarrow (\omega, C; 1,0)$. Similarly, there are n_S transitions from group (0,0) to group (1,1): $(0,0;0,0) \rightarrow (0,0;1,1)$; $(1,0;1,1) \rightarrow (1,0;1,1)$;…; $(w_1, c_1; 0,0) \rightarrow (w_1, c_1; 1,1)$

and $(\omega, C; 0,0) \rightarrow (\omega, C; 1,1)$. Thus, for each group (w_1, c_1), we will have n_S states and n_S state transitions between groups, so we have $(n_S * n_S)$ states on the diagram of Figure 3.

The balance equation can be written as

$$\begin{aligned}(\gamma_{w_1} + \gamma_{w_2} + \mu_{w_1} + \mu_{w_2} + \gamma_{c_1} + \gamma_{c_2} + \mu_{c_1} + \mu_{c_2})\pi_{w_1,c_1;w_2,c_2} \\ = \gamma_{w_1-1}\pi_{w_1-1,c_1;w_2,c_2} + \mu_{w_1+1}\pi_{w_1+1,c_1;w_2,c_2} \\ + \gamma_{c_1-1}\pi_{w_1-1,c_1-1;w_2,c_2} + \gamma_{c_1+1}\pi_{w_1+1,c_1+1;w_2,c_2} \\ + \gamma_{c_2-1}\pi_{w_1,c_1;w_2-1,c_2-1} + \mu_{c_2}\pi_{w_1,c_1;w_2+1,c_2+1} \\ + \gamma_{w_2-1}\pi_{w_1,c_1;w_2-1,c_2} + \mu_{w_2+1}\pi_{w_1,c_1;w_2+1,c_2}\end{aligned} \quad (2)$$

where $\sum_{w_1,c_1;w_2,c_2} \pi_{w_1,c_1;w_2,c_2} = 1$

Based on the transition rules in Figure 2 and the state transition diagram in Figure 3, the burst contention will occur in the following cases:

- *Contention due to the lack of available wavelengths*: when all wavelengths are busy at 2 output ports (corresponding with state $(\omega, c_1;\ \omega, c_2)$, where $c_1 \leq C$ and $c_2 \leq C$.
- *Contention due to the lack of available wavelength converters*: when the required wavelength for an arriving burst is busy and no wavelength converter is available at 2 output ports (corresponding with state $(w_1, C; w_2, C)$ where $C \leq w_1 \leq \omega$ and $C \leq w_2 \leq \omega$.
- *Contention caused by the deflection probability*p: the possibility of deflection from port 1 to port 2 or from port 2 to port 1. This case corresponds with state $(\omega, c_1; w_2, c_2)$, where $0 \leq c_1 \leq C$, or $(w_1, C; w_2, c_2)$, where $C \leq w_1 \leq \omega$, or $(w_1, c_1; \omega, c_2)$, where $0 \leq c_2 \leq C$, or $(w_1, c_1; w_2, C)$, where $C \leq w_2 \leq \omega$, without deflection (with the probability $(1 - p)$).

Because of the contention of each above mentioned case are independent, the blocking probability (denoted as PB) of all OBS core node is as follows:

$$\begin{aligned}PB = \sum_{c_2=0}^{C}\sum_{c_1=0}^{C}\pi_{\omega,c_1;\omega,c_2} + \sum_{w_2=C}^{\omega}\frac{w_2}{\omega}\left(\sum_{w_1=C}^{\omega-1}\frac{w_1}{\omega}\pi_{w_1,C;w_2,C}\right) \\ + (1-p)\sum_{c_2=0}^{C}\sum_{w_2=c_2}^{\omega-1}\sum_{c_1=0}^{C}\pi_{\omega,c_1;w_2,c_2} \\ + (1-p)\sum_{c_2=0}^{C-1}\sum_{w_2=c_2}^{\omega}\sum_{w_1=C}^{\omega-1}\pi_{w_1,C;w_2,c_2}\end{aligned} \quad (3)$$

To determine the values PB in equation (3), we must calculate the steady-state probability $\pi_{w_1,c_1;w_2,c_2}$ by solving equation (2).

4 The Computation of the Steady State Probabilities

State transition rate matrix Q is built as follows:

Input: The space of states (diagram in Figure 3).
Step 1. Building state transition rate matrix Q (Table 1):

Table 1. General matrix Q

QQJ_0	B_2^0			
C_2^1	QQJ_1	B_2^1		
	C_2^2	QQJ_2	…	
		…	…	B_2^j
			C_2^j	QQJ_C

Table 2. Matrices $QQJ_j (j = 0 \dots C)$

QQ	A_2^j			
$A_2'^{j+1}$	QQ	A_2^{j+1}		
	$A_2'^{j+2}$	QQ	…	
		..	…	$A_2^{\omega-1}$
			$A_2'^{\omega}$	QQ

Step 1.1. Building matrices QQJ_j $\left((\omega + 1 - j) * n_S\right)$ x $((\omega + 1 - j) * n_S)$, $0 \le j \le C$ (Table 2).

Step 1.1.1. Given $\boldsymbol{A_2^j, A_2^{j+1}, \dots, A_2^{\omega-1}}$ are state transition matrices of n_S x n_S at port 2, which has the state transition from $(w_1, c_1; w_2, c_2)$ to $(w_1, c_1; w_2 + 1, c_2)$; $(0 \le c_1 \le C, 0 \le w_1 \le \omega; 0 \le c_2 \le j, j \le w_2 \le \omega - 1)$ and the state transition rate $\gamma_{w_2} = (\omega - w_2)\,\gamma/\omega$ or $p \cdot \gamma_{w_1}$ (when $w_1 = \omega$).

Step 1.1.2. Given $\boldsymbol{A_2'^{j+1}, A_2'^{j+2}, \dots, A_2'^{\omega}}$ are the state transition matrices of n_S x n_S at port 2, which has the state transition from $(w_1, c_1; w_2, c_2)$ to $(w_1, c_1; w_2 - 1, c_2)$, $(0 \le c_1 \le C, 0 \le w_1 \le \omega; 0 \le c_2 \le j, j + 1 \le w_2 \le \omega)$, and the state transition rate $\mu_{w_2} = (w_2 - c_2)\mu$.

Step 1.1.3. The matrix $\boldsymbol{QQ}$ $(n_S$ x $n_S)$ is formed from the matrices A_1^k, B_1^k, C_1^k corresponding with the transitions at port 1 corresponding with each state of port 2 (Table 3):

$A_1^k(m, n)$: determine the state transition rate from $(m, k; w_2, c_2)$ to $(n, k; w_2, c_2)$ where $k \le m, n \le \omega$; $0 \le k \le C$. The size of matrix $A_1^k(m, n)$ is $(\omega + 1 - k)$ x $(\omega + 1 - k)$. The non-zero elements of matrix $A_1^k(m, n)$ are calculated as follows:

Table 3. Matrices QQ $(0 \le j \le C)$ of

A_1^0	B_1^0			
C_1^1	A_1^1	B_1^1		
	C_1^2	A_1^2	…	
		…	…	B_1^k
			C_1^k	A_1^k

Table 4. Matrices $B_2^j (0 \le j \le C - 1)$

BB_2^0				
	BB_2^1			
		BB_2^2		
			…	
				BB_2^k

$A_1^k(m,m-1) = k \cdot \mu$ where $1 \le k \le (\omega - j)$ and $A_1^k(m,m+1) = \gamma \cdot (1 - (m+k)/\omega)$ where $0 \le m \le (\omega - 1 - k)$.

$B_1^k(m,n)$: determine the state transition rate from $(m,k;w_2,c_2)$ to $(n,k+1;w_2,c_2)$ where $k \le m \le \omega;\ k+1 \le n \le \omega;\ 0 \le k \le C-1$. The size of matrix $B_1^k(m,n)$ is $(\omega + 1 - k)$ x $(\omega - k)$. The non-zero elements of matrix $B_1^k(m,n)$ are calculated as follows: $B_1^k(m,m) = (m+k) \cdot \gamma/\omega$ where $0 \le m \le (\omega - 1 - k)$.

$C_1^k(m,n)$: determine the state transition rate from $(m,k;w_2,c_2)$ to $(n,k-1;w_2,c_2)$ where $k \le m \le \omega;\ (k-1) \le n \le \omega;\ 1 \le k \le C$. The size of matrix $C_1^k(m,n)$ is $(\omega + 1 - k)$ x $(\omega + 2 - k)$. The non-zero elements of matrix $C_1^k(m,n)$ are calculated as follows: $C_1^k(m,n) = k \cdot \mu$ where $0 \le m \le (\omega - k)$.

Step 1.2. Matrices $\boldsymbol{B_2^j}$ $(0 \le j \le C-1)$ determine the state transition from $(w_1,c_1;w_2,c_2)$ to $(w_1,c_1;w_2+1,c_2+1)$ for each state (w_1,c_1), where $0 \le w_2 \le (\omega - 1), 0 \le c_2 \le C$ (Table 4). The size of matrices B_2^j is $((\omega + 1 - j) * n_S$ x $(\omega - j) * n_S)$ and their state transition rate is $\gamma'_{w_2} = w_2\,\gamma/\omega$ or $p \cdot \gamma'_{w_1}$ (when $w_1 = \omega$).

Matrices BB_2^k $(0 \le k \le \omega - 1)$ of n_S x n_S are determined as follows: $BB_2^k(m,m) = k \cdot \gamma/\omega$ where $0 \le m \le (n_S - 1)$; The remaining elements are 0.

Step 1.3. Matrices $\boldsymbol{C_2^j}$ $(1 \le j \le C)$ determine the state transition rate from $(w_1,c_1;w_2,c_2)$ to $(w_1,c_1;w_2-1,c_2-1)$ for each state (w_1,c_1), where $1 \le w_2 \le \omega, 1 \le c_2 \le C$. The size of matrices $\boldsymbol{C_2^j}(1 \le j \le C)$ is $(\omega + 1 - j) * n_S$ x $(\omega + 2 - j) * n_S$ and the state transition rate is $\mu'_{w_2} = c_2\mu$.

Matrices C_2^j are determined as follows: $C_2^j(m,m) = j \cdot \mu$ where $0 \le m \le (\omega + 1 - j) * n_S$; The remaining elements are 0.

Step 2: Calculate the values on diagonal of the matrix Q:

$$Q(i,i) = -\sum_{i,j=0,i\neq j}^{n_S * n_S - 1} Q_{i,j} \text{ where } \sum_{i,j=0}^{n_S * n_S - 1} Q_{i,j} = 0$$

Output: Matrix Q with size $(n_S * n_S)$ x $(n_S * n_S)$.

Because the structure of Q, a method [4][8][9] can be used to compute the steady state probabilities.

5 Results and Analysis

In what follows, we will investigate the dependency of the blocking probability of OBS core node on traffic load (ρ), the number of wavelengths (ω) and the number of wavelength converters (C).

Figure 4 describes the variation of blocking probabilities when increasing traffic load: obviously the more traffic load the more blocking probability. In case of allowing deflection with the variant levels, corresponding to the deflection probabilities: $p = 0$ (no deflection), $p = 0.5$ and 0.7 (allowing the deflection with probabilities 0.5 and 0.7) and $p = 1$ (full deflection), Figure 4 shows that the blocking probability decreases significantly when increasing the deflection probability. When $p = 0$, the

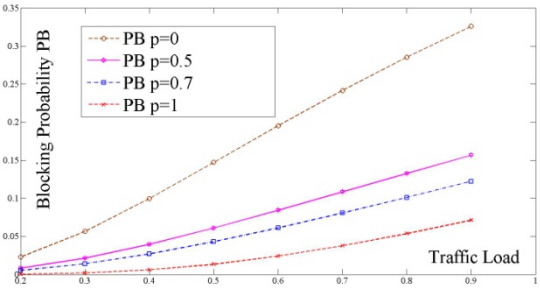


Fig. 4. The blocking probabilities with ω = 3, C = 2, p = [0,0.5,0.7,1] vs. β

results in Figure 4 is completely coincide with the results in [3], in which only one output port is considered (no deflection occurs).

If we increase the amount of used wavelength converters, the blocking at each port reduces significantly. Figure 5 illustrates the dependency of the blocking probability on the number of converters. Furthermore, the increase of the deflection probability decreases the blocking probability.

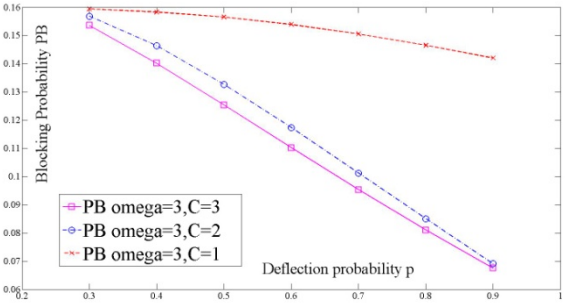


Fig. 5. Blocking probabilities with ω=3, C=1,2,3 vs β=0.8

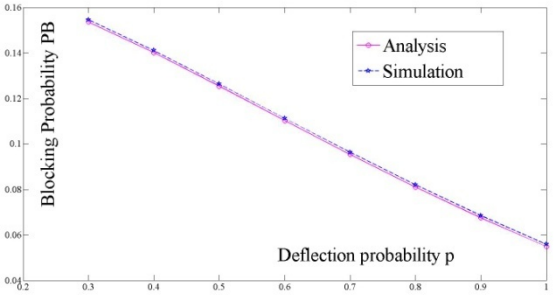


Fig. 6. Blocking probabilities in the analytical case and simulation

We also implement a special case of simulation on NS-2, with full wavelength conversion ($C = \omega$) and high traffic load $\beta = 0.8$, in order to compare analytical results with simulation. Figure 6 shows that there is a good match between the analysis results and simulation.

6 Conclusion

This article considers a model of OBS core nodes with the SPL architecture, multi-output ports, and supporting deflection routing. We proposed a Markov queuing model to evaluate the performance of an OBS core node with multiple constraints such as multi-output ports, the amount of used wavelengths and wavelength converters, and the variant abilities of deflection. Numerical results show that the proposed continuous-time Markov chain can be efficiently used to compute the blocking probabilities. The numerical results illustrate that the blocking probability strongly depends on the traffic load density, the amount of used wavelengths and wavelength converters; and the variant abilities of deflection. However, the case in which an arriving burst has its own deflection probability is not considered yet. This issue will be handled in our future research plan.

Acknowledgments. We deeply thank to Dr. Do Van Tien, Professor of Budapest University of Technology& Economics, Hungary, who has provided invaluable comments and advices.

References

1. Chen, Y., Qiao, C., Yu, X.: Optical Burst switching: a new area in optical networking research. IEEE Network 18(3), 16–23 (2004)
2. Xu, Y., Shi, K.-Y., Fan, G.: Performance analysis of optical burst witching node with limited wavelength conversion capabilities. Springer (2009)
3. Li, H., Thng, I.L.-J.: Performance analysis of a Limited Number of Wavelength Converters in an Optical Switching Node. IEEE Photonics Technology Letters 17(5) (May 2005)
4. Do, T.V.: Comparison of Allocation Schemes for Virtual Machines in Energy-aware Server Farms. Computer Journal 54(11), 1790–1797 (2011)
5. Akimaru, H., Kawashima, K.: Teletraffic: Theory and Applications, pp. 71–104. Springer, Berlin (1993)
6. Reviriego, P., Guidotti, A.M., Raffaelli, C., Aracil, J.: Blocking of optical burst Switches with share wavelength converters: exact formulation and analytical approximations. Photon Netw. Commun. (2008)
7. Wu, H., Qiao, C.: Modeling iCAR via Multi-Dimensional Markov Chains. Mobile Networks and Application 8, 295–306 (2003)
8. Van Do, T., Rotter, C.: Comparison of Scheduling Schemes for On-Demand IaaS Requests. Journal of Systems and Software 85, 1400–1408 (2012)
9. Van Do, T., Do, N.H., Chakka, R.: A New Queueing Model for Spectrum Renting in Mobile Cellular Networks. Computer Communications 35, 1165–1171 (2012)

Ordering of Potential Collaboration Options

Sylvia Encheva

Stord/Haugesund University College, Bjørnsonsg. 45, 5528 Haugesund, Norway
sbe@hsh.no

Abstract. This work focuses on investigating the importance of evaluating the likelihood that a particular collaboration option is going to be profitable for a firm. Some collaboration options for a firm are first evaluated by experts on previously agreed upon criteria. Methods from rough set theory are afterwords employed for ordering of the agreed upon criteria with respect to their significance in predicting collaboration outcomes.

Keywords: Collaboration, assessment, rough sets.

1 Introduction

Most firms have to consider new business strategies due to rapidly changing international markets. Strategic alliances can join resources and thus improve on products qualities and at the same time shorten delivery time. Preliminary assessment of collaboration options is often required in order to avoid significant loses. Market considerations imply that early entry into large, growing markets is more likely to lead to success [19].

This work focuses on investigating the importance of evaluating the likelihood that a collaboration option is going to be profitable. Collaboration options for a particular firm are evaluated by experts on previously agreed upon criteria. Methods from rough set theory are employed for ordering of chosen criteria. Collaboration is understood as a generic, cooperative interaction between firms to achieve some agreed upon objectives. Theory and application of collaboration cost-benefit analysis are presented in [16] and [17]. Business decisions on projects with potential positive rate of return are discussed in [14] and the human side of a decision making process is considered in [15].

The rest of the paper is organized as follows. Related work and supporting theory may be found in Section 2. The evaluation model in Section 3. The paper ends with a conclusion in Section 4.

2 Related Work

Inspired by the Aristotle writing on propositions about the future - namely those about events that are not already predetermined, Lukasiewicz has devised a three-valued calculus whose third value, $\frac{1}{2}$, is attached to propositions referring

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 162–171, 2012.

to future contingencies [10]. The third truth value can be construed as 'intermediate' or 'neutral' or 'indeterminate' [18].

Another three-valued logic, known as Kleene's logic is developed in [9] and has three truth values, truth, unknown and false, where unknown indicates a state of partial vagueness. These truth values represent the states of a world that does not change.

A brief overview of a six-valued logic, which is a generalized Kleene's logic, has been first presented in [11]. The six-valued logic was described in more detail in [7]. In [4] this logic is further developed by assigning probability estimates to formulas instead of non-classical truth values.

Extensions of Belnap's logic [1] are discussed in [5] and [8].

Two kinds of negation, weak and strong negation are discussed in [20]. Weak negation or negation-as-failure refers to cases when it cannot be proved that a sentence is true. Strong negation or constructable falsity is used when the falsity of a sentence is directly established.

The six-valued logic distinguishes two types of unknown knowledge values - permanently or eternally unknown value and a value representing current lack of knowledge about a state [6].

Let P be a non-empty ordered set. If $sup\{x, y\}$ and $inf\{x, y\}$ exist for all $x, y \in P$, then P is called a *lattice* [2]. In a lattice illustrating partial ordering of knowledge values, the logical conjunction is identified with the meet operation and the logical disjunction with the join operation.

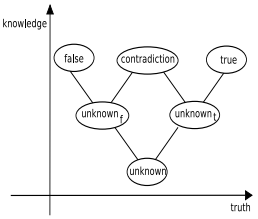


Fig. 1. Knowledge lattice

A lattice [2] showing a partial ordering of the elements f, $\bot_f$, $\top$, $\bot_t$, $\top$, t by degree of knowledge is presented in Fig. 1. The knowledge lattice illustrates how the truth value of a formula that has a temporary truth value can be changed as more knowledge becomes available. Suppose a sentence has a truth value $\bot_f$ at one point of time and f at another. Its truth value is then determined as f, i.e. the system allows belief revision as long as the revision takes place in an incremental knowledge fashion.

Below is a truth table for the six-valued logic as shown in [6].

Rough Sets were originally introduced in [12]. The presented approach provides exact mathematical formulation of the concept of approximative (rough)

Table 1. Truth table for the six-valued logic

$\wedge$	t	f	$\top$	$\bot_t$	$\bot_f$	$\bot$
t	t	f	$\top$	$\bot_t$	$\bot_f$	$\bot$
f	f	f	f	f	f	f
$\top$	$\top$	f	$\top$	$\top$	$\bot_f$	$\bot_f$
$\bot_t$	$\bot_t$	f	$\top$	$\bot_t$	$\bot_f$	$\bot$
$\bot_f$	$\bot_f$	f	$\bot_f$	$\bot_f$	$\bot_f$	$\bot_f$
$\bot$	$\bot$	f	$\bot_f$	$\bot$	$\bot_f$	$\bot$

equality of sets in a given approximation space. An *approximation space* is a pair $A = (U, \theta)$, where U is a set called universe, and $\theta \subset U \times U$ is an indiscernibility relation.

Equivalence classes of θ are called *elementary sets* (atoms) in A. The equivalence class of θ determined by an element $x \in U$ is denoted by $\theta(x)$. Equivalence classes of θ are called *granules* generated by θ. The following definitions are often used while describing a rough set $X, X \subset U$:

- the θ-*upper approximation* of X, $\theta^{\star}(x) := \bigcup_{x \in U}\{\theta(x) : \theta(x) \cap X \neq \varnothing\}$
- the θ-*lower approximation* of X, $\theta_{\star}(x) := \bigcup_{x \in U}\{\theta(x) : \theta(x) \subseteq X\}$
- the θ-*boundary region* of X, $RN_{\theta}(X) := \theta^{\star}(X) - \theta_{\star}(X)$

In the rough set theory [13], objects are described by either physical observations or measurements. Consider an information system $\mathcal{A} = (U, A)$ where information about an object $x \in U$ is given by means of some attributes from A, i.e., an object x can be identified with the so-called signature of $x : Inf(x) = a(x) : a \in A$.

The θ-positive region of X with respect to the relation θ is $POS_{\theta}(X) = \underline{\theta}X$ The θ-negative region of X with respect to the relation θ is the set $NEG_{\theta}(X) = U - \overline{\theta}X$ The θ-boundary region of X with respect to the relation θ is the set $BN_{\theta}(X) = \overline{\theta}X - \underline{\theta}X$.

The approximation quality function $\gamma : 2^U \rightarrow [0, 1]$ in the approximation space (U, θ) is defined as

$$\gamma(X) = \frac{|X| + |U \backslash X|}{|U|}$$

where $X \subseteq U$.

The approximation quality of Q with respect to d, is defined as

$$\gamma(Q \rightarrow d) = \frac{|\cup\{X \in \mathcal{P}(Q) : X \ \text{ is } \ d - \text{deterministic}\}|}{|U|}$$

where the partition induced by θ is denoted by $\mathcal{P}(Q)$.

Significance testing [3]

Suppose that we want to test the statistical significance of the rule $Q \to d$. Let Σ be the set of all permutations of U. For each $\sigma \in \Sigma$, we define a new set of feature vectors $\overline{x}_\sigma^\Omega$ by

$$x_\sigma^r \stackrel{def}{=} \begin{cases} \sigma(x)^d, & \text{if } r = d, \\ x^r, & \text{otherwise.} \end{cases}$$

In this way, we permute the x^d values according to σ, while leaving everything else constant. The resulting rule system is denoted by $Q \to \sigma(d)$. We now use the permutation distribution $\{\gamma(Q \to \sigma(d)) : \sigma \in \Sigma\}$ to evaluate the strength of the prediction $Q \to d$. The value $p(\gamma(Q \to d)|H_0)$ measures the extremeness of the observed approximation quality and it is defined by

$$p(\gamma(Q \to d)|H_0) := \frac{|\{\sigma \in \Sigma : \gamma(Q \to \sigma(d)) \geq \gamma(Q \to d)\}|}{|U|!}$$

If $\alpha = p(\gamma(Q \to d)|H_0)$ is low, traditionally below 5%, we reject the null hypothesis, and call the rule significant, otherwise, we call it casual.

3 Collaboration Options

We propose employment of a decision support system for selecting the most desirable collaboration options. In our scenario all collaboration options for a particular firm are evaluated by three experts with respect to two criteria, importance and relevance. Importance refers to the degree of interest a collaboration option has regarding the firm's current goals and relevance refers to evaluation of the amount of resources a collaboration option requires considering results and organizational constrains.

In Subsection 3.1 we group experts' recommendations with respects to a single criterion based on rough set approximations. This can be further used for automated selection of desired outcomes. In Subsection 3.2 approximation qualities and significance of a single criterion along with combination of criteria for predicting of respective outcome are presented.

3.1 Triplets Related to a Single Criterion

To each criterion an expert can assign exactly one of the following recommendations -

- recommended (r)
- adequate (a)
- inadequate (i)

The experts' recommendations are not graded, i.e. they carry equal weight and their order does not effect the decision process. The response combination triplets per criterion where all experts have delivered their recommendations are

- recommended, recommended, recommended, abbreviated (rrr)
- recommended, recommended, adequate, abbreviated (rra)
- recommended, recommended, inadequate, abbreviated (rri)
- recommended, adequate, adequate, abbreviated (raa)
- recommended, adequate, inadequate, abbreviated (rai)
- recommended, inadequate, inadequate, abbreviated (rii)
- adequate, adequate, adequate, abbreviated (aaa)
- adequate, adequate, inadequate, abbreviated (aai)
- adequate, inadequate, inadequate, abbreviated (aii)
- inadequate, inadequate, inadequate, abbreviated (iii)

In real life situations it is always possible to have a case with a missing response. Such an occurrence is denoted by n. However it is important to keep in mind that including a missing response in the possible response combinations considerably increases the amount of triplets to work with.

We first group all response triples in four rough sets approximations as in Fig. 2, Fig. 4, Fig. 3, and Fig. 5 with respect to the number of recommendations and the level of consistency of each answer combination. The four granules in lower approximations are 'homogeneous' (three responses of the same type, f. ex. rrr) and have no common elements. The rest of the granules in lower approximations have two responses of the same type.

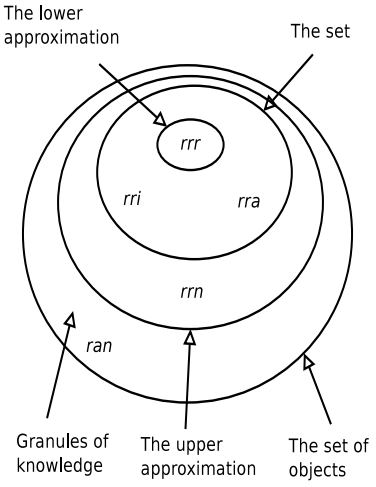


Fig. 2. Local rough set approximations including rrr

Response triples placed in two 'homogeneous' granules in lower approximation sets should be given serious consideration. Response triples placed in two 'inhomogeneous' granules smaller differences.

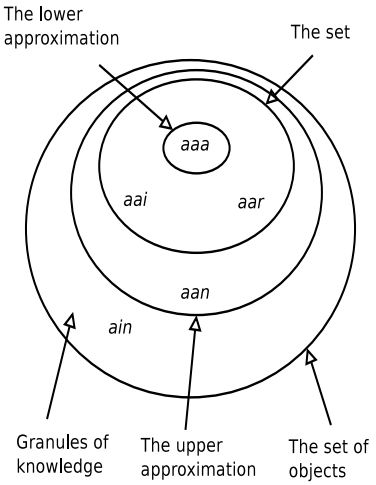


Fig. 3. Local rough set approximations including *aaa*

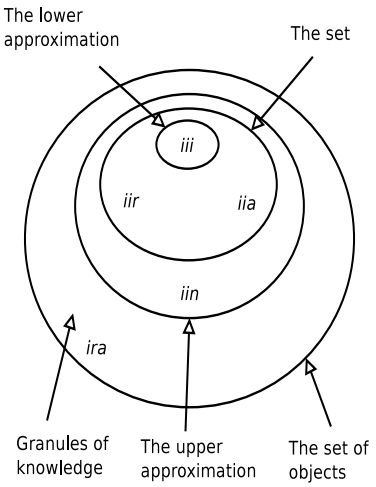


Fig. 4. Local rough set approximations including *iii*

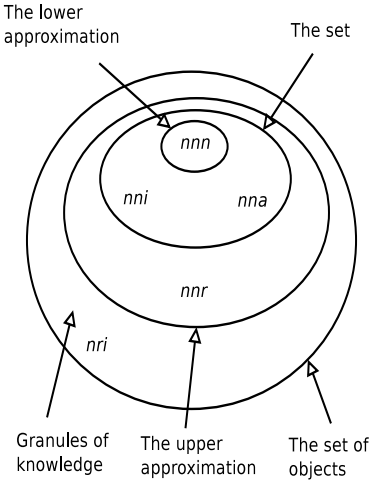


Fig. 5. Local rough set approximations including missing responses

3.2 Triplets Related to a Collaboration Option

Every collaboration option is assigned an ordered pair of response combination triplets where the first one is related to the importance criteria and the second one to the relevance criteria.

We assume projects are considered as being profitable if they are assigned an ordered pair of triplets where both triplets belong to the set {rrr, rra, rri}.

Two collaboration options assigned symmetrical pair of triplets like f. ex. {rrr, rra} and {rra, rrr} are considered equally profitable and are therefore placed in one node.

Collaboration options, criterion C1 and criterion C2, and previous collaboration outcomes are presented in Table 2.

Objects within equivalence classes of θ in Table 3 are indiscernible with respect to listed attributes.

The equivalence classes of θ with respect to criteria C1 and C2 are arranged in a knowledge lattice, Fig 6. This can be interpreted as follows: options A, C, D are considered to be equally profitable, option G comes afterwords while option I is note recommended.

The approximation qualities and the significance of a single criterion along with combination of criteria for predicting of respective outcome are presented in Table 4.

The performed calculations indicate that either of the chosen criteria can be used for predicting collaboration outcomes, provided working with the set of triplets {rrr, rra, rri}.

Table 2. Evaluation criteria

	C1	C2	Outcome
A	rrr	rrr	successful
B	rra	rra	not successful
C	rra	rrr	successful
D	rrr	rra	successful
E	rra	rri	not successful
F	rri	rra	successful
G	rrr	rri	successful
H	rri	rrr	successful
I	rri	rri	not successful

Table 3. Equivalence classes

Q	θ
C1	{A, B, C, D, F, G}, { E, H, I }
C2	{A, B, C, D, E, H}, { F, G, I }
C1 and C2	{A, C, D }, {B}, {E}, {F, H}, {G}, {I}
Outcome	{A, C, D, E, G, H}, {B, E, I}

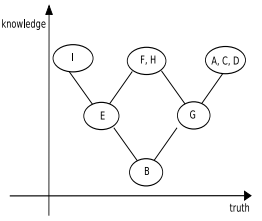


Fig. 6. A lattice for equivalence classes of θ with respect to criteria C1 and C2

Table 4. Significance of criteria C1 and C2

Criteria	γ	α	Elusidation
{C1}	0.66	0.05	Not casual
{C2}	0.66	0.05	Not casual
{C1, C2}	0.56	0.03	Casual

4 Conclusion

Both many valued logic and rough sets theory can be applied for evaluating collaboration criteria. In our study six valued logic show clear indications with respect to which collaboration options should be further considered in more details. On the other hand prediction methods originating from rough set theory can be used for ordinal ranking of both evaluation criteria and collaboration options.

References

1. Belnap, N.J.: How a computer should think. In: Contemporary Aspects of Philosophy. Proceedings of the Oxford International Symposia, Oxford, GB, pp. 30–56 (1975)
2. Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge University Press, Cambridge (2005)
3. Duntsch, I., Gediga, G.: Rough set data analysis: A road to non-invasive knowledge discovery. Methods Publishers (2000) ISBN: 190328001X
4. Fitting, M.: Kleene's Logic, Generalized. Journal of Logic and Computation 1(6), 797–810 (1991)
5. Font, J.M., Moussavi, M.: Note on a six valued extension of three valued logics. Journal of Applied Non-Classical Logics 3, 173–187 (1993)
6. Garcia, O.N., Moussavi, M.: A Six-Valued Logic for Representing Incomplete Knowledge. In: Proceedings of the 20th International Symposium on Multiple-Valued Logic (ISMVL), pp. 110–114. IEEE Computer Society Press, Charlotte (1990)
7. García-Duque, J., López-Nores, M., Pazos-Arias, J., Fernández-Vilas, A., Díaz-Redondo, R., Gil-Solla, A., Blanco-Fernández, Y., Ramos-Cabrer, M.: A Six-valued Logic to Reason about Uncertainty and Inconsistency in Requirements Specifications. Journal of Logic and Computation 16(2), 227–255 (2006)
8. Kaluzhny, Y., Muravitsky, A.Y.: A knowledge representation based on the Belnap's four valued logic. Journal of Applied Non-Classical Logics 3, 189–203 (1993)
9. Kleene, S.: Introduction to Metamathematics. D. Van Nostrand Co., Inc., New York (1952)
10. Lukasiewicz, J.: On Three-Valued Logic. Ruch Filozoficzny 5 (1920), English translation in Borkowski, L. (ed.): Lukasiewicz, J.: 1970, Selected Works. North Holland, Amsterdam (1920)
11. Moussavi, M., Garcia, O.N.: A Six-Valued Logic and Its Application to Artificial Intelligence. In: Proceedings of the Fifth Southeastern Logic Symposium (1989)
12. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)

13. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
14. Bierman, H., Smidt, S.: The Capital Budgeting Decision. Routledge, New York (2007)
15. Klein, G.A.: Recognition-Primed Decision Making. In: Klein, G.A. (ed.) Sources of Power: How People Make Decisions, pp. 15–30. MIT Press, Cambridge (1998)
16. Layard, R., Glaister, S.: Cost-Benefit Analysis, 2nd edn. Cambridge University Press (1994)
17. Nas, T.F.: Cost-benefit Analysis, Theory and Application. Sage Publications (1996)
18. Sim, K.M.: Bilattices and Reasoning in Artificial Intelligence: Concepts and Foundations. Artificial Intelligence Review 15(3), 219–240 (2001)
19. Zirger, B.J., Maidique, M.: A model of new product development: an empirical test. Management Science 36, 867–883 (1990)
20. Wagner, G.: Vivid Logic. LNCS (LNAI), vol. 764. Springer, Heidelberg (1994)

Interface Design for Decision Systems

Ching-Shen Dong and Ananth Srinivasan

Department of Information Systems and Operations Management, University of Auckland, Auckland 1142, New Zealand

Abstract. A key aspect of well-designed decision systems is the recognition of important decision functions implemented as explicit facets of the system. The literature is replete with discussion about what these facets ought to be. The essential ones involve the decision model, agents to handle task distribution and completion, data, solvers that execute model instances, visualization, and scenario development to explore problem instances. The interaction among these facets in conjunction with the user is what makes for effective problem solving. The additional dimension that must be recognized in design is the fact that users typically possess a wide range of abilities but are all vested with decision making authority. This makes it necessary for the interface design to provide the functionality described above while allowing for effective interaction by a range of users. We use a classification scheme for users by labeling them as system builders, professionals, and naïve users. We develop a framework that juxtaposes system facets with user abilities to derive some interface design principles. We present the results of our work with examples in the supply chain domain.

Keywords: decision systems, decision support systems, decision makers, user interface.

1 Introduction

Systems that support problem solving in unstructured decision environments have been studied by numerous scholars. Among the earliest references to this class of systems, referred to as Decision Support Systems (DSS), mention the importance of supporting users with a diverse skill set interacting with sophisticated problem solving assistance provided by a system [16], [5]. Many of the fundamental issues raised by them still remain as important considerations for the design of such systems today. The development of technology has provided a continuing impetus for the improvement in design methods for the implementation of such systems. Following the classic treatise on the subject by [2], improvements in design have focused on the effective integration of datasets, models and algorithms that operate on these datasets, and interfaces to allow users to interact with such sophisticated systems. Two fundamentally different approaches to developing such applications have been general purpose versus application specific development strategies. In this paper, we argue for the consideration of approaching the design of decision systems by acknowledging

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 172–181, 2012.

that users come with a variety of skill sets and yet have a sufficiently high degree of decision making authority and therefore need to be supported appropriately. We approach the design problem by focusing on effective interface design to provide such support. By developing a prototype in the supply chain domain [9], we show how proper design of the interface coupled with solid first principles can be effective in implementing decision systems in modern organizations. The paper is organized as follows. In Section 2 we provide a review of underlying theories of decision making that are important considerations for decision system design. Section 3 discusses the motivation for accommodating a diverse set of users. Section 4 describes the implementation of a prototype with a few examples of interaction scenarios. We conclude the paper with some implications for future work in this area.

2 Decision Making Processes

Decision making is the structuring and executing of decision making processes. Structuring and executing the process may be interwoven [14]. An organisational decision making process can be either rational or anarchical or in-between these two extremes in the continuum. Researchers have proposed several decision making models and six representative models, and implications for their support, are reviewed in this section.

In the bounded rationality model, the decision making process is a sequence of finite steps of intelligence, design and choice [15]. This model implies that the DSS user interface should help decision makers to specific programming routines and/or DSS components such as data, models and solvers (i.e. resources) to help decision makers process information efficiently.

In the garbage can model [4], problems are semi-structured and/or unstructured. The decision making processes are anarchical, with problems, issues, solutions and participants of decision making mixed up in the organisation. This model implies that the DSS user interface should help decision makers to utilise the richer concepts of its decision environment such as resources, processes (or lifecycle), locations, time and the participants involved.

The bounded rationality model is unrealistically ideal and the garbage can model is anarchical. The iterative sequence model [10] is something in-between these two extremes. The focus of this model is on the iterative decision making process between design and choice activities. This model implies that the DSS user interface should help decision makers to perform loop learning or what-if analysis

In this model, problem and solution spaces are gradually reduced during the decision making process [8]. Continuous user inputs are critical in this decision making process model. Decision making does not occur at distinct times and locations but is made gradually at different times and locations. This model implies that the DSS user interface should help decision makers to efficiently access data, model and solver resource, and to refine these resources for their decision making without time (i.e. time) and location restrictions (locations).

Insights from decision makers play an important role in the decision making process [8]. The DSS provides an inspirational environment for decision makers. This model implies that the DSS user interface should help decision makers to dramatically narrow the problem space, such as providing a process for extracting high density information from large volumes of data; and/or narrow the solution space, perhaps by providing a process to modify models at run time or adding new solvers from external systems and providing full model lifecycle support.

The interwoven model suggests that decisions interact within organizations. In other words [8], decision making processes are linked and affected by other decision making processes across time, locations and people. The decision process can not be isolated and it will affect or be affected by other decision processes for the same or different issues.

From reviewing decision making process models, the support required by decision making processes is summarised in Table 1. From Table 1, we can conclude that DSS user interface should be designed to help decision makers to address the issues of Resource, Lifecycle, Location and Time. The skill levels of decision makers affect the decision making process. We explore the DSS users in next section.

Table 1. Decision making models and problem dimensions

Decision making process	Issues should be addressed	Issues categories
Bounded Rationality Model	Specific programming routines or DSS components	Resource and lifecycle
Iterative Sequence Model	Loop learning or what-if	Resource and lifecycle
Garbage Can Model	Resources, processes, locations, time and the participants	Resource, lifecycle, location and time
Convergence Model	Time	Time
Insightful Model	Extensibility	Time, lifecycle, resource and location
Interwoven model	Time, integration, resource, multiple participants and locations	Time, lifecycle, resource and location

3 Decision Support Systems and Skill Levels of Decision Makers

[16]'s DSS component framework has been widely cited for the design and development of DSSs. In the framework, he divides a DSS into five components: Database, i.e. data used in DSSs; Model base, i.e. models used to represent problems to be solved in DSSs; Database Management Software or Database Management System (DBMS); Model Base Management Software or Model Base Management System (MBMS); and Dialogue Generation and Management Software (DGMS) or user interface.

[16] builds a framework in which three levels of Decision Support Systems are defined. These three levels of technologies are DSS tools, Specific DSSs (SDSSs) and

DSS Generators (DSSGs) and are used in the development and operation of a DSS by its users such as manager, intermediary, DSS Builder, technical supporter, and toolsmith.

Most decision support systems are designed for specific users, varying from end-users to model builders or system builders. Their skill levels also vary, from naïve, advanced beginner, competent and proficient to expert, and their support needs are different. Support for decision makers/stakeholders is very restricted and that for collaboration between stakeholders is lacking. In the Dreyfus decision style model, the skill levels of decision makers can be categorized into novice, advanced beginner, competent, proficient, expert and master [12]. Decision makers at different skill levels use different approaches. Novices tend to decompose problems into small parts and they use analytical approaches to make decisions. Masters treat problems in a holistic fashion and they use intuitive judgment. A novice decision maker may become a master decision maker over time. This model implies that requirements for decision makers vary with different skill levels. DSSs need to cater for the needs of different decision makers.

To simplify the categories of DSS users at different skill levels, we can classify them into Naïve User, Professional and System Builder. A knowledge worker can be a naïve user, professional or system builder. A business worker can be classified at different skill levels at different stages of her/his business career. The DSS user interface needs to be designed to fit different user needs. It does not mean that one category of users is more important than the others. We discuss these three levels of Decision Support Systems in the following section.

4 DSS Implementation to Support Key DSS Stakeholders – System Builder, Professional and Naïve User

4.1 DSS Component-DSSG-SDSS-Scenario Lifecycle

The motivation for our design lies in charting out some lifecycle ideas involved in DSS design and use. Using the concepts of the single and double loop learning of [1]'s organizational decision making we describe the DSS Component-DSSG-SDSS-Scenario lifecycle (Fig. 1). DSS components are assembled to build a DSSG which is used to generate SDSSs for solving specific problems. Decision makers take a snapshot of their decision making processes in the SDSS to save as a scenario – a specific problem instance - which can be used in the future as a DSS for the same or similar decision problems. If the results of scenarios can no longer produce expected solutions, based on the feedback of end users, new scenarios can be produced by changing some of the actions of the SDSSs. This is referred to as "single loop learning". If the SDSS can not meet the decision maker needs, using feedback from end user developers, DSS builders can generate new SDSSs with new settings of the DSS components; for example, modifying the DSSG by adding different DSS components. This is referred to as "double loop learning". Due to changes in business needs and user requirements, current decision making paradigms (and related DSS components) may no longer be able to support decision making. The feedback from DSS builders or end user developers are used by toolsmiths to modify existing DSS components or design new DSS components to support the new paradigm; we term

this as "triple loop learning". The DSS Component-DSSG-SDSS-Scenario lifecycle involves different skill levels of users at each phase. Typically, toolsmiths have more system knowledge and end user developers possess more domain knowledge. We apply these principles to our implementation described in the next section.

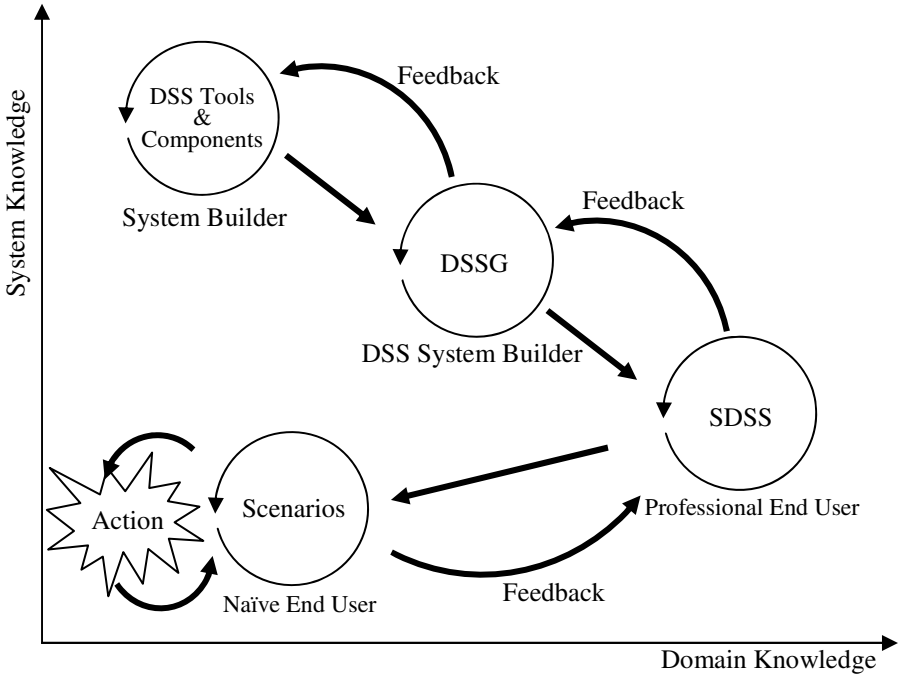


Fig. 1. The role of DSS users in the implementation

4.2 Implementation Domain

We implement an Agent-enabled Distributed Decision Support System Generator (ADDSSG) in the Collaborative Planning, Forecasting and Replenishment (CPFR) aspect in the Supply Chain domain. The implementation of the CPFR process consists of retailer-manufacturer (two-tier) deployment. The purpose is to highlight the potential of the agent-enabled distributed decision support system generator for solving SCM problems in the real world for different skill levels of users. The prototype uses the concepts of the independence of DSS components, the decision support system generator, specific decision support system and scenario, to address the collaboration problem of supply chain partners (retailer and manufacturer) to implement information exchange, sharing and discovering.

To address the resource, location, lifecycle and time issues and support different levels of DSS users, such as DSS builders, intermediary and DSS end-users, this prototype system provides configurations for system builders, professional and naïve users (Fig. 2). Assigning functions to different skill levels of users should be based on

requisite system knowledge and domain knowledge that are briefly discussed in section 4.1. For example, the system builder mode is set for DSS builders who have the system knowledge and domain knowledge and toolsmiths who have system knowledge; the professional mode is for knowledge workers who have both system knowledge and domain knowledge; and the naïve user mode is for DSS end-users who may only have limited domain knowledge. Based on the requirements of system and domain knowledge, the different functions allocated to various DSS users in our prototype are listed in Table 2.

Table 2. Functions assigned for different user levels in the prototype

Functions		System Builder	Professional	Naïve User
Model	Loading	v	v	v
	Parameter	v	v	v
	Operation	v	v	v
	Creation	v		
	Modification	v		
	Persistence	v	v	
	Pool	v	v	
Agent	Get Object	v	v	
	Send Agent & Object	v	v	
	Get Remote Agent & Object	v	v	
Data	View Local Data	v	v	v
	Loaded Data	v	v	v
	Pool	v	v	
Solver	Loading	v	v	
	Parameter	v	v	v
	Operation	v	v	v
	Pool	v	v	
Visual	Loading	v	v	
	Display	v	v	v
	Parameter	v	v	v
	Operation	v	v	v
	Pool	v	v	
Scenario	States	v	v	v
	Processes	v	v	
	Objects in Mapping	v	v	

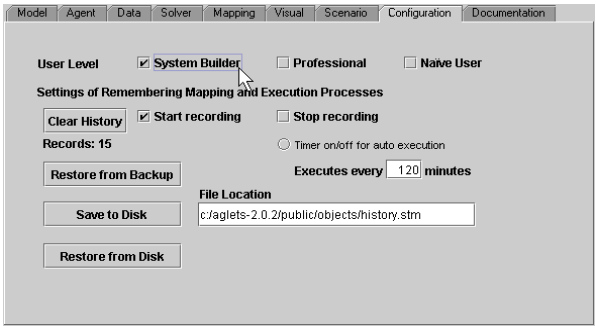


Fig. 2. Configuration of user types

4.3 System Builder

A system builder uses software agents (services), data, models, solvers and visualization components to build SDSS in the DSSG. The system builder is authorized to create and modify models that are in solving specific problems for decision makers. The decision support systems built by the system builder is called SDSS. In some cases, DSSGs can also be used directly by decision makers. The system gives the system builder the ability to address resource, location, lifecycle and time issues with all the basic functions through the upper tabs (Model, Agent, Data, Solver, Mapping, Visualization, Scenario, Configuration and Document) and advanced functions at bottom tabs. The screen shots in Fig. 3 & 4 show that the system builder loads the forecasting models from either a model library or gets the model instance from software mobile agents [6], [7]. Under the Model tab, the system builder performs more advance functions such as, loads models (Loading tab), adds more model parameters (Parameter tab), adds more new methods to the model (Operation tab), creates complete a new model at runtime (Creation tab), changes the model parameters and methods for the loaded model (Modification tab), saves the model in the system to storage (Persistence tab) for future use and/or adds the model to a DSS component pool shared by other decision makers (Pool tab).

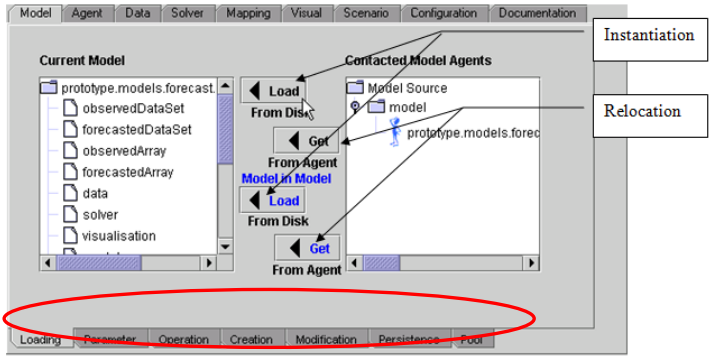


Fig. 3. The user interface used for loading a model by system builders

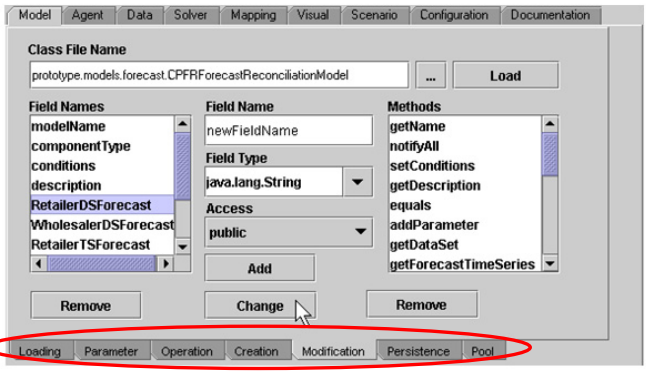


Fig. 4. The user interface used for modifying models by system builders

4.4 Professional User

A SDSS is built with specific model or models for decision makers. A professional user loads the data and applies solvers to solve problems in their business domain. The professional user can use what-if analysis to identify the most appropriate data and solvers to achieve solutions. Various combinations of loaded data and solvers with the models in a SDSS can be saved at different points in time as different scenarios for solving specific problems. The system gives the professional user the power to address resource, location, lifecycle and time issues with all the basic functions through the upper tabs (Model, Agent, Data, Solver, Mapping, Visualization, Scenario, Configuration and Document) and selected functions at bottom tabs (e.g. Loading, Parameter, Operation, Persistence and Pool for modeling) in Fig. 5. It should be noted that the bottom tabs for the Professional User are composed of a subset of the bottom tabs for the System Builder (see highlighted portions of Fig. 4, 5, and 6).

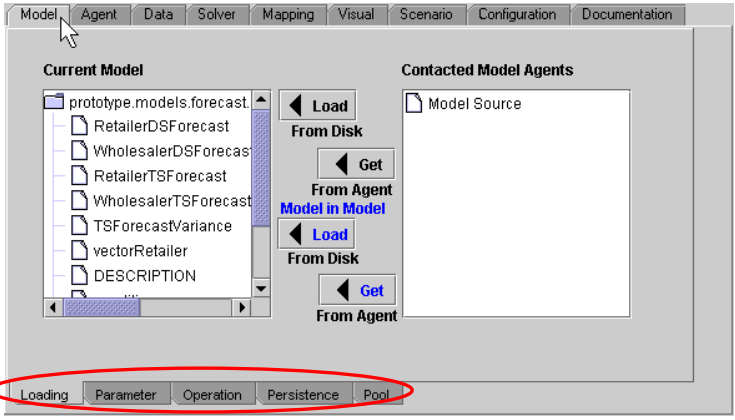


Fig. 5. Functions under the Model tab for professional users

4.5 Naïve User

A naïve user may not have enough working knowledge or decision skill to solve complicated problems. Scenarios can be used for problem analogs in a business domain. In organizations, a naïve user is typically given the scenarios for solving specific problems. The system gives the naïve user the power to address resource, location, lifecycle and time issues in a simple format. The naïve user is given all the basic functions through the upper tabs (Model, Agent, Data, Solver, Mapping, Visualization, Scenario, Configuration and Document) and only minimum advanced functions at bottom tabs (e.g. Loading, Parameter and Operation for modeling) in Fig. 6. Again note that the bottom tabs for the Naive User is composed of a subset of the bottom tabs for the Professional User.

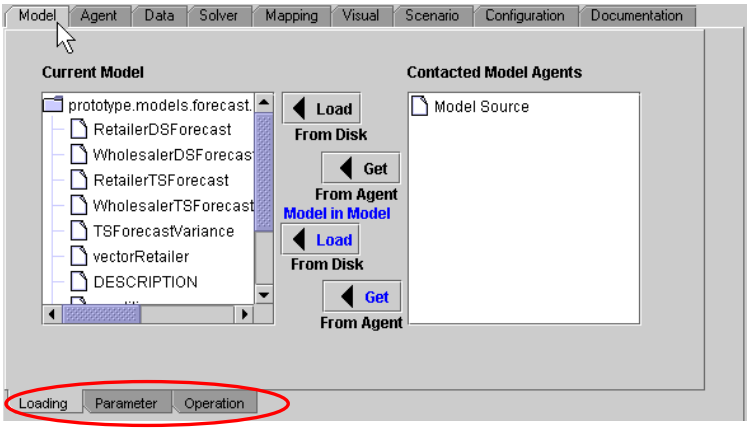


Fig. 6. Functions under the Model tab for naïve users

5 Conclusion

The proposed frameworks, architectures and implementations in this research are evaluated by peer review using the iterative approach of evaluation, theory building, systems development. Further evaluation cycles fine-tune our frameworks, architectures and prototype. This process of fine-tuning driven evaluation is followed by a review process carried out by professional experts to seek further improvement. We view this step as constituting an expertise-based validation of the research. The experts involved in this phase were drawn from the areas of decision support systems, system design, operations management and supply chain management. An evaluation framework was presented to the experts along with a prototype of the system and they were asked to rate the prototype according to the framework. This exercise yielded good results as viewed by the experts. While this is a preliminary attempt at evaluation, a more systematic evaluation exercise is warranted.

In this paper we make a case for the design of decision systems that support organizational decision making by focusing on interface aspects of the design to enable support of users with varying degrees of system and domain knowledge. Such a design approach allows for these systems to be more widely used and appreciated in organisations. It also overcomes the generally reported problem of application specific designs which get used by a small number of analysts thereby bringing the issue of cost effectiveness of such implementation into focus.

References

1. Argyris, C.: Single-loop and double-loop models in research on decision making. Administrative Science Quarterly 21, 363–375 (1976)
2. Ariav, G., Ginzberg, M.J.: DSS Design: A Systemic View of Decision Support. Communications of the ACM 28(10), 1045–1052 (1985)
3. Broadbent, M., Weill, P., St. Clair, D.: The implications of information technology infrastructure for business process redesign. MIS Quarterly 23(2), 159–182 (1999)
4. Cohen, M.D., March, J.G., Olsen, J.P.: A garbage can model of organizational choice. Administrative Science Quarterly 17(1), 1–25 (1972)
5. Gorry, G.A., Morton, M.S.C.: A framework for management information systems. Sloan Management Review 13(1), 55–70 (1971)
6. Kotz, D., Mattern, F.: Agent systems, mobile agents, and applications. Springer, Zurich (2000)
7. Lange, D., Oshima, M.: Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley Professional (1998)
8. Langley, A.H., Mintzberg, P.P., Posada, E., Saint-Macary, J.: Opening up decision making: The view from the black stool. Organization Science 6(3), 260–279 (1995)
9. Mabert, V.A., Venkataramanan, M.A.: Special Research Focus on Supply Chain Linkages: Challenges for Design and Management in the 21st Century. Decision Sciences 29(3), 537–552 (2007)
10. Mintzberg, H., Raisinghani, D., Theoret, D.: The Structure of “unstructured” decision processes. Administrative Science Quarterly 21(2), 246–275 (1976)
11. Power, D.J., Sharda, R.: Model-Driven Decision Support Systems: Concepts and Research Directions. Decision Support Systems 43, 1044–1061 (2007)
12. Sage, A.P.: Decision Support Systems Engineering. Wiley-Interscience, New York (1991)
13. Singh, M.P., Huhns, M.N.: Service-oriented computing semantics: Semantics, processes, agents. John Wiley, New York (2005)
14. Silver, M.S.: Systems that support decision makers. John Wiley, New York (1991)
15. Simon, H.A.: The new science of management decision. Harper & Row, New York (1960)
16. Sprague, R.H.: A Framework for the development of DSS. MIS Quarterly 4(4), 1–26 (1980)

Opponent Modeling in Texas Hold'em Poker

Grzegorz Fedczyszyn, Leszek Koszalka, and Iwona Pozniak-Koszalka

Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
g.fedczyszyn@gmail.com,leszek.koszalka@pwr.wroc.pl,
iwona.pozniak-koszalka@pwr.wroc.pl

Abstract. In this paper a new algorithm for prediction opponent move in Texas Hold'em Poker game is presented. The algorithm is based on artificial intelligence approach – it uses several neural networks, each trained on a specific dataset. The results given by algorithm may be applied to improve players' game. Moreover, the algorithm may be used as a part of more complex algorithm created for supporting decision making in Texas Hold'em Poker.

Keywords: Poker game, algorithm, artificial intelligence, neural network.

1 Introduction

Texas Hold'em Poker is currently the world's most played card game. Hundreds of thousands of people play this game every day and can play in online Poker rooms as well as in real life. One of the main reasons for Poker's recent success is its fundamental dynamics. The 'hidden' elements of the game mean players must observe their opponent's characteristics to be able to make a good move, i.e., to choose a good decision.

In order to play Poker well, a Poker player needs to constantly think about next move to be made by his opponents. For example, when the Poker player hand is very likely to win in certain spot, he should figure out how to win the most money from his opponents. Good player also has to recognize spots when the opponents may fold and try to somehow estimate how often will this happen in certain spot and when bluffing may be a profitable option. To do this, the player must take several factors into account, including such as: what kind of opponents is he facing, what kind of board is on the table, what was his and the opponents' previous action, does he have position over the opponent, and many more.

In literature, it can be found some ideas of opponent modeling in Texas Hold'em Poker. Van der Klein [1] described algorithm based on decision tree but Mccurley [2] proposed using artificial intelligence agent approach.

In this paper, we propose a machine learning algorithm that predicts opponent's action when certain information about the opponent and the current state of the game is given. The algorithm is based on several neural networks trained in predicting Poker player moves.

The rest of the paper is organized as follows. In Section 2, Texas Hold'em Poker rules have been explained. In Section 3, two algorithms for prediction Poker moves have been described, including the proposed one. Section 4 focuses on the designed

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 182–191, 2012.

and implemented experimentation system. Section 5 is devoted investigations – it contains brief analysis of the results of experiments and their discussion. Conclusion and perspectives appear in Section 6.

2 Texas Hold'em Poker

2.1 Game Description

Texas Hold'em Poker is played with a standard deck of 52 cards. Typically the number of player in cash games varies from 2 to 9. Each player is dealt 2 card faces down. These cards are called hole cards or pocket cards. There are used to compose the five cards hand – each of this hands belongs to one of categories listed in Table 1. This category determines the strength of player's hand.

Table 1. Poker hands ranks

Hand	Example	Description
Royal flush	T♠ J♠ Q♠ K♠ A♠	A straight flush T to A
Straight flush	2♠ 3♠ 4♠ 5♠ 6♠	A straight of a single suit
Four of a kind	A♠ A♣ A♡ A♢ 5♠	Four cards of the same rank
Full house	A♠ A♣ A♡ K♠ K♣	Three of a kind + one pair
Flush	A♠ 4♠ 7♠ 9♠ K♠	Five cards of the same suit
Straight	2♠ 3♣ 4♣ 5♡ 6♢	Five cards of sequential rank
Three of a kind	A♠ A♣ A♡ K♠ Q♣	Three cards of the same rank
Two pairs	A♠ A♣ K♠ K♣ 3♡	Two pairs
One pair	A♠ A♣ K♠ Q♣ 3♡	Two cards of the same rank
High card	A♠ 3♣ 7♣ 9♡ Q♢	None of the above

Game starts when two players next to the dealer pays small blind and big blind. Dealer is one of the players and that designation moves clockwise around the table after each hand is finished. Small blind and big blind are small amounts of money that has to be paid by two players to the left of the dealer before they can see their hole cards. Those forced bets are costs of the game – their forces all players to play more hands - without the blinds players could just sit at the table and wait for the best starting hand and fold any other hand.

When the blinds are on the table players may look at their hole cards and the pref-lop betting round begins. Each betting round is also called street or barrel. In pre-flop betting round players takes action clockwise starting from a player left to big blind player. There are five possible player actions. When player is not facing a bet he can bet or check and when he is facing a bet he can raise, call or fold. Of course player can also fold when he is not facing a bet but this move does not make any logical sense and it will not be counted as a possibility for player in this work. Each move represents one of the following actions:

- **Fold:** the player does not put in any more money, discards his cards and surrenders his chance at winning the pot. If he was one of two players left in the game, the remaining player wins the pot and does not have to show his pocket cards.
- **Call:** the player matches the current maximum amount wagered and stays in the game and if he is last to act in a betting round game continuous with next betting round or showdown.
- **Raise:** the player matches the current maximum amount wagered and additionally puts in extra money that the other players now have to call in order to stay in the game.
- **Bet:** this is similar to a raise. When a player was not facing a bet by an opponent (and the amount to call consequently was 0) and puts in the first money, he bets.
- **Check:** this is similar to a call, except the amount to call must be 0: if no one has bet (or the player has already put in the current highest bet, which could be the case if he paid the big blind) and the player to act does not want to put in any money, he can check.

Betting round is over when either all players but one folded or all players that did not fold matched the current highest bet by calling it or going all-in (putting all their money into the pot). After preflop is over three cards are being dealt face up on the table. These three cards are called flop cards and the next betting round called flop occurs. First player left to the button is first to act. Actions that can be performed by players are the same as in preflop round. After this betting round is over one more card is being dealt face up and it is added to community cards. This card is called turn card and next betting round is begun. After it is over and there are still two or more players in a game that did not fold fourth and last card is dealt on the table. This card is called river - it starts the last betting round. When the river betting round has completed and two or more players are still in the game, the showdown follows.

On showdown each remaining players reveal their pocket cards and the player with a strongest five cards poker hand wins the pot, if two or more players have the same hands they split the pot. Poker hand must be composed from five cards. Player can use both of his hole cards, one of them or non – in this case his hand is composed only from community cards.

2.2 States of Poker Game

As described above Poker game consists of 4 betting rounds (also named streets or barrels): preflop, flop, turn and river. In each of those rounds player can be facing a bet and may have an option to raise, call or fold or not facing a bet and be able to check, bet or fold. Of course, folding while not facing a bet is possible but does not make any logical sense, so this move will not be considered as it almost never appears in a real Poker game.

That kind of representation gives up 8 states of a Poker game:

(i) preflop facing a bet,
(ii) preflop not facing a bet,
(iii) flop facing a bet,
(iv) flop not facing a bet,

(v) turn facing a bet,
(vi) turn not facing a bet,
(vii) river facing a bet,
(viii) river not facing a bet.

3 Algorithms

3.1 Opponent Modeling Using Decision Trees

The algorithm, presented in [1], applied a machine learning to predict opponent moves. Authors decided to use decision trees as they give results very fast when trained and have many advantages needed in this particular case. In the implementation the input data set is composed by over 40 different features taken into account for different states and aspects of game. Authors of this algorithm have also proposed a method called group specific opponent modeling. In this method the different trees are created not only for different states of the game but also for different kind of opponents whose moves they are trying to predict. In the algorithm they created 9 different opponent models using K-model clustering. Finally, the number of 72 different decision trees - each for different opponent model and different state of the game – are considered. More detailed description of the algorithm can be found in [1].

3.2 Opponent Modeling Using Neural Networks

In this sub-section the newly created algorithm for prediction opponent moves is presented. The algorithm is a modification of method presented in 3.1. The most important difference is that instead of decision trees - neural networks has been used.

This modification may significantly improve quality of classification of moves as the neural networks are known to perform well in a noisy domain [4].

Inputs and Outputs. For each state of Poker game, one multi-layer neural network is created. Each network differs in number of inputs as on each street more information can be used to predict opponent's move. For pre-flop it is 20 inputs, on a flop it is 31, on a turn 37 and 43 on a river. Chosen inputs are describing state of the game using information about the current board, opponents that the player is facing, his position on a table, stack sized etc. Chosen inputs are representing aspects of a game that the good poker players are usually taking into account when trying to predict opponent move. Each network has 2 or 3 outputs: 2 for not facing a bet (check or bet) and 3 for facing a bet (fold, call, and raise).

K-model Clustering. In Poker game, the player uses different strategies when playing. For one player some factors may be more important than for other. Some players tend to call more when others wait for a very strong hand and fold all other hands. Good poker players usually label each of their opponents to remember how to play against them. Because of such clusterization of players, one should improve quality of opponent modeling. The idea of K-model clustering, which is used in this work, does not require any expert knowledge or distance measure like in k- mean clustering.

In K-model approach, clustering one set of neural networks is created for each cluster of players. At first, each player is randomly assigned to one cluster. Then, a set

of neural networks corresponding to each cluster is being trained on hands of players in cluster (Fig. 1). Next step is to forget of the current cluster assignment for players and to find a new one. Repeating, we have models trained on random players (Fig. 2).

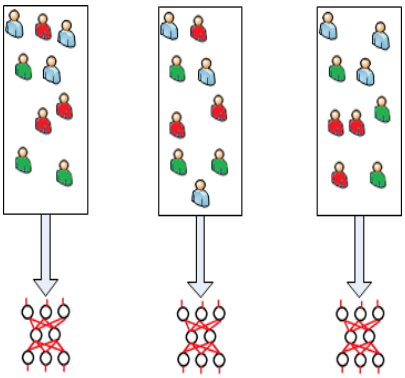


Fig. 1. Training models

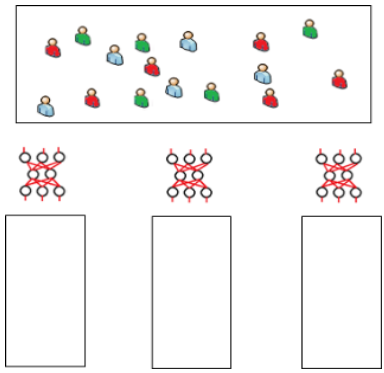


Fig. 2. After training

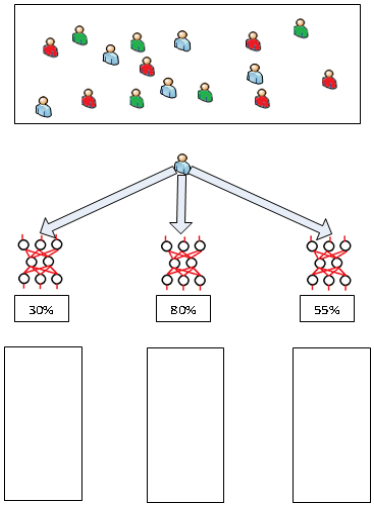


Fig. 3. Choosing a new cluster

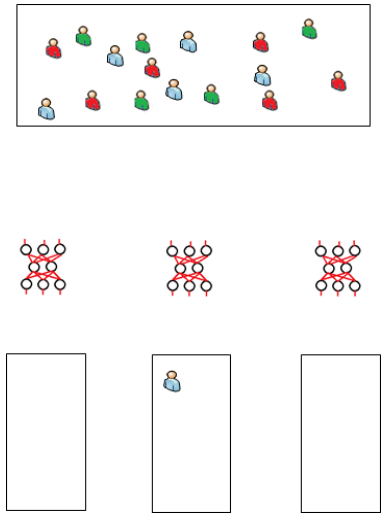


Fig. 4. Moving player to a new cluster

Then, we are trying to find a new cluster for each of the players by calculating the accuracy of the reached model for each of the players hands (Fig. 3). We assign each player to model that gives the highest correct classification rate for this player's hands (Fig. 4).

After we do it for all of the players we should get results like in Fig. 5 where each cluster contains mostly players of the same type.

Next, we repeat this procedure and we stop it after none of the players changes his cluster assignment (Fig. 6).

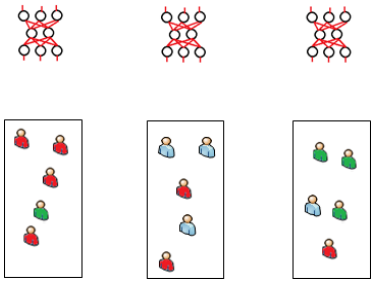


Fig. 5. Example results after first iteration

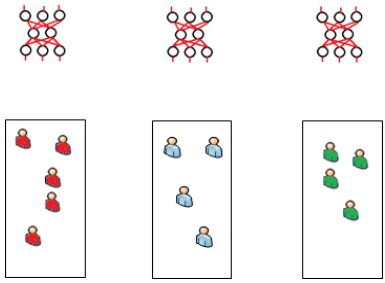


Fig. 6. Example results after convergence

4 Experimentation System

To perform tests and check how the proposed algorithm works an experimentation system was created. The idea of the system is presented in Fig. 7. There are distinct two inputs: (i) problem parameters - data set, and (ii) algorithm parameters - neural networks parameters. As outputs we consider trained neural networks, and the obtained results of classification process.

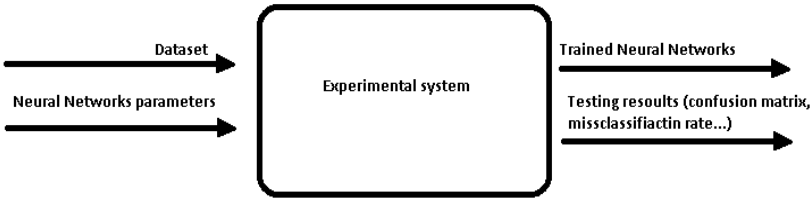


Fig. 7. Experimentation system

Input Dataset. Used dataset was created by observing an online Poker games and saving hand histories in a text files. For experiments, 100.000 hands played on No Limit $2, No Limit $5 and No Limit $10 with 5 and 6 players on a table were used. That dataset gave 753.595 moves (network input objects) performed by players. Such a big number of input objects were used due to noise and variance of the domain which in this case is very high. Dataset statistics can be found in Table 2. As shown in this table, the most common move on flop is fold (70% of moves), on flop 39% of moves are checked, on turn 43% of moves are checked, and on river it is 44%. That means that dummy classificatory that would always choose most common move on each street as an answer would get 61% of overall correct classification accuracy.

Neural Network Parameters. All networks were trained using back propagation learning algorithm with learning rate set to 0.3. Each network was trained over the same dataset 500 times as this number of iteration was found to be sufficient to minimize the error as much as it was possible. K-model clustering was set to divide players into 8 clusters (K=8). Each network is a 2 layer network where number of neurons in a middle layer is a half of a number of inputs. Activation function in all

neurons was bipolar sigmoid function with alpha parameter set to 1. Momentum of each network was set to 0.1.

Table 2. Dataset statistics

	Preflop	Flop	Turn	River	Total
Fold	3383038 (71.71%)	149375 (17.62%)	54951 (13.32%)	31097 (13.68%)	3618461 (58.31%)
Check	80197 (1.70%)	338292 (39.91%)	183084 (44.39%)	107659 (47.36%)	709232 (11.43%)
Call	556992 (11.81%)	100507 (11.86%)	49470 (11.99%)	19437 (8.55%)	726406 (11.71%)
Bet	18833 (0.40%)	226735 (26.75%)	111137 (26.94%)	62217 (27.37%)	418922 (6.75%)
Raise	678843 (14.39%)	32747 (3.86%)	13830 (3.35%)	6920 (3.04%)	732340 (11.80%)
Total	4717903 (76.03%)	847656 (13.66%)	412472 (6.65%)	227330 (3.66%)	6205361 (100.00%)

System Outputs. As the training process takes a lot of time, after training, each network is saved as an XML file and can be used later for performing more tests. Moreover, the results of experiments are automatically gathered and saved – the confusion matrices and the correct classification rates for each network are created.

5 Investigation

All eight networks where trained on given dataset 500 times using 90% of input dataset. Remaining 10% was used for testing. Results of experiments are shown in the confusion matrices (Table 3 and Table 4), where AC means the assigned class and TC means the true class. The confusion matrices are corresponded to the states of a game.

Table 3. Facing a bet results

	TC - Fold	TC - Call	TC - Raise	Total
AC - Fold	75.52 %	12.42 %	11.94 %	85.47 %
AC - Call	29.16 %	59.11 %	11.47 %	11.88 %
AC - Raise	31.00 %	9.86 %	54.98 %	2.65 %
Total	69.11 %	17.87 %	13.02 %	73.16 %

Each row in tables contains percentage value of correct or incorrect (confused) classification for each class. The last column denoted as ‘Total’ contains percentage values of how many objects there were in a testing set while ‘Total’ row informs of how many objects were classified as objects of a given class. The cell, where ‘Total’ row and ‘Total’ column are crossed, contains percentage of correct classifications.

Table 4. Not facing a bet results

	TC - Check	TC - Bet	Total
AC - Check	75.27 %	24.73 %	71.12 %
AC - Bet	33.41 %	66.59 %	28.88 %
Total	63.18 %	36.82 %	72.77 %

In order to measure the performance of clustering we used three quality measures:

VPIP - Voluntary Put money Into Pot which tells us how often player plays a game preflop (does not fold preflop).
PFR - PreFlop Raise which informs how often player raises pre-flop.
AF - Aggression Factor which informs how aggressive player is.

The obtained results are shown in Fig. 8 and in Fig. 9.

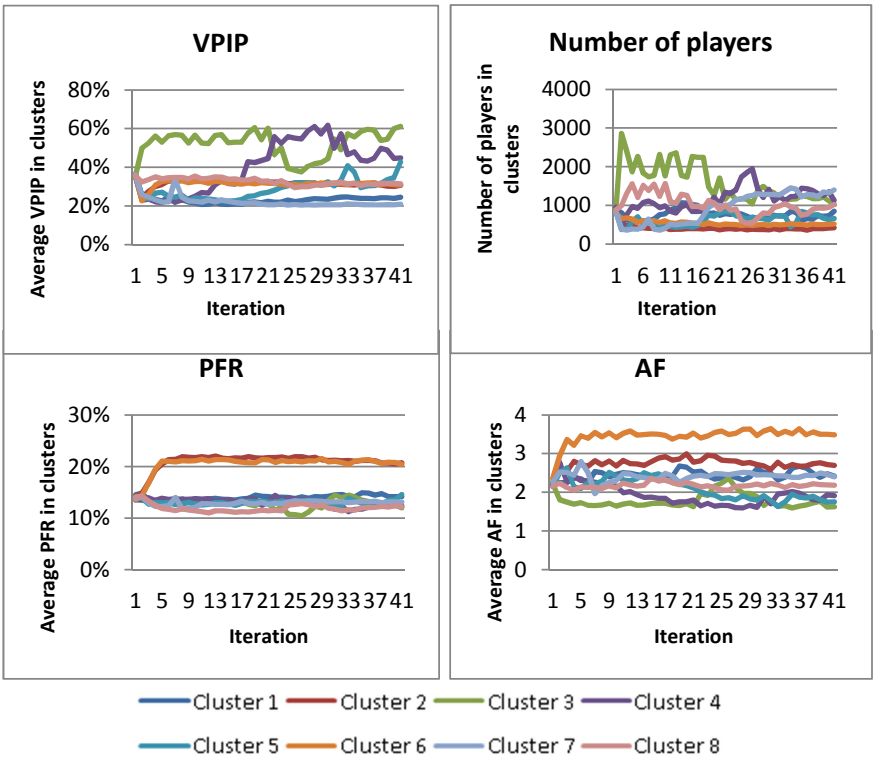


Fig. 8. Performance of VPIP, PFR, and AF indicators

The confusion matrices are corresponded to the two states of a game. Each row in tables contains percentage value of correct or incorrect (confused) classification for each class. The last column denoted as 'Total' contains percentage values of how many objects there were in a testing set while 'Total' row contains percentage values

of how many objects were classified as objects of a given class. The cell, where 'Total' row and 'Total' column are crossed, contains percentage of correct classifications.

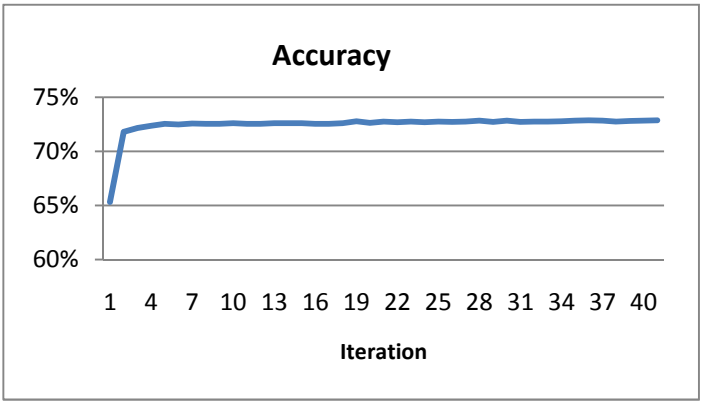


Fig. 9. Performance – overall accuracy.

K-model clustering algorithm divided the player pool into 8 clusters. It may be observed on the presented graphs (Fig. 8) that after 41 iterations of the algorithm each cluster represented different kind of player. Average VPIP in each cluster varies from 20% to about 60%, average PFR from 12% to 22%, and AF from 1.8 to 3.5. Also on "Number of players" graph we can see that the number of players in each cluster changed during the clustering process. Unfortunately, the proposed algorithm was stopped before the stop condition was met – none of the players changes cluster – because it was working for a long time and the improvement in accuracy on any iteration was very small.

Overall correct classification accuracy achieved on whole testing set was 72.8%. (Fig. 9). Surprisingly, algorithm did better for facing a bet state than for not facing. Accuracy also increased during iterations and a gain was biggest for several first iterations but on later phase it was also continuously improving.

Comparing those results to the results from [1] we observed that there are about 4% better for 6 players' game. Unfortunately the authors of [1] tested their algorithm over different datasets composed of hands that did not include real money but only "play" money. They obtained 71% accuracy of predicting opponent moves for 6 players' dataset. We were not able to access the dataset considered in [1], so the comparison may not be accurate.

6 Conclusion

In this paper, the method for predicting opponent's moves has been presented. This method uses 64 neural networks that are trained to predict different opponent's actions in various stages of a game. As a result, we obtained a universal tool for predicting opponent's moves that do not depend on opponent's playing style and strategy.

The overall correct classification percentage obtained when using the created algorithm was about 18%. It is better than a simplest classifier that would choose most common move on each betting round. It is also about 4% better than the result presented in [1] while method based on decision trees was used. Neural networks seem to operate better in very noisy environment which game of Poker surely is.

By looking at the obtained confusion matrices, it is easy to notice that a lot of moves were confused with folds and that the fold was the most correctly recognized move. That stems from the fact that the players usually fold to a bet. We can observe this phenomenon in Table 2, where only 3% of all moves occurred at river – the last betting round. Having that in mind, we might come up with an improvement to compose the dataset mostly of hands that gave more action and hands in which the pot was bigger than usual. That would also require us to somehow predict where the hand is going before actual prediction of a move.

K-model clustering algorithm seems to be an effective way to divide poker players into categories basing only on hand that they played in the past. It divided the player pool into clusters that differ in average statistics, thus, we may suppose that these players also use different poker strategy or game style. Taking it in to consideration should improve the quality of overall classification.

References

1. Van der Kleij, A.A.J.: Monte Carlo Tree Search and Opponent Modeling through Player Clustering in no-limit Texas Hold'em Poker. University of Groningen, The Netherlands (2010)
2. Mccurley, P.: An Artificial Intelligence Agent for Texas Hold'em Poker. Dissertation, University of Newcastle upon Tyne, The U.K. (2009)
3. Forge, A.: NET documentation available at `http://code.google.com/p/aforge/`
4. Xhemali, D., Hinde, C.J., Stone, R.G.: Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. University Loughborough, Leicestershire (2009)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2009)
6. Davidson, A.: Opponent Modeling in Poker: Learning and Acting in a Hostile and Uncertain Environment. Master's thesis, University of Alberta, Edmonton, Canada (2002)
7. Davidson, A., Billings, D., Schaeffer, J., Szafron, D.: Improved Opponent Modeling in Poker. In: Proceedings of International Conference on Artificial Intelligence, ICAI 2000, Las Vegas, Nevada, The U.S., pp. 1467–1473 (2000)
8. `http://www.pokerstrategy.com/glossary/` - Poker glossary

On Axiomatization of Power Index of Veto

Jacek Mercik

Wroclaw University of Technology, Wroclaw, Poland
jacek.mercik@pwr.wroc.pl

Abstract. Relations between all constitutional and government organs must be moderated and evaluated depending on their way of decision making. Among their attributes one may find the right to veto. It is known already that a priori veto is rather strengthening the position of beholder. The evaluation of a power to make a decision is directly connected with a way of power measuring, i.e. with power index choice. In the paper we consider axiomatic base for such choice of an index of power evaluation.

Keywords: veto, power index, axioms.

1 Introduction

Relations between all constitutional and government organs must be moderated and evaluated depending on their way of decision making. Among different attributes of collective decision making one may find the right to veto. We think that veto a priori is rather strengthening the position of beholder. So, any considerations about consensus process must include evaluation of veto attribute as well, for to preserve the balance between sides.

The main goal of the paper is the analysis of axioms of power measure connected with veto attribute of a decision maker. In certain cases, it is possible to calculate a value of power of veto attributed to the decision maker and to give the exact value of the power index as well. In other cases, it is only possible to compare the situation with and without veto attribute. However, significant numbers of power indices are in use for evaluation of power of player with special emphasizing done for power of veto (for example: Bertini *et al.* 2012, Chessa, Fragnelli 2012, Mercik 2009, 2011), but there is no convincing arguments for choice of one or another power index. The main differences between these indices are the ways in which coalition members share the final outcome of their cooperation and the kind of coalition players choose to form. In this paper we would like to examine the base of such choice in axiomatic way.

2 Example of Veto Game

In Poland all bills are resolved if:

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 192–200, 2012.

- An absolute majority of representatives and the president are for[1], or
- In the case of a veto by the president, at least 3/5 of the representatives are for[2].

In the case of Polish parliamentary bills acceptation process we may see that generally there is cooperative game schema where the president, the *Sejm* and the Senate must form a coalition for to accept a bill. Therefore a given power index may be in use directly to evaluate their influence on the legislation process with an idea that greater value of the index represents greater influence on the process. The overview of possible indices one may find for example in Hołubiec and Mercik (1994).

3 Basic Notions

Let N = {1,2,…,n} be the set of players. A game on N is given by a map $v: 2^N \to R$ with $v(\varnothing) = 0$. The space of all games on N is denoted by G. A coalition $T \in 2^N$ is called a carrier of v if $v(S) = v(S \cap T)$ for any $S \in 2^N$.

The domain $SG \subset G$ of *simple games* on *N* consists of all $v \in G$ such that

(i) $v(S) \in \{0,1\}$ for all $S \in 2^N$;

(ii) $v(N) = 1$;

(iii) v is monotonic, i.e. if $S \subset T$ ***then*** $v(S) \le v(T)$.

A coalition S is said to be winning in $v \in SG$ if $v(S) = 1$, and losing otherwise. Therefore, the voting upon a bill is equivalent to formation of a winning coalition consists of voters.

A simple game *(N, v)* is said to be proper, if and only if it is satisfied that for all $T \subset N$, if $v(T)=1$ then $v(N \setminus T)=0$.

Consider a simple game *(N, v)* and a coalition $S \subset N$. *(N, v)* is said to be **S**-unanimous, if and only if it is satisfied that v(T)=1 if and only if $T \supset S$.

4 The Sense of Veto

The meaning of veto can be explained by the following artificial example: {2; 1_a, 1_b, 1_c} where the voting is a majority voting (voting quota equals 2) and weights of all voters *a, b, c* (*N=3*) are equal and fixed at 1. As it can be seen, there are four winning

[1] One may notice that the Polish *Senate* has no effective influence during the legislative process. The *Sejm* may reject the objections of the Senate at any moment by a simple majority, i.e. 231 deputies when all of them are present (460). Usually in the a priori analysis we only consider simple majority winning coalitions.

[2] This is a slightly simplified model, because the Supreme Court may also by simple majority recognize the bill as contradicting the Constitutional Act (or both chambers may change the Constitutional Act itself).

coalitions: {a,b}, {a,c}, {b,c}, {a,b,c}. The first three coalitions are vulnerable and the veto (called the veto of the first degree (Mercik, 2011)) of any coalition's members transforms it from winning into non-winning one. The classical example of such a veto is a possible veto of permanent members of the Security Council of the United Nations.

The last coalition, {a,b,c}, is different: a single member's veto can be overruled by two other members. This type of veto is called the second degree veto. A very typical example of such a veto is a presidential veto (at least in Poland or USA, for example), which under certain circumstances can be overruled.

A coalition structure $P = \{P_1, P_2, \ldots, P_m\}$ over N is a partition of N, that is $\bigcup_{k=1}^{m} P_k = N$ and $P_k \cap P_h = \emptyset$ when $k \neq h$. A coalition structure with veto $Pv = \{P_1, \ldots, \{j\}, \ldots, P_m\}$ over N for $j=1, m$ is a coalition structure P where at least one union is a singleton and at least one of the singletons is attributed with veto. The veto can be of the first or the second degree type.

The Example

One possible partition of Security Council of the UN's members:

$$P = \{\{P_1\}, \{P_2\}, \{P_3\}, \{P_4\}, \{P_5\}, \{NP_6, \ldots, NP_{15}\}\},$$

where each permanent member P_i has veto attribute and the rest of SC's members create a coalition. Of course, different combinations of partitions are also possible.

A power index is a mapping $\varphi: SG \rightarrow R^n$. For each $i \in N$ and $v \in SG$, the i^{th} coordinate of $\varphi(v) \in R^n$, $\varphi(v)(i)$, is interpreted as the voting power of player i in the game v. In the literature there are two dominating power indices: the Shapley-Shubik power index and the Banzhaf power index. Both base on the Shapley value concept [3]. The Shapley (1953) value is the value $\varphi: G \rightarrow R^n$, $v \rightarrow (\varphi_1(v), \varphi_2(v), \ldots, \varphi_n(v))$ where for all $i \in N$

$$\varphi_i^{SS}(v) = \sum_{S \subset N, i \notin S} \frac{s!(n-s-1)!}{n!} [v(S \cup \{i\}) - v(S)] \quad (1)$$

The Shapley-Shubik power index for simple game (Shapley, Shubik 1954) is the value $\varphi: SG \rightarrow R^n$, $v \rightarrow (\varphi_1(v), \varphi_2(v), \ldots, \varphi_n(v))$, where for all $i \in N$

$$\varphi_i^{SS}(v) = \sum_{S \subset N, i \notin S} \frac{s!\,(n-s-1)!}{n!} \quad (2)$$

The Banzhaf power index (Banzhaf, 1965) for simple game [4] is the value $\varphi: SG \rightarrow R^n$, $v \rightarrow (\varphi_1(v), \varphi_2(v), \ldots, \varphi_n(v))$ where for all $i \in N$

[3] The overview of the discussion about both power indices one may find in Laruelle, Valenciano (2000), Turnovec *et al.* (2004, 2008).

[4] This power index is called also as Banzhaf-Penrose power index. The Penrose's work from 1946 presented an analogue attempt to the concept of power for simple games.

$$\varphi_i^B(v) = \frac{1}{2^{n-1}} \sum_{S \subseteq N\{i\}} [v(S \cup \{i\}) - v(S)] \quad (3)$$

If one applies a partition structure P then Shapley-Shubik power index may be defined as following (Alonso-Meijide *et al.* 2007):

$$\varphi_i^{SS}(v, P) = \sum_{R \subseteq M\{k\}} \sum_{T \subseteq P_k\{i\}} \frac{(r+t)!(m+p_k-r-t-2)!}{(m+p_k-1)!} [v(Q \cup T \cup \{i\}) - v(Q \cup T)] \quad (4)$$

for all $i \in N$ and all (N, v, P) being a game with partition structure, where $P_k \in P$ is the union such that $i \in P_k$, $Q = \cup_{r \in R} P_r$.

An analogue definition of the Banzhaf power index for a game with partition structure can be formulated as following (Owen, 1982):

$$\varphi_i^B(v, P) = \sum_{R \subseteq M\{k\}} \sum_{T \subseteq P_k\{i\}} \frac{1}{2^{m-1}} \frac{1}{2^{p_k-1}} [v(Q \cup T \cup \{i\}) - v(Q \cup T)] \quad (5)$$

for all $i \in N$ and all (N, v, P) being a game with partition structure, where $P_k \in P$ is the union such that $i \in P_k$, $m = \| M \|$, $p_k = \| P_k \|$ and $Q = \cup_{r \in R} P_r$.

Both power indices formulated for games with partition structure give us the opportunity to represent such decisive bodies as parliaments, parliament-president or so. Especially in a parliament the partition structure is evident when party system and block voting are observed.

As we may conclude at this stage of the proceeding the available solutions for the power of veto measuring maybe formulated as the following algorithm:

- Re-arrange partition structure including the logic of veto, and
- Apply any power index.

The main problem: what power index is "the best one" is still open. There is a huge of papers in literature on domination of a given index over all or some others, but in no one paper there is a reference to veto itself. Moreover, we think that veto changes axioms being in the background of power indices and we face the problem of how to define power of veto and how to measure this power.

5 Axiomatizing Veto

The idea of power index for partition structure with veto strongly depends on kind of veto (Mercik, 2011): the veto of the first degree as the one which cannot be overruled (Security Council of the UN is a good example of such type of veto) and the veto of the second degree as the one which can be overruled, as for the President of the United States or the President of Poland.

In the literature (for example in Kitamura, Inohara, 2009) there is a concept of blockability as ability to block the final result in voting. Let's first check whether veto is equivalent to blocking.

Blockability Principle (Kitamura, Inohara (2009))
Consider a simple game *(N, v)*. For coalitions *S* and *S'* $S \gtrsim^b S'$ is defined as: for all $T \in W(v)$, if T\S' $\notin W(v)$ then $T \backslash S \notin W(v)$. $\gtrsim^b$ is called the blockability relation on *(N, v)*.

As we may see the blockability principle is fulfilled only for veto of the first degree. A veto of the second degree may not fulfil this principle. So, blockability principle is stronger than veto: $(N, v, \gtrsim^b) \subseteq (N, v, veto)$, and, blockability may be not equivalent to veto.

In trying to axiomatise an index one looks for "natural" principles that an index should satisfy and then obtain the particular index as the unique solution satisfying these principles if such index exists.

The following axioms are widely accepted:

Axiom 1. (Value-added)

$$\varphi(v)(i_{veto}) \geq \varphi(v)(i) \tag{6}$$

For veto of the first degree one gets strong inequality $\varphi(v)(i_{veto}) > \varphi(v)(i)$ (some of possible winning coalitions of the others may-be prohibited). What more, $\varphi(v)(i_{veto}) - \varphi(v)(i)$ one may call a net value of veto.

For veto of the second degree, $\varphi(v)(i_{veto}) \geq \varphi(v)(i)$ holds. For example, veto of Polish Senate (if it is a case) can be overruled in all circumstances.

Axiom 2. (Gain-loss: GL axiom)

$$\varphi(v)(i) > \varphi(w)(j) \tag{7}$$

for some $v, w \in SG$ and $i \in N$, then there exists $j \in N$ such that $\varphi(v)(j) < \varphi(w)(j)$.

If we introduce veto, the "gain-loss" axiom looks like $\varphi(v)(i_{veto}) > \varphi(w)(j)$ for some $v, w \in SG$ and $i_{veto} \in N$, then there exists $j \in N$ such that $\varphi(v)(j) < \varphi(w)(j)$. We simply assume that right to veto may potentially increase value of power index for a given player. In that case someone else must lose some of its power. Axiom GL is weaker than efficiency and quantitatively less demanding. It specifies neither the identity of *j* that loses power on account of *i*'s gain, nor the extend of *j*'s loss.

Axiom 3 (Efficiency)

$$\sum_{i \in N} \varphi(v)(i) = 1 \tag{8}$$

for every $v \in SG$ with coalition structure with veto.

For coalition structure with veto $v, w \in SG$ ***define*** $v \wedge w, v \vee w \in SG$ by:

$$(v \vee w)(S) = \max\{v(S), w(S)\},$$
$$(v \wedge w)(S) = \min\{v(S), w(S)\}$$

for all $S \in 2^N$. It is evident that *SG* is closed under operations $\wedge, \vee$. Thus a coalition is winning in $v \vee w$ if, and only if, it is winning in at least one of *v* or *w*, and it is winning in $v \vee w$ if, and only if, it is winning in both *v* and *w*.

Axiom 4 (Transfer)

$$\varphi(v \vee w) + \varphi(v \wedge w) = \varphi(v) + \varphi(w) \tag{9}$$

for $v, w \in SG$. This axiom stays same for SG with or without veto.

Axiom 4'. (Transfer – Dubey *et al.* 1981).
Consider two pairs of games *v, v'* and *w, w'* in SG with coalition structure with veto and suppose that the transitions from *v'* to *v* and *w'* to *w* entail adding the same set of winning coalitions, i.e. $v \geq v', w \geq w'$, ***and*** $v - v' = w - w'$. Equivalent transfer axiom: $\varphi(v) - \varphi(v') = \varphi(w) - \varphi(w')$, i.e. that the change in power depends only on the change in the voting game.

Denote by $\Pi(N)$ the set of all permutations of N (i.e., bijections $\pi : N \to N$). For $\pi \in \Pi(N)$ and a game $v \in SG$, define $\pi v \in SG$ by $(\pi v)(S) = v(\pi(S))$ for all $S \in 2^N$. The game πv is the same as *v* except that players are relabelled according to π.

Axiom 5 (Symmetry)

$$\varphi(\pi v)(i) = \varphi(v)(\pi(i)) \tag{10}$$

for every $v \in SG$ with coalition structure with veto, every $i \in N$ (including i_{veto}) and every $\pi \in \Pi(N)$.

According to symmetry, if players are relabelled in a game, their power indices will be relabelled accordingly. Thus, irrelevant characteristics of the players, outside of their role in the game *v*, have no influence on the power index.

It seems obvious that introduction of a coalition structure with veto will not demolish this axiom.

Axiom 5' (Equal Treatment)
If $i, j \in N$ are substitute players in the game $v \in SG$ with veto, i.e. for every $S \subset N \setminus \{i, j\}$ $v(S \cup \{i\}) = v(S \cup \{j\})$, then $\varphi(v)(i) = \varphi(v)(j)$.

Axiom 6 (Null player)
If $i \in N$, and *i* is null player in *v*, i.e. $v(S \cup \{i\}) = v(S)$ for every $S \subset N \setminus \{i\}$, then $\varphi(v)(i) = 0$.

The null player cannot be attributed with veto. Otherwise, from "Added value axiom" we get $\varphi(v)(i_{veto}) \geq 0$ what may violate "Null player axiom". To some extend the Supreme Court is the null player with veto attribute. In the example of legislative way in Poland, the Supreme Court doesn't form a coalition with other sides of

legislative process (it is independent by definition) but may stop the process if legal contradictories are found.

Axiom 7 (Dummy)
If $v \in SG$, and i is a dummy player in v, i.e. $v(S \cup \{i\}) = v(S) + v(\{i\})$ for every $S \subset N \setminus \{i\}$, then $\varphi(v)(i) = v(\{i\})$.

Dummy axiom implies that $\sum_{i \in N} \varphi(v)(i) = 1$ in every game $v \in SG$ where all players are dummies. The dummy player cannot be attributed with veto. Otherwise, from "Added value axiom" we get $\varphi(v)(i_{veto}) \geq 0$ what may violate "Null Player Axiom" and "Dummy Axiom".

Axiom 8 (Local monotonicity)
LM requires that a voter i who controls a larger share of vote cannot have a smaller share of power than a voter j with a smaller voting weight. This axiom may not be applied for simple games with veto structure.

A voter i is called "at least as desirable as" voter j if for any coalition S such that the union of S and $\{j\}$ is winning coalition, the union of S and $\{i\}$ is also winning (LM is an implication of desirability).

Axiom 9 (Desirability with veto)
A voter i is called "at least as desirable as" voter j if for any coalition S such that the union of S and $\{j\}$ is winning coalition with at least one member with veto power, the union of S and $\{i\}$ is also winning and at least one member has veto power too.

Summing the conclusions from the above analysis of axioms we may say that at least three axioms may not be applied to simple games with coalitional structure with veto, namely: null player axiom, dummy player axiom and local monotonicity axiom. However, the last one can be replaced by the axiom temporarily called "desirability with veto".

In the paper (Einy, Haimanko, 2010) one can find the following two theorems**:**

Theorem 1: There exists one, and only one, power index satisfying Gain-Loss, Transfer, Symmetry and Dummy, and it is Shapley-Shubik power index.

Theorem 2: There exists one, and only one, power index satisfying Gain-Loss, Transfer, Equal Treatment and Dummy, and it is Shapley-Shubik power index.

It is obvious that both above theorems are not valid for simple games with coalitional structure with veto. In consequence, it may exclude Shapley-Shubik power index from the list of potential power indices for such cases, i.e. simple (voting) games where veto is applied. Probably, the same conclusion maybe formulated for Banzhaf power index too. It makes the problem of measuring power for decision making via voting where veto maybe applied still unsolved and using of Johnston power index is on intuitive base only.

6 Conclusions

The analysis of axioms connected with power indices for simple games with coalitional structure with veto leads to the following results: (1) Not all classical axioms maybe assumed for simple games with coalitional structure with veto, (2) Intuitive choice of Johnston power index for simple (voting) games with coalitional structure with veto is still valid, (3) The problem of finding the one and only one power index for simple (voting) games with coalitional structure with veto is still an open case.

References

Alonso-Meijide, J.M., Carreras, F., Fiestras-Janeiro, M.G., Owen, G.: A comparative axiomatic characterization of the Banzhaf-Owen coalitional value. Decision Support Systems 43, 701–712 (2007)

Banzhaf III, J.F.: Weighted voting doesn't work: a mathematical analysis. Rutgers Law Review 19, 317–343 (1965)

Bertini, C., Freixas, J., Gambarelli, G., Stach, I.: Comparing Power Indices. In: Fragnelli, V., Gambarelli, G. (eds.) Some Open Problems in Applied Cooperative Games - A Special Issue of the International Game Theory Review (forthcoming, 2012)

Chessa, M., Fragnelli, V.: Open problems in veto theory. In: Fragnelli, V., Gambarelli, G. (eds.) Some Open Problems in Applied Cooperative Games - A Special Issue of the International Game Theory Review (forthcoming, 2012)

Dubey, A., Neyman, R.J., Weber, R.J.: Value theory without efficiency. Mathematics of Operations Research 6, 122–128 (1981)

Einy, E., Haimanko, O.: Characterization of the Shapley-Shubik power index without efficiency axiom, Discussion Paper 10-04, Monaster Center for Economic Research, Ben-Gurion University of the Negev (2010)

Hołubiec, J.W., Mercik, J.W.: Inside Voting Procedures, Accedo Verlagsgesellschaft, Munich (1994)

Kitamura, M., Inohara, T.: A characterization of the com-pleteness of blockability relation with respect to unanimity. Applied Mathematics and Computation 197, 715–718 (2008)

Kitamura, M., Inohara, T.: An extended Power Index to Evaluate Coalition Influence Based on Blockability Relations on Simple Games. In: Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio (2009)

Laruelle, A., Valenciano, F.: Shapley-Shubik and Banzhaf indices revisited. IVIE Working Paper V-114-2000 (2002)

Mercik, J.W.: A priori veto power of the president of Poland. Operations Research and Decisions 4, 141–150 (2009)

Mercik, J.: On a Priori Evaluation of Power of Veto. In: Herrera-Viedma, E., García-Lapresta, J.L., Kacprzyk, J., Fedrizzi, M., Nurmi, H., Zadrożny, S. (eds.) Consensual Processes. STUDFUZZ, vol. 267, pp. 145–156. Springer, Heidelberg (2011)

Owen, G.: Modification of the Banzhaf-Coleman Index for Games with a Priori Unions. In: Holler, M.J. (ed.) Power, Voting and Voting Power, pp. 232–238. Physica Verlag, Wurzburg-Wien (1982)

Penrose, L.S.: The Elementary Statistics of Majority Voting. Journal of the Royal Statistical Society 109, 53–57 (1946)

Shapley, L.S.: A Value for n-person Games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games. Annals of Mathematical Studies, vol. 28, pp. 307–317. Princeton University Press (1953)

Shapley, L.S., Shubik, M.: A method of evaluating the distribution of power in a committee system. American Political Science Review 48(3), 787–792 (1954)

Turnovec, F., Mercik, J., Mazurkiewicz, M.: Power indices: Shapley-Shubik or Penrose-Banzhaf? In: Kulikowski, R., Kacprzyk, J., Słowiński, R. (eds.) Operational Research and Systems 2004. Decision Making. Methodological Base and Applications, pp. 121–127. Exit, Warszawa (2004)

Turnovec, F., Mercik, J., Mazurkiewicz, M.: Power indices methodology: decisiveness, pivots and swings. In: Braham, M., Steffen, F. (eds.) Power, Freedom, and Voting. Springer, Heidelberg (2008)

STRoBAC – Spatial Temporal Role Based Access Control

Kim Tuyen Le Thi[1], Tran Khanh Dang[1], Pierre Kuonen[2], and Houda Chabbi Drissi[2]

[1] HCMC Univerisity of Technology, VNU HCMC, Ho Chi Minh City, Vietnam
{tuyenltk,khanh}@cse.hcmut.edu.vn
[2] College of Engineering and Architecture, Fribourg, Switzerland
{pierre.kuonen,Houda.Chabbi}@hefr.ch

Abstract. The development of geography-based services and systems has created the demands in which access control is the primary concern for geospatial data security. Although there are a variety of models to manage geospatial data access, none of them can fulfil the access control requirements. The objective of this paper is to propose a model that can support both spatio-temporal aspects and other contextual conditions as well as access control based on the role of subject. We call this model Spatial Temporal Role Based Access Control (STRoBAC). In addition, we propose an extension of GeoXACML framework, which is highly scalable and can help in declaring and enforcing various types of rules, to support the proposed model. This is the crucial contribution of our research compared to the existing approaches and models.

Keywords: Access Control Model, GeoXACML, GIS Database, OrBAC, Spatio-temporal Data, STRBAC, STRoBAC.

1 Introduction

Geographic Information System (GIS) is the technology integration of hardware, software, and data for capturing, managing, analyzing, and displaying geographical information (e.g. buildings, streets, cities) [23]. Its applications have been widely developed and used in many fields (e.g. land and resource management, market analysis, geographic data browsing) [19]. However, the rapid growth of geography-based services and systems may pose serious threats to national security and personal privacy. Furthermore, geospatial data is very sensitive, especially locations of government buildings, military camps, etc. [1]. Therefore, the need to build an access control model to protect geographical data and prevent unauthorized dissemination is really necessary and urgent.

The current models for geospatial data are often based on the characteristics of geospatial objects or subjects (e.g. feature type, specific area of object, location of requester, his/her roles within the valid area, etc.). Nevertheless, they do not provide the specification and enforcement of security policies supporting fully the kinds of restrictions. For instance, GeoXACML [10], SDE [9] or View

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 201–211, 2012.

based [9] access control do not mention about role or spatial characteristics of subjects. Meanwhile, GeoRBAC [3] proposes the access control based on the assigned relations between spatial roles, privileges and users. Notable are STRBAC [7] provides the concept of spatio-temporal aspects, and Or-BAC [2] proposes the way of expressing contextual conditions (including spatial and temporal conditions). However, they are just ideas and do not propose specific solutions for implementing. The remainder of this paper represents related concepts, details about STRoBAC model and an extension of GeoXACML to support this model.

2 Related Works

2.1 Spatial Temporal Role Based Access Control (STRBAC)

STRBAC is an extension of RBAC (Role Based Access Control) model that supports temporal and location constraints [7]. For the spatial aspect, STRBAC uses the definition of raw location (or physical location which can be the signal from the user's mobile, infra-red sensor or GPS device) and logical location. STRBAC limits the access of resources from a particular set of pre-defined locations: Location-by-Address (according to physical location), Location-by-Use (associate location with its usage), Location-by-Organization (distinguish location based on the organization level), and User-Defined Location (for other purposes). For the temporal aspect, STRBAC proposes particular time intervals. Time is described by particular date is called non-recurring interval (e.g. each date has the start date and end date). On the other hand, recurring interval is represented by particular interval (e.g. daily, weekly, monthly and yearly).

In summary, STRBAC combines both spatio-temporal factors into access control data; however, it is just an idea and not provides the restrictions on user-role binding. Another issue is deactivating user role automatically as soon as he/she moves to the place where his/her role does not satisfy the constraints.

2.2 Organization Based Access Control (Or-BAC)

OrBAC defines permissions (or obligations, prohibitions) that apply within *organization* to control the *activities* performed by *roles* on *views* [2]. And hence, to activate a given access control, the subject must be assigned to a given role, the object must be used in a given view and the action must partake in some activities. Beside these conditions, there are *extra conditions* that are called *context*, such as spatial or temporal requirements and other contextual conditions.

There are eight basic sets of entities in Or-BAC: Org (a set of organization), S (a set of subjects), A (a set of actions), O (a set of objects), $\mathcal{R}$ (a set of roles), $\mathcal{A}$ (a set of activities), $\mathcal{V}$ (a set of views) and $\mathcal{C}$ (a set of contexts). An access control policy can be represented by a set of rules having the following forms:

$$\forall s, \forall \alpha, \forall o, (Condition \rightarrow Is\_permitted(s, \alpha, o))$$

which means every subject $s \in S$ is permitted to perform action $\alpha \in A$ on object $o \in O$, if a given condition is satisfied (similarly for $Is\_prohibited$, $Is\_obliged$ and $Is\_dispensed$). The determinant in the form is $Condition$ which is defined by the conjunction of variety kinds of condition and constraints:

$cond\_subject(s) \wedge cond\_action(\alpha) \wedge cond\_object(o) \wedge constraint(s, \alpha, o)$

where:

$cond\_subject(s)$ is condition of subject s (s is empowered in a role or not). Or-BAC provides the built-in predicate $Empower$ to represent this condition over domains $Org \times S \times \mathcal{R}$. If org is organization, s is subject and r is role, $Empower(org, s, r)$ means that org empowers s in r.

$cond\_action(\alpha)$ is condition of action α. Similar to $Empower$ predicate, over domains $Org \times A \times \mathcal{A}$, predicate $Consider(org, \alpha, a)$ means that org considers action α implements activity a.

$cond\_object(o)$ is condition of object o. Over domains $Org \times O \times \mathcal{V}$, predicate $Use(org, o, v)$ means that org uses object o in view v.

$constraint(s, \alpha, o)$ is extra condition (or context) that joins subject s, action α and object o. Satisfying the constraint is necessary to active the rule. Over domains $Org \times S \times A \times O \times \mathcal{C}$, predicate $Hold(org, s, \alpha, o, c)$ means that within organization org, context c holds between s, α and o.

Let us consider the rule "*a person who works in the Department of Defence is permitted to access the important defensive position on specific map layer if he is standing in his office*", which $cond\_subject(s)$, $cond\_action(\alpha)$ and $cond\_object(o)$ respectively is: s is a person who works in the Department of Defence, α is an action of accessing and o is a specific map layer. Meanwhile, $constraint(s, \alpha, o)$ is position of s must be within the area of s's office. Now, the policy can be modelled by the general rule Concrete Permission Derivation:

$$\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall r \in \mathcal{R}, \forall a \in \mathcal{A}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C}$$
$$Permission(org, r, \alpha, v, c) \wedge Empower(org, s, r) \wedge Use(org, o, v) \wedge$$
$$Consider(org, \alpha, a) \wedge Hold(org, s, \alpha, o, c) \rightarrow Is\_permitted(s, \alpha, o)$$

that is in the organization org, if (1) role r has permission to perform activity a on view v within context c, and (2) org empowers subject s in role r, and (3) org uses object o in view v, and (4) org considers that action α implements activity a, and (5) context c holds between s, α, and o, then s is permitted to perform action α on object o. Three similar general rules respectively called Concrete Prohibition, Obligation and Dispensation Derivation are used to derive instances of $Is\_prohibited$, $Is\_obliged$ and $Is\_dispensed$.

In addition, Or-BAC defines context algebra to build conjunctive ($c_1 \& c_2$), disjunctive ($c_1 \oplus c_2$) and negative ($\overline{c}$) contexts from elementary contexts.

$$\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall c \in \mathcal{C}, \forall c_1 \in \mathcal{C}, \forall c_2 \in \mathcal{C},$$
$$Hold(org, s, \alpha, o, c_1 \& c_2) \leftarrow (Hold(org, s, \alpha, o, c_1) \wedge Hold(org, s, \alpha, o, c_2))$$
$$Hold(org, s, \alpha, o, c_1 \oplus c_2) \leftarrow (Hold(org, s, \alpha, o, c_1) \vee Hold(org, s, \alpha, o, c_2))$$
$$Hold(org, s, \alpha, o, \overline{c}) \leftarrow \neg(Hold(org, s, \alpha, o, c)$$

In summary, although Or-BAC supports many different types of condition (more details are presented in [2]), beside the weakness that not propose a specific solution to implement the model, Or-BAC is sometimes not possible to express all the possible conditions. For instance, the medical context of urgency, there are many different possibilities so that it is actually impossible to provide an exhaustive definition of such a context. Next section will present a brief introduction about XACML and GeoXACML, the most suitable declaration and enforcement language for implementing access control model.

2.3 eXtensible Access Control Markup Language (XACML)

XACML is an standard of Organization for the Advancement of Structured Information Standards (OASIS), that describes both policy and access control decision request/response language [12]. The policy language is used to describes general access control requirements with standard extension points (functions, data types, combining logic, etc.). Meanwhile, the request/response language allows user to form a query asking a given action should be allowed or not. There are some basic elements of XACML can be mentioned: *PolicySet* contains other Policies or PoliciySets; *Policy* expresses a single access control policy through a set of *Rules*; *Target* element is used to find a policy that can apply to a given request; *Attribute* is named value of know type and represents the characteristics of the objects; and *Attribute Values* is the specific value of *Attribute* (name of user, the file which user want to access, the time of day, etc.).

Differences between XACML Version 2.0 and 3.0. Compared with version 2.0, XACML 3.0 has several advantages [13]. Firstly, version 3.0 supports to express more flexible for *Target* element. The whole schema of core XACML 2.0 and 3.0 [15] [16] and an example about *Target* element [13] will provide a clearer view. Secondly, user can customize the categories to extend some other contextual conditions instead of using the final set of categories (e.g. emergency category). In addition, version 3.0 defines *Advice* element which is analogous to *Obligation*. The obligation and the value of obligation id and arguments are forced to interpret; meanwhile it is optional for advice. In other words, obligations were static in the sense that we could not convey the value of an attribute that may change at runtime. The next advantage is the ability to delegate administrative rights. Namely, a global administrator can delegate constrained administrative rights to local administrators. And finally, multiple aspects on any category can be allowed (version 3.0), instead of only multiple resources (version 2.0). For instance, it is possible to ask "*Can I view resource 1 and can I view resource 2?*" in version 2.0 and "*Can I view and edit resource 1?*" in version 3.0, which the reply can be "*Permit to view and Deny to edit*". However, XACML 2.0 and all the associated profiles were approved as OASIS Standards on February 1, 2005 [14]; meanwhile version 3.0 still in progress. But with mentioned advantages, version 3.0 can support to describe policy and request/response more flexible and suitable to express the contextual conditions of the proposed model.

An Extended RBAC Profile of XACML. To support role based access control, XACML uses four specific types of policy as follow: $Role < PolicySet >$, $Permission < PolicySet >$, $HasPrivilegesOfRole < Policy >$, and $RoleAssignment < Policy >$. More details about these types of policy are represented in [17]. Notable is $RoleAssignment < Policy >$ is used by *Role Enablement Authority* (REA) that assigns various role attributes to users and enable them within a session. Based on this concept, the authors of [5] define additional four Enablement Authorities (EA). Namely, *ViewEA* (VEA, assigning objects to views), *ActivityEA* (AEA, deciding to use actions in activities), and *ContextEA* (CEA, evaluating contextual conditions). With this extension, the authors can use XACML to express the Or-BAC model as well as other models that based on RBAC. However, XACML does not support spatial data. Therefore, GeoXACML of Andreas Matheus [10] will be considered in the next section.

2.4 Geospatial eXtensible Access Control Markup Language

The first version of GeoXACML [10] provides a possible recommendation on how to declare and enforce access rights effectively and flexibly. According to the extension points of XACML, $< AttributeValue >$ is used to add new data types for GeoXACML by assigning the appropriate value to the *DataType* attribute, and ensuring the syntax corresponds with Geography Markup Language (GML) 2.1.2 definition (language is used for expressing geographical features) [11].

Beside, the declaration of spatial restrictions also requires spatial functions for testing the specific topological constellation between two geometries (e.g. within, touches, etc.). Furthermore, GeoXACML can support the basic spatial access restrictions based on rules (e.g. class-based, object-based and spatial restrictions). All examples for these kinds of restriction are represented in [10].

In addition, a prototype of GeoXACML has been implemented. However, it only has the component evaluates the access request with some basic functions for integrating spatial data and functions into XACML, and does not consider the restrictions about temporal characteristics, contextual information and user roles. Nevertheless, because GeoXACML uses XML encoding to express access rights, it can be flexible to add new tags or re-define the structure of files, as well as can be extended by adding new data types, functions, the components for processing temporal conditions, etc. The evidence is the current version of GeoXACML, version 1.0 [18] which supports many kinds of spatial functions. A general view about the mentioned concepts is presented in Table 1[1]; note that all the conditions are considered as contextual conditions and divided into three kinds of concerning objects: user, data, and other (for role and rule based).

In summary, the proposed model will support the following contextual conditions: (1) location of user (user spatial aspect), (2) the time when user send his request (user temporal aspect), (3) the spatial boundary of the resource (data spatial aspect), (4) role and rule based access control (other kind of objects),

[1] Spa. is Spatial; Temp. is Temporal; $\surd$ means has; _ means do not have; ? means unclear; UD is User Define; Ru is Rule and Ro is Role.

Table 1. A general view about the proposed model with contextual restrictions

	User		Data		Other	
	Spa.	Temp.	Spa.	Temp.	Spa.	Temp.
STRBAC	√	√	√	-	√	√
Or-BAC	√	√	√	-	√ + (UD)	√ + (UD)
XACML 3.0	-	√	-	?	-	√ + (Ru)
XACML 3.0 + RBAC	-	√	-	?	-	√ + (Ro + Ru)
GeoXACML 1.0	√	-	√	-	√ + (Ru)	-
STRoBAC	√	√	√	-	√ + (Ro + Ru)	√ + (Ro + Ru)

and finally (5) content of data (e.g. when and where data is stored) is optional. Next section will represent in detail about the proposed model, STRoBAC.

3 The Proposed Model: STRoBAC

STRoBAC model is developed from the work of [8], it based on STRBAC model to describe location and time information; express the contextual conditions based on Or-BAC and use GeoXACML extension based on XACML 3.0 associated with RBAC to implement the model.

Similar to Or-BAC model , STRoBAC has basic sets of entities: S (subjects), A (actions), O (objects), $\mathcal{R}$ (roles), $\mathcal{A}$ (activities), $\mathcal{V}$ (views) and $\mathcal{C}$ (contexts), (Org set will not be considered). Any entities in STRoBAC model may have some attributes. This is represented by predicates that associate the entities with the value of these attributes. For instance, if s is a subject, then $Work\_in(s, Department)$ will return true if s work in $Department$.

Let us back to the section 2.2 and 2.3 about the general rule of Or-BAC and extended RBAC profile of XACML. Mapping between two sections, the built-in predicates of Or-BAC ($Empower(org, s, r)$, $Use(org, o, v)$, $Consider$, and $Hold(org, s, \alpha, o, c)$) can be replaced by $REA(s, r)$, $VEA(o, v)$, $AEA(\alpha, a)$, and $CEA(s, \alpha, o, c)$, respectively. Furthermore, based on the core RBAC [7], permission includes which action is performed on which object. Therefore, predicate $Permission(org, r, \alpha, v, c)$ is replaced by a conjunctive of $Permission(p, a, v)$ and $PEA(p, r, c)$; where $Permission(p, a, v)$ defines permission p that perform activity a on view v and $PEA(p, r, c)$ assign permission p to role r within context c. Beside, unlike other models, STRoBAC model considers not only context holds between subject, action and object, but also the context of assigning subject in role, using object in view and considering action in activity. Therefore, context element c is added into each predicate. Namely, within context c: $REA(s, r, c)$ is true if subject s is assigned in role r, $VEA(o, v, c)$ is true if object o is used in view v and $AEA(\alpha, a, c)$ is true if action α is considered in activity a. The definitions of role, activity, view, context and policy are based on these predicates. A role definition corresponds to a logical rule that has REA predicate in the conclusion. Let us consider again the example in section 2.2 with condition when assign role to subject: "*a staff holds the Observer role in the Department of Defence is permitted to access the important defensive position on specific map*

layer, if he is standing in his office; senior staffs (staff works more than 5 years) have the working time between 7 – 11 AM can be assigned to Observer role".

$$\forall s \in S, \forall d \in O, REA(s, Observer, Working\_time) \leftarrow$$
$$Work\_in(s, DD) \wedge Working\_years(s, d) \wedge (d > 5) \wedge Working\_time(s, 7, 11)$$

where $Working\_years(s, d)$ and $Working\_time(s, 7, 11)$ respectively is true if d is the working year of subject s and working time of s is between 7 – 11 AM. Activity and view definition can be expressed in the similar way. Meanwhile, reuse the example above with additional temporal condition "*subject's request time must be between 7 – 11 AM*", the context definition (contextual conditions hold between subject, action and object) can be expressed as follow:

$$\forall s \in S, \forall \alpha \in A, \forall o, po \in O,$$
$$CEA(s, \alpha, o, Personal\_Office\&Request\_time) \leftarrow$$
$$(Personal\_office(s, po) \wedge Is\_located(s, po)) \wedge Request\_time(s, 7, 11)$$

where $Personal\_office(s, po)$ returns true if po is the personal office of subject s; $Is\_located(s, po)$ returns true if subject s is standing in his office and $Request\_time(s, 7, 11)$ is true if his request time is between 7 AM – 11 AM.

Now, the policy can be expressed by the components of STRoBAC as follow:

$$\forall s \in \mathrm{S}, \forall \alpha \in \mathrm{A}, \forall o \in \mathrm{O}, \forall r \in \mathcal{R}, \forall a \in \mathcal{A}, \forall v \in \mathcal{V}, \forall c, c_1, c_2, c_3, c_4 \in \mathcal{C},$$
$$REA(s, r, c_1) \wedge VEA(o, v, c_2) \wedge AEA(\alpha, a, c_3) \wedge PEA(p, r, c_4) \wedge$$
$$CEA(s, \alpha, o, c) \wedge Permission(p, a, v) \rightarrow Is\_permitted(s, \alpha, o)$$

where c can be one of c_1, c_2, c_3, c_4 or combination of them. The policy of above-mentioned example can be expressed as follow:

$$\forall s \in S, \forall \alpha \in S, \forall o \in O,$$
$$REA(s, Observer, Working\_time) \wedge VEA(o, Specific\_Map\_Layer) \wedge$$
$$AEA(\alpha, Access) \wedge PEA(Access\_Specific\_Layer, Observer) \wedge$$
$$CEA(s, \alpha, o, Personal\_Office\&Request\_time) \wedge$$
$$Permission(Access\_Specific\_Layer, Access, Specific\_Map\_Layer)$$
$$\rightarrow Is\_permitted(s, \alpha, o)$$

where, $Permission$ predicate defines permission $Access\_Specific\_Layer$ that performs activity $Access$ on view $Specific\_Map\_Layer$. The context of VEA, AEA and PEA are absent in this policy. The similar way is used to express the definition of $Is\_prohibited$, $Is\_obliged$ and $Is\_dispensed$.

Beside, the concept of spatial and temporal in STRBAC model (section 2.1) can be expressed by the predicates in STRoBAC model. Some basic predicates are used (similar to Or-BAC model): $Before\_Time$, $After\_Time$, $Before\_Date$, $After\_date$, On_Day, $Is\_Located$, $Location$, etc. For instance:

- Logical location: $Location$ and $Is\_Located$ predicate are used to determine location of subject and whether subject is located in a specific area or not.
- Non recurring interval: example "*between February 22, 2012 to February 28, 2012*" can be expressed in STRBAC is (2012/02/22 . . . 2012/02/28) and in STRoBAC is $After\_Date(2012/02/22)\&Before\_Date(2012/02/28)$

– Recurring Interval: example "*between 9 AM to 5 PM everyday*" can be expressed in STRBAC is $(09:00:00 \ldots 17:00:00)$ and in STRoBAC is $(After\_Time(09:00:00) \& Before\_Time(17:00:00))$

Because the logical location is always used to define location constraints [7], therefore the way to express physical location will not be considered. These examples also show that STRoBAC can support to express the mentioned contextual conditions in Table 1. Another model can be mentioned is X-STROWL[2] [21], which focus on integrates XACML with the OWL ontology for semantic reasoning on hierarchical roles and describes the general contextual constrains, instead of proposing the general rule for expressing policy with specific spatio-temporal constrains like the model in this paper. Next section will represent how to extend GeoXACML to support STRoBAC model.

4 Extend GeoXACML to Support STRoBAC

Both of GeoXACML and XACML use the same process of evaluating user's requests. Therefore, to support STRoBAC, some additional components will be added in the process, (e.g. REA, VEA, AEA, CEA and PEA). However, permission includes which action is performed on which object, hence, PEA will also includes VEA and AEA. Data-flow in Fig. 1 represents this extension. More details about this process is represented in the core specification [15] [16].

The process begins with the basic steps are performed similar to the first steps in XACML (1–5). The Context Handler will does the major part of work is collecting all of the necessary information and then return to PDP (23). For instance, Context Handler requests (6) and receives (19) a list of selected role from RoleEA. However, to evaluate which role is selected, RoleEA needs to know about the attributes of these roles (7). These requests are sent again from Context Handler to PIP (8). Then PIP obtains the attributes from the Repository (9) and returns them to the Context Handler (10). Beside, because REA predicate have context element c in the definition, hence Context Handler has to validate the context before returning attributes to RoleEA (11). Similar to RoleEA, ContextEA needs to know the context attributes, then step 12–15 are performed similar to step 7–10. After receives the necessary attributes (16), ContextEA validates the context and returns the result to Context Handler (17). Then, RoleEA receives the role attributes(18) and returns the list of selected roles to Context Handler (19). Note that the steps according to PermissionEA are performed in parallel (or the order does not matter) and totally similar to the steps of RoleEA. Beside, there are some other kinds of attributes can be requested (e.g. number of access requests from the log file). Therefore, Context Handler obtains them from PIP (20, 21, 22) and returns all of necessary attributes to the PDP (23). Furthermore, to evaluate the final context element in the general rule of STRoBAC (context holds between subject, action and object), the context has

[2] Similar idea with STRoBAC, but they are developed independently at the same time.

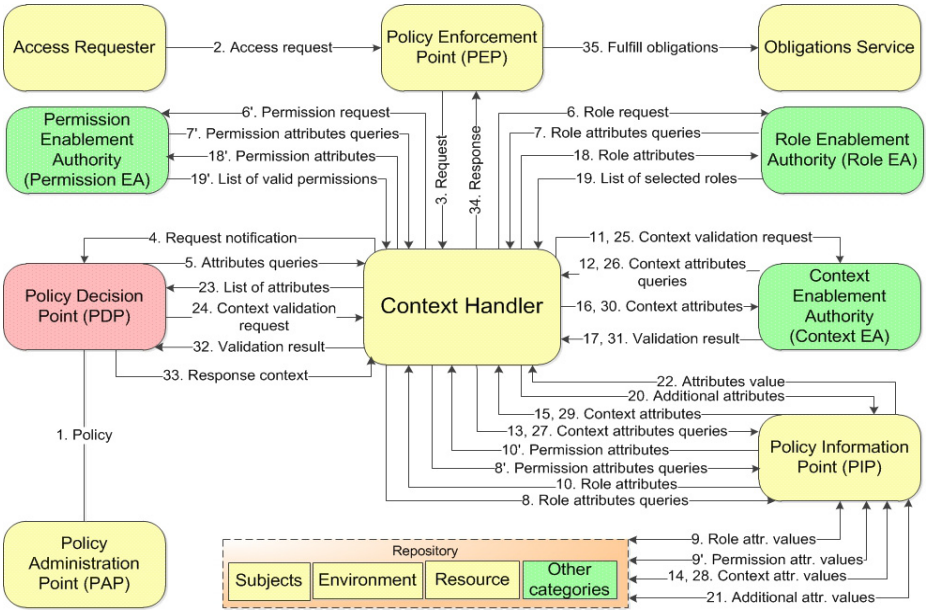


Fig. 1. Data-flow diagram of extended GeoXACML to support STRoBAC

to be evaluated again and returns the context validation result to PDP (24–32). The last steps (34, 35) are performed similar to steps in XACML.

In addition, to support STRoBAC model, spatial functions, data types and new identifiers/attributes have to be added in GeoXACML. However, beside the implementation of GeoXACML (based on Sun's XACML 1.1), [14] provides many open sources implement XACML. Notable is the implementation of Sun's XACML [20], HERAS-AF [6], XACMLight [22] and Enterprise Java XACML [4]. Nevertheless, Sun's XACML just supports XACML version 1.1, meanwhile the implementation of Enterprise Java XACML has not been updated for a long time and it does not have a clear manual. Compare with XACMLight, HERAS-AF supports more components and has the fuller manual. For these reasons, HERAS-AF will be chosen for the implementation of this research. More details about HERAS-AF as well as the way to extend it will be considered in the future.

5 Conclusion

In this paper, we proposed STRoBAC model that can support spatio-temporal aspects and other contextual conditions for GIS data. STRoBAC is the combination of using spatio-temporal concepts of STRBAC with the way of expressing contextual conditions in Or-BAC and the process of role, activity, view and context assignment, proposed in the extended RBAC profile of XACML. Furthermore, unlike other models, beside the context holds between subject, action and

object, we proposed other additional contexts (e.g. contexts of assigning subject in role, using object in view, and assigning action in activity). The paper also represents a way of extending GeoXACML to support STRoBAC model. More details about new functions, data types, identifiers/attributes and the implementation of GeoXACML that supports STRoBAC based on XACML 3.0 and HERAS-AF will be discussed in the future.

References

1. Chun, S.A., Atluri, V.: Geospatial Database Security. In: Gertz, M., Jajodia, S. (eds.) Hand Book of DB Security App. and Trends, pp. 247–248. Springer (2007)
2. Cuppens, F., Boulahia, N.C.: Modeling Contextual Security Policies. International Journal of Information Security 7(4), 285–305 (2008)
3. Damiani, M.L., Bertino, E., Catania, B., Perlasca, P.: GEO-RBAC: A Spatially Aware RBAC. ACM Trans. on Info. and System Security 10(1) (2007)
4. E.J. XACML (June 2012), http://code.google.com/p/enterprise-java-xacml/
5. Haidar, D.A., Cuppens-Boulahia, N., Cuppens, F., Debar, H.: An Extended RBAC Profile of XACML. In: 3rd ACM Workshop on Secure Web Services, pp. 13–22 (2006)
6. HERAS-AF (June 2012), http://www.herasaf.org/
7. Kumar, M., Newman, R.E.: STRBAC – An Approach Towards Spatio-Temporal Role-Based Access Control. In: Communication, Network and Information Security, USA, pp. 150–155 (2006)
8. Le, T.K.T., Tran, T.Q.N., Dang, T.K.: An Enhanced Access Control Model for GIS Database Security. In: 4th Regional Conference on Information and Communication Technology, Vietnam, pp. 129–136 (2011)
9. Lin, J., Fang, Y., Chen, B., Wu, P.: Analysis of Access Control Mechanisms for Spatial Database. In: ISPRS (2008)
10. Matheus, A.: Declaration and Enforcement of Access Restrictions for Distributed Geospatial Information Objects, Master Thesis, Fakultät für Informatik Technische Universität München (2005)
11. Matheus, A.: GeoXACML, A Spatial Extension to XACML. The Federal Armed Forces Germany Univ., Discussion paper 05-036 (June 16, 2005)
12. OASIS Brief Introduction to XACML (April 2012), http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html
13. OASIS Differences between XACML 2.0 and XACML 3.0 (April 2012), http://wiki.oasis-open.org/xacml/DifferencesBetweenXACML2.0AndXACML3.0
14. OASIS XACML (April 2012), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#CURRENT
15. OASIS XACML 2.0 Core Specification (April 2012), http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
16. OASIS XACML 3.0 Core Specification (April 2012), http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf
17. OASIS XACML 3.0 and Core Hierarchical Role Based Access Control (April 2012), http://docs.oasis-open.org/xacml/3.0/xacml-3.0-rbac-v1-spec-cs-01-en.pdf

18. OGC GeoXACML (April 2012), http://www.opengeospatial.org/standards/geoxacml
19. Sophat, S.: Fundamentals of Geographic Information Systems. Royal University of Phnom Penh (2007)
20. Sun's XACML (June 2012), http://sunxacml.sourceforge.net
21. Tran, T.Q.N., Dang, T.K.: X-STROWL: A Generalized Extension of XACML for Context-aware Spatio-Temporal RBAC Model with OWL. In: 7th International Conference on Digital Information Management, Macau (to appear, 2012)
22. XACMLight (June 2012), http://sourceforge.net/projects/xacmllight/
23. What is GIS (October 2011), http://www.gis.com/content/what-gis

Rescheduling of Concurrently Flowing Cyclic Processes

Grzegorz Bocewicz[1] and Zbigniew A. Banaszak[2]

[1] Koszalin University of Technology,
Dept. of Electronics and Computer Science, Koszalin, Poland
bocewicz@ie.tu.koszalin.pl
[2] Warsaw University of Technology, Faculty of Management,
Dept. of Business Informatics, Warsaw, Poland
Z.Banaszak@wz.pw.edu.pl

Abstract. The paper presents a declarative modeling framework enabling to evaluate the cyclic steady state of a given system of concurrently flowing cyclic processes (SCCP) on the base of the assumed topology of transportation routes, dispatching rules employed, resources and operation times as well as an initial processes allocation. The objective is to provide sufficient conditions guaranteeing rescheduling among cyclic schedules reachable in a given SCCP. The properties providing such conditions as well as illustrative examples are presented.

Keywords: cyclic processes, rescheduling, state space, declarative modeling.

1 Introduction

A cyclic schedule [1], [3] is one in which the same sequence of states is repeated over and over again. In everyday practice cyclic scheduling arise in different application domains (such as manufacturing, time-sharing of processors in embedded systems, and in compilers for scheduling loop operations for parallel or pipelined architectures) as well as service domains (covering such areas as workforce scheduling (e.g., shift scheduling, crew scheduling), timetabling (e.g., train timetabling, aircraft routing and scheduling), and reservations (e.g., reservations with or without slack) [2], [3], [5], [6]. Such cyclic scheduling problems belong to decision problems, and because of their integer domains belong to a class of Diophantine problems [4].

Consequently, not all the behaviors (including cyclic ones) are reachable under constraints imposed by system's structure. The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural constraints. So, the system structure configuration must be determined for the purpose of processes scheduling, yet scheduling must be done to devise the system configuration.

Many models and methods have been proposed to solve the cyclic scheduling problem [4]. Among them, the mathematical programming approach (usually IP and MIP), max-plus algebra [6], constraint logic programming [2], [7] evolutionary algorithms and Petri nets [1] frameworks belong to the most frequently used. Majority of them are oriented at finding of a minimal cycle or maximal throughput while assuming deadlock-free processes flow.

In that context our main contribution is to propose a declarative framework enabling modeling and performance evaluation of a system of concurrent cyclic processes (SCCP). The cyclic steady states space of a given system provides a formal

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 212–222, 2012.

framework enabling one to develop conditions sufficient for rescheduling of concurrently flowing cyclic processes. The following questions are of main interest: Does the assumed system behavior (e.g. a cyclic steady state) can be achieved under the given system's structure constraints? Does the assumed local processes cyclic steady state is reachable from another one?

2 Systems of Concurrent Cyclic Processes

2.1 Model

The example of Systems of Concurrent Cyclic Processes (SCCP) is shown on Fig. 1. In this kind of system the cyclic local processes are executed on the resources along the given processes routes.

In considered case six **local cyclic processes** are considered P_1, …, P_6. The processes follow the **routes** composed of resources $R = \{R_1, \dots, R_c, \dots, R_{12}\}$, ($R_c$ - the c-th resource). In general case processes may contain many sub-processes (streams P_i^k): $P_i = \left\{P_i^1, P_i^2, \dots, P_i^{ls(i)}\right\}$, i.e. processes moving along the same route. The local processes of Fig. 1 contain only unique streams: $P_1 = \{P_1^1\}$, $P_2 = \{P_2^1\}$,..., $P_6 = \{P_6^1\}$. Processes can interact each other through common shared resources, i.e. transportation sectors. Routes of local processes streams considered are as follows:

$$p_1^1 = (R_1, R_2, R_3, R_4),\ p_2^1 = (R_5, R_6, R_7, R_8),\ p_3^1 = (R_9, R_1, R_{12}, R_5),$$
$$p_4^1 = (R_2, R_{10}, R_8, R_9),\ p_5^1 = (R_3, R_{10}, R_7, R_{11}),\ p_6^1 = (R_4, R_{12}, R_6, R_{11}),$$

where: R_1,…, R_{12} are shared resources, since each one is used by at exactly two streams (i.e. R_3 is used by P_1^1, P_2^1).

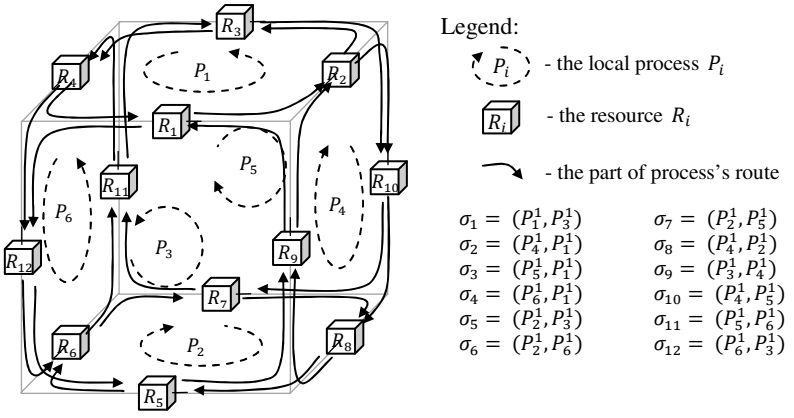


Fig. 1. The SCCP model

In that context, a SCCP can be stated as a set of processes $P = \left\{P_i = \left\{P_i^1, P_i^2, \dots, P_i^{ls(i)}\right\} | \ i = 1 \dots n\right\}$, containing streams P_i^k where each stream is characterized by a set of operations executed on the resources R along a given route p_i^k.

The class of the SCCP considered follows the constraints stated below [2]:

- the new operation of local processes streams may start on a resource only if the current operation has been completed and the resource has been released,
- the local processes streams share the common resources in the mutual exclusion mode, the local process operation can be suspended only if designed resource is occupied, the suspended local processes cannot be released, local processes are non-preempted, i.e. the resource may not be taken of a process till it is using it,
- the local processes are executed cyclically with periods Tc; resources occur uniquely in each transportation route,
- in a cyclic steady state, each the i-th stream has to pass its local route the same number of times $\Xi \cdot \psi_i$, the factors Ξ, ψ_i are defined below.

A resource conflict (caused by mutual exclusion protocol usage) is resolved with help of a priority dispatching rule [1] determining an order in which streams make their access to common shared resources (for instance, in case of the resource R_1, $\sigma_1 = (P_1^1, P_3^1)$ – the priority dispatching rule determines the order in which streams can access to the shared resource R_1, i.e. at first to the stream P_1^1, and next to the stream P_3^1, and so on). Each stream P_i^k has to occur the same number of times in each dispatching rule associated to resources appearing in its route. So, the SCCP shown in Fig. 1 is specified by the following set of dispatching rules $\Theta = \{\sigma_1, \ldots, \sigma_{12}\}$, as well as $f_1(P_1^1) = f_2(P_1^1) = f_3(P_1^1) = f_4(P_1^1) = 1$, $f_5(P_2^1) = f_6(P_2^1) = f_7(P_2^1) = f_8(P_2^1) = 1$ etc., where $f_c(P_i^k)$ – a number of P_i^k occurrences in the c-th priority dispatching rule. That means the each stream $(P_1^1, P_2^1, P_3^1, P_4^1, P_5^1, P_6^1)$ repeats only ones during the same period. It means the priority rules determine frequencies of mutual appearance of local processes sharing the same resource.

In general case, the set of dispatching rules Θ implies the sequence of relative frequencies of local processes mutual executions, and denoted by $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$, where: $\psi_i \in \mathbb{N}$,

$$\psi_i = \|\{b \mid crd_b\sigma_c = P_i^1 ; b \in \{1, \ldots, lp(c)\}\}\|, \quad \forall i \in \{1, \ldots, n\}, \forall \sigma_c \in \Theta_i, \quad (1)$$

where: Θ_i – the set of dispatching rules associated to resources occurring in the route followed by P_i, $crd_b\sigma_c$ - the b-th entry of the sequence σ_c, n – a number of processes, $lp(c)$ – the length of σ_c.

So, the SCCP shown in Fig. 1 is specified by the sequence: $\Psi = (1,1,1,1,1,1)$. That means one execution of local processes P_1 falls on one executions of rest processes $(P_2, P_3, P_4, P_5, P_6)$, and one execution of local processes P_2 falls on one executions of rest processes $(P_1, P_3, P_4, P_5, P_6)$, etc.

Since the sequence Ψ of relative frequencies of local processes mutual executions does not necessary encompass cyclic steady state of a SCCP, hence a new parameter describing the number of Ψ occurrences within a cyclic steady state, denoted by $\Xi \in \mathbb{N}$, is introduced. For the considered SCCP, the value $\Xi = 2$, means that two executions of the sequence $\Psi = (1,1,1,1,1,1)$, i.e., two executions of local process P_1, fall on two executions of the process P_2, P_3, P_4, P_5, P_6, etc.

In general case, the following notations are used:

- $p_i^k = (p_{i,1}^k, p_{i,2}^k, \ldots, p_{i,lr(i)}^k)$ – **the route of the local process's stream P_i^k** (k-th stream of the i-th local process P_i), and its components define the resources used in course of operations execution, where: $p_{i,j}^k \in R$ (the set of resources: $R = \{R_1, R_2, \ldots, R_c, \ldots, R_m\}$) denotes the resource used by the k-th stream of the i-th local process in the j-th operation; in the rest of the paper **the j-th operation executed on resource $p_{i,j}^k$ in the stream P_i^k** is denoted by $o_{i,j}^k$; $lr(i)$ - denotes the length of cyclic process route (each stream's route p_i^k of P_i has the same length).
- $x_{i,j,q}^k(l) \in \mathbb{N}$ – the moment the operation $o_{i,j}^k$ starts its q-th execution in the l-th cycle of the stream P_i^k.
- $t_i^k = (t_{i,1}^k, t_{i,2}^k, \ldots, t_{i,lr(i)}^k)$ – **the local process operation times**, where $t_{i,j}^k$ denotes the time of execution of operation $o_{i,j}^k$ ($t_{i,j}^k = 1$ are used in SCCP from Fig. 1).
- $\Theta = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ – the set of **the priority dispatching rules**, where $\sigma_c = (s_{c,1}, \ldots, s_{c,lp(c)})$ is the sequence components of which determine an order in which the processes can be executed on the resource R_c, $s_{c,d} \in H$ (the set of local process streams).

Using the above notation a SCCP with local processes can be defined as a pair:

$$SC = (R, SL), \tag{2}$$

where: $R = \{R_1, R_2, \ldots, R_m\}$ – the set of resources, m – the number of resources,
$SL = (ST_L, BE_L)$ – the local processes structure, i.e.

$ST_L = (U, T)$ – the variables describing layout of local processes,

$U = \left\{p_1^1, \ldots, p_1^{ls(1)}, \ldots, p_n^1, \ldots, p_n^{ls(n)}\right\}$ – the set of local process routes, $ls(i)$ – the number of streams belonging to the process P_i, n – a number of local processes,

$T = \left\{t_1^1, \ldots, t_1^{ls(1)}, \ldots, t_n^1, \ldots, t_n^{ls(n)}\right\}$ – the set of sequences of operation times in local processes.

$BE_L = (\Theta, \Psi, \Xi)$ – the variables describing the local processes behavior,

$\Theta = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ – the set of dispatching priority rules,

$\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$ – the sequence of relative frequencies of local processes mutual executions,

Ξ – the number of Ψ occurrences within a cyclic steady state.

The considered model (2) can be seen as a basic (i.e., a lowest) level in the multilevel (taking into account multimodal processes [2]) model of the SCCP [2].

Problem Statement

Consider the SCCP (2) specified by the given set R of resources, dispatching rules Θ, processes routes U, and initial processes allocation. The main question concerns of SCCP periodicity, i.e. does the cyclic execution of local processes exist? In case when they are periodic the another question can be stated: What is the period Tc? The other questions regard of multimodal processes cyclic execution.

The problems stated above have been studied in [2], [7], where for solution the constraints programming techniques have been successfully employed. For instance using the method submitted in [2],the 25 cyclic behaviors of the SCCP from Fig. 1 can be determined.

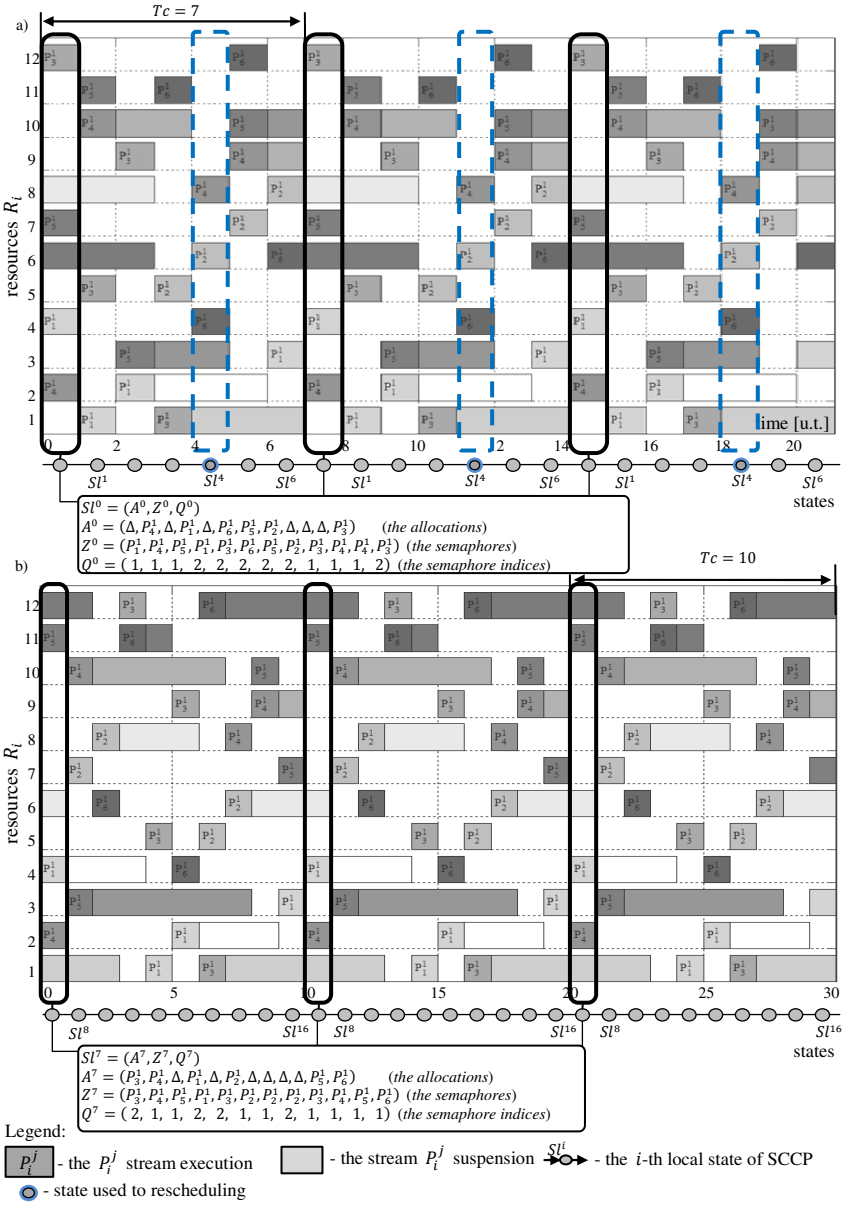


Fig. 2. The example of two available behaviors of SCCP form Fig. 1: with a period $Tc = 7$ a) and a period $Tc = 10$

For instance, two cyclic behaviors specified by cycles $Tc = 7$ and $Tc = 10$, respectively are shown in Fig. 2.

A new problem regarding possible switching among cyclic steady states can be seen as their obvious consequence. In that context, the newly arising questions are: Is it possible to reschedule cyclic schedules as to "jump" from one cyclic steady state to another one? For example: is it possible to "jump" from the cyclic steady state behavior described by the Gantt's chart shown in Fig. 2a) to the second one specified by the Gantt's chart shown in Fig. 2b)? Is it possible to "jump" directly or indirectly? What are the control rules allowing one to do it? These kind of questions are of crucial importance for manufacturing and transportation systems aimed at short run production and/or passengers itinerary (e.g. in a sub-way network) planning.

3 States Space

The Gantt's charts from Fig. 2 provide the graphical illustration of modeled SCCP cyclic behaviors. However, associating to each column of the Gantt's chart the state distinguished by "○" (see Fig. 2) the cyclic behavior can be treated in terms of periodically executed set of such distinguished states.

Let us consider the following SCCPs state definition describing processes allocation:

$$Sl^r = (A^r, Z^r, Q^r), \tag{3}$$

where: Sl^r – is the state of local processes,

$A^r = ({a_1}^r, {a_2}^r, \ldots, {a_m}^r)$ – the processes allocation in the r-th state, ${a_c}^r \in P \cup \{\Delta\}$, $a_c = P_i^k$ – the c-th resource R_c is occupied by the local stream P_i^k, and ${a_c}^r = \Delta$ – the c-th resource R_c is unoccupied.

$Z^r = ({z_1}^r, {z_2}^r, \ldots, {z_m}^r)$ – the sequence of semaphores corresponding to the r-th state, ${z_c}^r \in P$ – means the name of the stream (specified in the c-th dispatching rule σ_c, allocated to the c-th resource) allowed to occupy the c-th resource; for instance ${z_c}^r = P_i^k$ means that at the moment stream P_i^k is allowed to occupy the c-th resource.

$Q^r = ({q_1}^r, {q_2}^r, \ldots, {q_m}^r)$ – the sequence of semaphore indices, corresponding to the r-th state, ${q_c}^r$ determines the position of the semaphore ${z_c}^r$ in the priority dispatching rule σ_c, ${z_c}^r = crd_{({q_c}^r)}\sigma_c$, ${q_i}^r \in \mathbb{N}$. For instance ${q_2}^r = 2$ and ${z_2}^r = P_1^2$, that means the semaphore ${z_2}^r = P_1^2$ takes the 2nd position in the priority dispatching rule σ_2.

The State Sl^k is Feasible [2] when:

- semaphores of occupied resources indicate the streams allocated to them,
- each local stream is allotted to a unique resource due to a relevant process route.

The introduced concept of the i-th state Sl^i enables to create the states space $\mathbb{S}l$ of reachable states (feasible states). For the purpose of illustration let us consider the state space of the SCCP from Fig. 1. The states Sl^i are noted by "○" (like in Fig. 2).

Transitions linking feasible states $Sl^i, Sl^i \in \mathbb{S}l$ while following non-preemption and mutual exclusion constraints are denoted by $Sl^i \rightarrow Sl^j$, and encompass the next state function δ: $Sl^i = \delta(Sl^j)$, definition of which [2] leads to the following property:

Property 1
Each $Sl^i \in \mathbb{S}$ *can have many predecessors seen as the states* $\mathbb{SP}^i$, $\mathbb{SP}^i \subset \mathbb{S}l$ *(also* $\mathbb{SP}^i = \emptyset$*), i.e.* $\forall Sl^k \in \mathbb{SP}^i$, $Sl^i = \delta(S^k)$ *however only one descendent seen as the unique state* $Sl^j \in \mathbb{S}$, *i.e., there exists at most one* $Sl^j \in \mathbb{S}l$, $Sl^j = \delta(Sl^i)$.

Proof follows directly from definition of the next state function δ: $\mathbb{S}l \rightarrow \mathbb{S}l$, i.e., the mapping from $\mathbb{S}l$ into $\mathbb{S}l$ [2]. That means the $Sl^i \in \mathbb{SP}^i$ *can result in a the unique* Sl^j. □

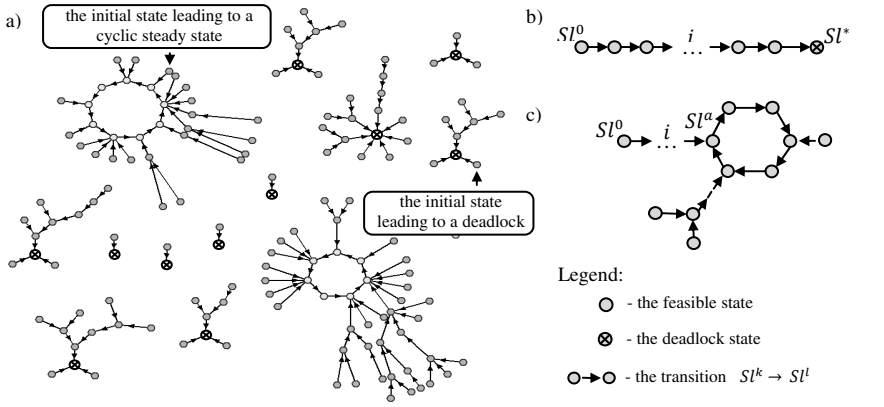


Fig. 3. The space of feasible states encompassing the SCCP's behavior (see SCCP from Fig. 1) a), the string-like digraph ending with a deadlock state b), and the string-like digraph ending with a state belonging to a cyclic steady state c)

In that context states $Sl^* \in \mathbb{S}l$ without the descendants are called the deadlock states. In general case two kinds of steady state behaviors can be considered: a **cyclic steady state** and a **deadlock state**.

The set $Sc^* = \{Sl^{k_1}, Sl^{k_2}, Sl^{k_3}, \ldots, Sl^{k_v}\}$, $Sc^* \subset \mathbb{S}l$ is called **a reachability state space of local processes** generated by the set of initial states $Si \subset Sc^*$ (see Fig. 3c)), if the following condition holds:

$$\forall_{Sl^{k_x} \in Si}\, Sl^{k_x} \rightarrow \ldots \rightarrow Sl^{k_i} \xrightarrow{v-i-1} Sl^{k_v} \xrightarrow{i} Sl^{k_i} \tag{4}$$

where: $Sl^a \xrightarrow{i} Sl^b$ – the next state transition defined in [2], $Sl^{k_1} \xrightarrow{i} Sl^{k_{i+1}} \equiv Sl^{k_1} \rightarrow Sl^{k_2} \rightarrow Sl^{k_3} \rightarrow \ldots \rightarrow Sl^{k_{i+1}}$

The set $Sc = \{Sl^{k_i}, Sl^{k_{i+1}}, \ldots, Sl^{k_v}\}$, $Sc \subseteq Sc^*$ is called **a cyclic steady state of local processes** (i.e., a cyclic steady state of the $SCCP$) with the period $Tc = \|Sc\|$, $Tc > 1$.

$$Sc \cup Si = Sc^* \text{ and } Sc \cap Si = \emptyset \tag{5}$$

In other words a cyclic steady state contains such a set of states in which starting from any initial state it is possible to reach the rest of states and finally reach this distinguished state again:

$$\forall_{Sl^k \in Sc} \left(Sl^k \xrightarrow{Tc-1} Sl^k \right) \tag{6}$$

Moreover, since an initial state $Sl^{k_1} \in \mathbb{S}l$ either lead to Sc or to a deadlock state Sl^*, i.e. $Sl^{k_1} \xrightarrow{i-1} Sl^{k_i} \xrightarrow{v-i-1} Sl^{k_v} \rightarrow Sl^*$, hence local processes can reach a **deadlock state**, (denoted by "⊗" in Fig. 3b) also.

Property 2
Consider two sets of states Sc_1^* and Sc_2^* leading to the two different cyclic steady states Sc_1 and Sc_2, respectively. $Sc_1^* \cap Sc_2^* = \emptyset$ holds.

Proof (by contradiction): Assume there exists $\mathrm{Sl}^{\mathrm{i}} \in \mathrm{Sc}_1^* \cap \mathrm{Sc}_2^*$. That means at least two of its predecessors $\mathrm{Sl}^{\mathrm{k}_1}$, $\mathrm{Sl}^{\mathrm{k}_2}$ there exist (i.e., $\mathrm{Sl}^{\mathrm{k}_1} \rightarrow \ldots \rightarrow \mathrm{Sl}^{\mathrm{k_j}} \xrightarrow{\mathrm{Tc}_1-1} \mathrm{Sl}^{\mathrm{k_j}}$, and $\mathrm{Sl}^{\mathrm{k}_2} \rightarrow \ldots \rightarrow \mathrm{Sl}^{\mathrm{k_i}} \xrightarrow{\mathrm{Tc}_2-1} \mathrm{Sl}^{\mathrm{k_i}}$). Consequently, the contradiction follows from the Property 1. □

Due to presented definitions the reachability problem of local processes cyclic steady states space can be defined following:

Given is the SC specified by (2). Two cyclic steady states $\mathbf{Sc_1}$ and Sc_2 of the SC (encompassing cyclic steady states of local processes) are known. Is the cyclic steady state Sc_2 reachable from the Sc_1?
So, the question we are facing with is: Is it possible to switch directly or indirectly from one cyclic steady state of local processes to an assumed another one? For instance, let us consider cyclic steady states Sc_1 and Sc_2 from Fig. 3a).

Searching for direct switching between Sc_1 and Sc_2 assumes the state $Sl^x \in \mathbb{S}l$ belonging to both cyclic steady states has to exist. That is impossible because *property 2*. What is impossible at the $\mathbb{S}l$ level can be possible, however, at the $\mathbb{A}$ level (i.e., $\mathbb{S}l$ evaluation from "allocations space" perspective), see Fig. 4. At the level $\mathbb{A}$ there are allocations A^i possessing more than one descendent. Such situation corresponds to an allocation belonging to the several states Sl^i. For instance, A^4 belongs to $Sl^4 = (A^4, Z^4, Q^4)$ and $Sl^{17} = (A^4, Z^{17}, Q^{17})$, simultaneously. The states of local processes specified by the common allocation are different because the semaphores and indices are different (Z^4, Q^4 and Z^{17}, Q^{17}).

Such rules of semaphores and indices changes can be treated as relevant control rules. In this case changes do not require any allocations change, and then do not lead to the system stoppage. Consequently, the following properties can be stated:

Property 3
Consider the SC model. The cyclic steady state $Sc_1 \subseteq Sc_1^*$ is reachable from the cyclic steady state $Sc_2 \subseteq Sc_2^*$, (i.e., $Sc_2 \longrightarrow Sc_1$), only if states $Sl^a \in Sc_1^*$, $Sl^b \subseteq Sc_2$ posses the same allocation A^x.

Proof: Consider two states $S^a = (A^x, Z^a, Q^a) \in Sc_1^*$, $S^b = (A^x, Z^b, Q^b) \in Sc_2$ possessing the same allocation A^x. Assume Z^b, Q^b in S^b (from Sc_2) have been converted into Z^a, Q^a belonging to S^a (from Sc_1). Due to (5) such change results in switching from Sc_2 to Sc_1. □

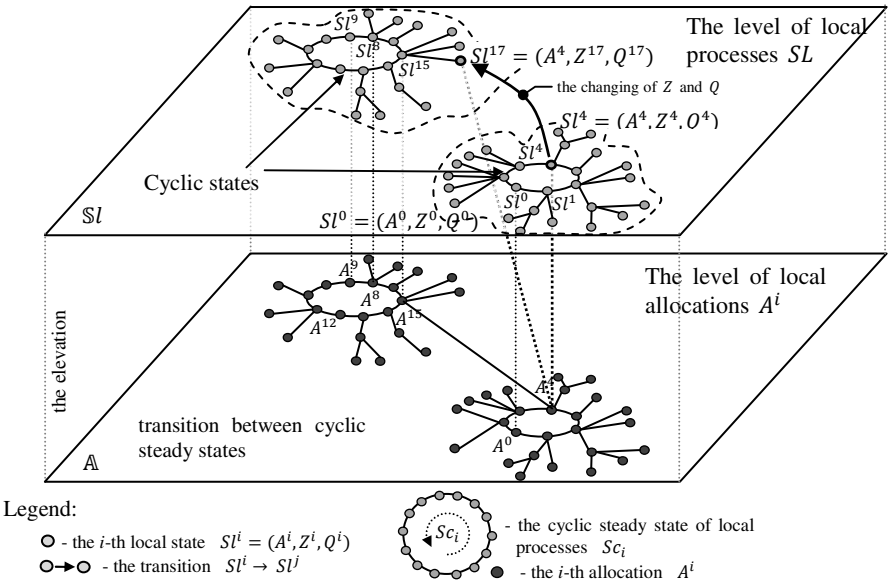


Fig. 4. Projection of $\mathbb{S}l$ onto $\mathbb{A}$ (see the behaviors from Fig. 2)

Property 4
Two cyclic steady states $Sc_1 \subseteq Sc_1^*$ and $Sc_2 \subseteq Sc_2^*$ from the SC are mutually reachable, (i.e., $Sc_2 \leftrightarrow Sc_1$) if and only if, $Sc_1 \longrightarrow Sc_2$ and $Sc_2 \longrightarrow Sc_1$ hold.

Proof: Assume $Sc_1 \longrightarrow Sc_2$ and $Sc_2 \longrightarrow Sc_1$. From the Property 3 it follows that Sc_2 reachable from Sc_1 and Sc_1 is reachable from Sc_2.That means both states Sc_1 and Sc_2 are mutually reachable from each other. □

So, the reachability problem of the cyclic steady states space, e.g. regarding of switching between two states $Sc_1 \subseteq Sc_1^*$ and $Sc_2 \subseteq Sc_2^*$, concludes in the question: Does there exist two states $Sl^a \in Sc_1$ and $Sl^b \subseteq Sc_2^*$ ($Sl^a \in Sc_1^*$ and $Sl^b \subseteq Sc_2$) sharing the same allocation A^x of local cyclic processes?

4 Illustrative Example

Given the SCCP see Fig. 1. The available cyclic steady states spaces Sc_1 and Sc_2 are shown in Fig 2. Consider illustration of the property 3, where switching between Sc_1 and Sc_2 occurs on $Sl^4 = (A^4, Z^4, Q^4)$ from Sc_1 (see Fig. 4) possessing the same allocation $A^4 = (P_3^1, P_1^1, P_5^1, P_6^1, \Delta, P_2^1, \Delta, P_4^1, \Delta, \Delta, \Delta, \Delta)$ as Sl^7 fromSc_2. At this state

the semaphore $Z^4 = (P_3^1, P_1^1, P_5^1, P_6^1, P_3^1, P_2^1, \boxed{P_2^1}, P_4^1, P_4^1, P_5^1, P_5^1, P_6^1)$ associated to R_7 has been changed by $Z^{17} = (P_3^1, P_1^1, P_5^1, P_6^1, P_3^1, P_2^1, \boxed{P_5^1}, P_4^1, P_4^1, P_5^1, P_5^1, P_6^1)$ as well as the index $Q^4 = (2,2,1,1,2,2,\boxed{1},1,2,2,2,1)$ has been converted by $Q^4 = (2,2,1,1,2,2,\boxed{2},1,2,2,2,1)$. That indirect (i.e., in cost of introduction of one extra state) rescheduling between two cyclic steady states is shown Fig. 5.

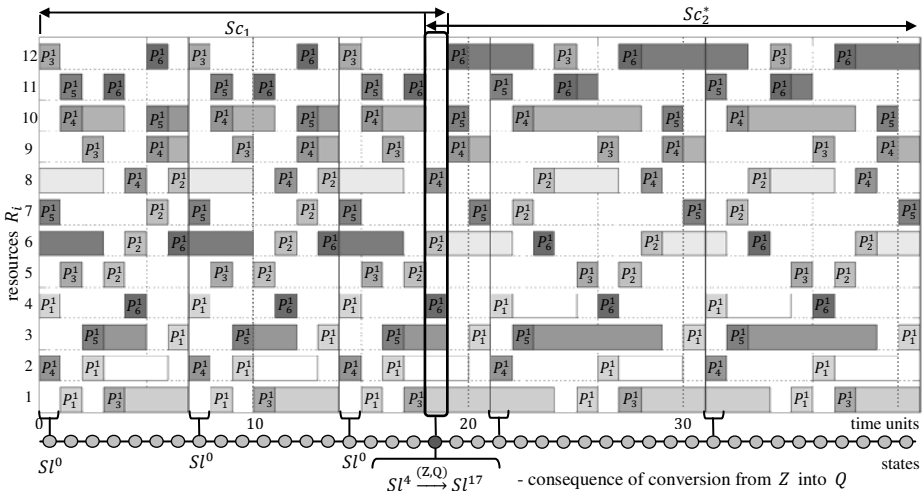


Fig. 5. Gantt's chart illustrating the way the cyclic steady state Sc_2 can be reachable from Sc_1

5 Concluding Remarks

The approach proposed is based on concurrently flowing cyclic processes concept assuming their cyclic steady state behavior guaranteed by a given set of dispatching rules and assumed set of initial processes allocations. The objective is to provide the rules useful in the course of scheduling and rescheduling of SCCPs. In that context the sufficient conditions enabling direct and indirect rescheduling of SCCP cyclic steady state behaviors are provided.

References

1. Alpan, G., Jafari, M.A.: Dynamic analysis of timed Petri nets: a case of two processes and a shared resource. IEEE Trans. on Robotics and Automation 13(3), 338–346 (1997)
2. Bocewicz, G., Banaszak, Z.: Declarative approach to cyclic scheduling of multimodal processes. In: Golińska, P. (ed.) EcoProduction and Logistics, vol. 1. Springer (2012)
3. Fournier, O., Lopez, P., Lan Sun Luk, J.-D.: Cyclic scheduling following the social behavior of ant colonies. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 450–454 (2002)

4. Levner, E., Kats, V., de Pablo, D.A.L., Cheng, T.C.E.: Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers and Industrial Engineering 59(2), 352–361 (2010)
5. Liebchen, C., Möhring, R.H.: A case study in periodic timetabling. Electronic Notes in Theoretical Computer Science 66(6), 21–34 (2002)
6. Polak, M., Majdzik, P., Banaszak, Z., Wójcik, R.: The performance evaluation tool for automated prototyping of concurrent cyclic processes. Fundamenta Informaticae 60(1-4), 269–289 (2004); ISO Press, Editor-in-Chief, Skowron, A.
7. Wójcik, R.: Constraint programming approach to designing conflict-free schedules for repetitive manufacturing processes. In: Cunha, P.F., Maropoulos, P.G. (eds.) Digital Enterprise Technology. Perspectives and Future Challenges, pp. 267–274. Springer (2007)

Comparison of Allocation Algorithms in Mesh Oriented Structures for Different Scheduling Techniques

Bartosz Bodzon, Leszek Koszalka, Iwona Pozniak-Koszalka, and Andrzej Kasprzak

Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
bodzon@yahoo.com, {leszek.koszalka, iwona.pozniak-koszalka,andrzej.kasprzak}@pwr.wroc.pl

Abstract. The paper concerns task allocation problem in mesh structured system. The dynamic case is considered. Four allocation algorithms have been evaluated. The research was focused on the impact of task scheduling technique co-operated with allocation algorithms. Two queuing schemes were compared: well-known First Come First Served and newly created, by the authors of this paper, heuristic scheduling technique called First Few Random. The comparison of efficiencies of different allocation algorithms combined with different queuing schemes has been done on the basis of simulation experiments made with a designed experimentation system. The discussion of the obtained results confirms that the proposed approach and created queuing scheme seem to be promising.

Keywords: Mesh structure, task allocation, algorithm, scheduling, experimentation system.

1 Introduction

In order to solve today's complex problem such as weather forecasting or molecular modelling high computational power is required. One of the most powerful supercomputers called K computer consists of 548,352 cores (in a 6D mesh/torus interconnects), and it totally consumes 9.89 MW [1]. An efficient resource management is necessary to decrease maintenance cost, especially for services which are based on computational power renting for a private sector. If we simplify available resources to CPUs, then task allocation algorithm can be considered as a resource manager, since each task requires a fixed number of processing units.

The mesh network structures due to its simplicity, modularity, scalability, structural regularity are relatively common in multicomputer systems [2-5]. For instance, IBM's Blue Gene/L, Blue Gene/P, Cray's XT3 and XT4 have a 3D torus interconnect [6]. In such representation each processing unit can be described as a node. If the nodes within the task are required to be adjacent then the contiguous allocation algorithms are desired such as: 2D Buddy, Frame Sliding or First Fit. The major disadvantage of contiguous allocation algorithms is a fragmentation, i.e., unused nodes. Fragmentation can be divided into internal and external [7]. Internal fragmentation

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 223–232, 2012.

occurs when the allocation algorithm requires more nodes than task requires. External fragmentation occurs when the mesh has sufficient number of available nodes, but they are not physically adjacent. In contrary, non-contiguous allocation algorithms such as random allocation suffers from possible lack of communication, although they are free from fragmentation. Contiguous allocation algorithms can be further divided to structure preserving and non-preserving. Non-preserving methods modify the tasks' shape in order to increase their allocation possibility. One of the drawbacks of the shape modification is the increase of the execution time – due to prolongation of a communication time (see [8], and [9]).

The allocation procedure can be mainly divided into static and dynamic [2]. In static mode, in contrary to dynamic mode, execution time of each task is treated as infinite. Additionally, in static mode we know the exact number of tasks and their sizes, so additional pre-processing can be done – such as queue sorting.

To measure algorithms performance, the indices such as execution time, turnaround time and fragmentation are used. Execution time is a time from algorithm start to de-allocation of the last task in mesh. Turnaround time is time interval between task addition to the queue and its de-allocation. Fragmentation is a ratio of free processing units to all units.

This paper focuses on dynamic contiguous allocation algorithms in 2D rectangular mesh structures.

The rest of the paper is organized as follows: In Section 2, a very brief introduction to mesh structures is made. Section 3 concentrates on review of different types of contiguous allocation algorithms. The heuristic scheduling technique applied to allocation process, created by the authors of this paper, is presented in Section 4. It is followed by description of experimentation system and presentation of some results of simulation experiments in Section 5. Section 6 contains discussion of the obtained results. Finally, conclusion with perspectives for further research in the considered area appears in Section 7.

2 Mesh Oriented Structure

An example of the two dimensional rectangular mesh M(W,H) with width W and height H, which consists W x H nodes, is shown in Fig. 1. Each node Ni,j (besides boundary nodes) is connected to Ni-1,j, Ni,j-1, Ni,j+1, Ni+1,j nodes. A base node BN is considered as upper-left node (i.e., 0,0). Allocated tasks cannot overlap and are represented by following tuples:

- A rectangular busy sub-mesh $S_i(a_i,b_i)$ and position $P_i(x,y)$.
- An L-shape rectangular busy sub-mesh $LS_i(c_i,d_i,e_i,f_i)$ and position $P_i(x,y)$.

L-shape sub mesh $LS_i(c_i,d_i,e_i,f_i)$ can be described as pair of rectangular busy sub-meshes:

$$S_{ia}(c_i,d_i) \quad \text{and} \quad S_{ib}(e_i,f_i), \text{ where } \quad a_i \, x \, b_i = c_i \, x \, d_i + e_i \, x \, f_i.$$

As mentioned before, each task after allocation is executed for a fixed period of time, then it gets to be de-allocated and the nodes can be used once again. The 3D rectangular mesh is presented in Fig. 2 and 2D torus in Fig. 3.

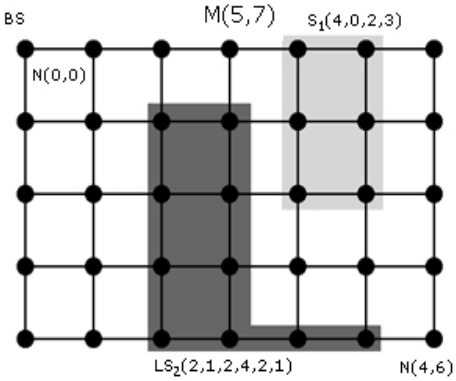


Fig. 1. Mesh M(5,7) with two allocated tasks

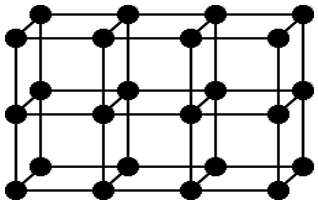


Fig. 2. Example of 3D rectangular mesh

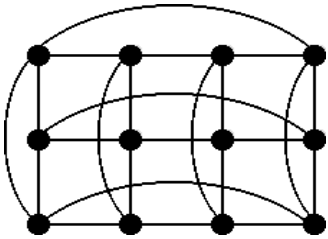


Fig. 3. Example of 2D torus mesh

3 Allocation Algorithms

3.1 2D Buddy

A two dimensional buddy scheme introduced in [10] is one of the simplest methods of allocation; however it suffers from both the internal and the external fragmentation [8]. The idea of the algorithm is as follows. Firstly, an initial block (sub-mesh) is created S(A,A), A = min(W,H) where M(W,H). If the incoming task is smaller than square sub-mesh of a side A/2, then four buddies are created Sa,b,c,d(A/2,A/2), Pa(0,0), Pb(0,A/2), Pc(A/2,0), Pd(A/2,A/2). The procedure repeats until task size is smaller than buddy sub-mesh. The task is allocated in first smallest free sub-mesh.

3.2 First Fit

Although a first fit (FF) algorithm, introduced in [11], is more complex and slower than the 2DB, it is free from internal fragmentation. Algorithm maintains a busy array which is a simple bitmap representation of a mesh where busy nodes are denoted as 1 and free as 0. The algorithm is scanning the busy array from left to right and from top to bottom looking for task size array which contains all zeroes. If such an array is found algorithms stops.

3.3 Stack Based Algorithm. Best Fit Stack Based Algorithm

A stack based algorithms (SBA) uses a candidate blocks CB as possible blocks in which task can be allocated. The algorithm works as follows: Initially, the CB is whole mesh and it is put on the stack. Also, all busy sub-meshes BSs are put on another stack.

Then one BS is taken from the stack. Each CB is checked whether it has common nodes with current BS. If such condition is satisfied, up to four new CBs are created by subtraction from the original CB a BS. Subsequently, CBs are put on stack. The procedure continues until there are no more BSs on the stack. Best fit stack based algorithm (BFSBA) [12] is SBA modification which chooses the CB with minimal height and minimal horizontal position [13].

3.4 FlexFold

A FlexFold [8] is shape transformation, which tries to allocate (using e.g., FF) transformed task. Search order of feasible transformations task *S(a,b)* is:

- *S(a,b), S(b,a)*
- if *a* is even then *S(a/2, 2b), S(2b, a/2)*
- if *b* is even then *S(2a, b/2), S(b/2, 2a)*

To analyse impact of task's shape transformation to its execution time let consider task S as a graph G(V,E) and task after embedding ε as graph a G'(V',E'). A dilation of embedding is maximal distance between two nodes in the G' (the nodes which are adjacent to the G). A congestion is maximum number of edges in the G whose equivalent paths share a common edge in the G' [8]. The S(a,b) → S(b,a) dilation and congestion equals to 1. In contrary, widening and narrowing has at most dilation and congestion equalled to 2. Shape manipulation affects only a communication time which is a fraction f of task's execution time TET. Finally, folded task TET can increase at most 4 x f x t – t is TET of the unfolded task [8]. An example of shape transformation is presented in Fig. 4.

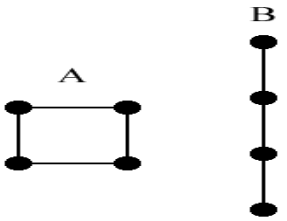


Fig. 4. Example of task transformation with FlexFold (before – A, after - B)

Allocation depends on the conservative check procedure CC which compares the increase of the execution time to time that left to the earliest finish task. Therefore, if the first one is smaller, then allocation is made. Communication time can be divided into network propagation and end-node time. Propagation is simply the time the message spends in a network including routers delays; whereas end-node time denotes time spend in source and destination nodes [8]. In most cases end-node time is

significantly larger than propagation time (e.g., Intel's Delta message transfers – 67μs, less than 1μs time spent in network) [14]. Making the algorithm independent from architecture no division will be made for propagation and end-node time, although obtained results will represent the upper boundary.

3.5 L-Shaping Algorithm

An L-shaping algorithm (LSA) [9] is the shape transformation algorithm based on the FlexFold introducing additional feasible embedding. The LSA uses cutting side *ct* of *S(a,b)* which is a maximal value of *a* or *b* if their product is even and minimal value otherwise [9]. Transformations in order of search are as follows [9]:

- *S(a,b), S(b,a)*
- if *a* is even then *S(a/2, 2b), S(2b, a/2)*
- if *b* is even then *S(2a, b/2), S(b/2, 2a)*
- *L-shaping(ct)*

An example of such transformation is depicted in Fig. 5. The dilation of task *S(a,b)* is equal to (*a/2+1*) for the even cutting side *a,* and is equal to (*3+2k)* for the odd cutting side. Both embedding has the congestion of 2. Therefore the maximal communication time increase is *2a* [9]. To justify the transformation algorithm also incorporates the CC.

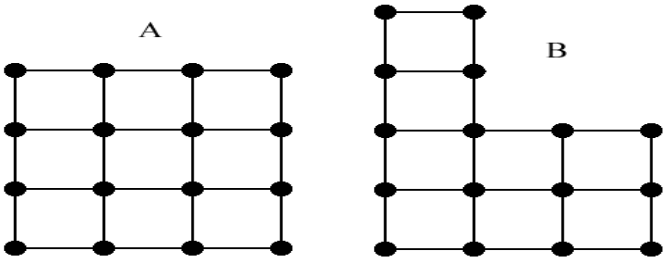


Fig. 5. One of feasible task transformation with LSA (before - A, after - B)

4 Queuing Schemes

4.1 First in First Out

A classic first in first out (FIFO) queue scheme (called also as a first come first served) is a simple scheme which returns task in the same order of putting them; hence it is considered as a fair strategy. The order of obtaining items from the queue is fully deterministic.

4.2 First Few Random

A first few random (FFR) is a queuing scheme created by the authors of this paper. It is based on observations that smaller tasks could be allocated instead of being kept in

the queue when first task in the queue cannot be allocated. The FFR divides queue into two parts.

The first part has predefined fixed size n. additionally each task in n-sized queue has assigned probability of being chosen p_i, which is inversely proportional to its size. Using a uniformly distributed random generator j task is chosen. If j-*th* is not the first task, the probability of p_1 is increased by ξ-times to prevent starvation. At any point in time a constraint (1) must be hold, i.e., increase of probability of p_1 effectively leads to decrease of other p.

$$\sum_{i}^{n} p_i = 1 \tag{1}$$

The second part is a simple FIFO queue. A scheduling example with one step (task $j = 3$ is chosen) is shown in Table 1.

The tuning ξ - parameter allows to decrease the time for which the currently first task stays in the queue. The greater value of this parameter is then the FFR becomes more similar to FIFO, which may be important for systems demanding more deterministic behavior.

Although the n value increases the performance of the system, it is not always a case when one knows what the next n tasks are going to be. Moreover, an increment of n value may lead to longer queue occupation by the first task.

Table 1. Example of the FFR for $n = 5$, $\xi = 20\%$. Right part represents queue after $j = 3$ task being chosen.

Task	Size *a* x *b*	1/Size	p_i
…	…	0.000	0.000
7	4	0.000	0.000
6	2	0.000	0.000
5	1	1.000	0.410
4	4	0.250	0,118
3	**2**	**0.500**	**0.235**
2	4	0.250	0.118
1	8	0.125	0.059
	Sum	**2.125**	**1.000**

Task	Size *a* x *b*	1/Size	p_i
…	…	0.000	0.000
…	…	0.000	0.000
6	4	0.000	0.000
5	2	0.500	0.233
4	1	1.000	0.465
3	4	0.250	0.116
2	4	0.250	0.116
1	8	0.150	0.070
	Sum	**2.150**	**1.000**

5 Experimentation System

All studies were performed using a Monte-Carlo algorithm with 50 repetitions on the 100x100 mesh grid. Probability of incoming task at any iteration was fixed to the 80%. In any iteration there were simulated 1000 incoming tasks. A single task size was generated using normally distributed random number generator with additional

constraint that task size cannot exceed the mesh size. All experiments were performed on the PC class computer with Intel® Core™ i5-2520 processor and 4GB of RAM using program written in the Java™. The four algorithms presented in Section 3 were implemented by the authors of this paper.

The two different studies were made. A first one (Experiment 1) compares the system load against the average turnaround time, whereas a second one (Experiment 2) focus on correlation between tasks' size standard deviation against the average turnaround time. The turnaround time was used as performance indicator, since for end user perspective it is the most important one [8].

Experiment 1. The objective was to compare the system load against the average turnaround time corresponded to the considered allocation algorithms in two cases: with FIFO scheme and with FFR scheme. The system load (2) describes the average occupation of processor [15] and is defined as follows:

$$\text{System load} = \frac{\text{Avg.residency } x \text{ Avg.request size}}{\text{Architecture size } x \text{ Avg. inter} - \text{arival time}} \tag{2}$$

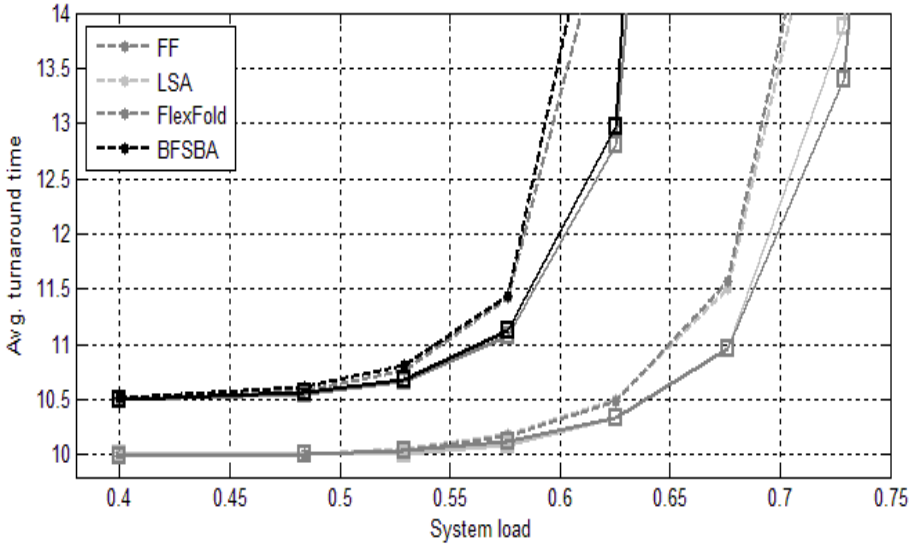


Fig. 6. Average turnaround time as function of queue type and system load

For presented simulation study: architecture size, avg. residency and avg. inter-arrival time [8] were fixed. The residency size was generated with normal distribution N(10,4). The FF, the LSA, the FlexFold and the BFSBA allocation algorithms were taken into consideration. The communication fraction was set to 10%, and parameters of the FFR were set accordingly to n=10 and $\xi = 20\%$.

The results are presented in Fig. 6, where the dash lines describe allocation algorithms with FIFO queue and the continuous lines with the proposed FFR queue.

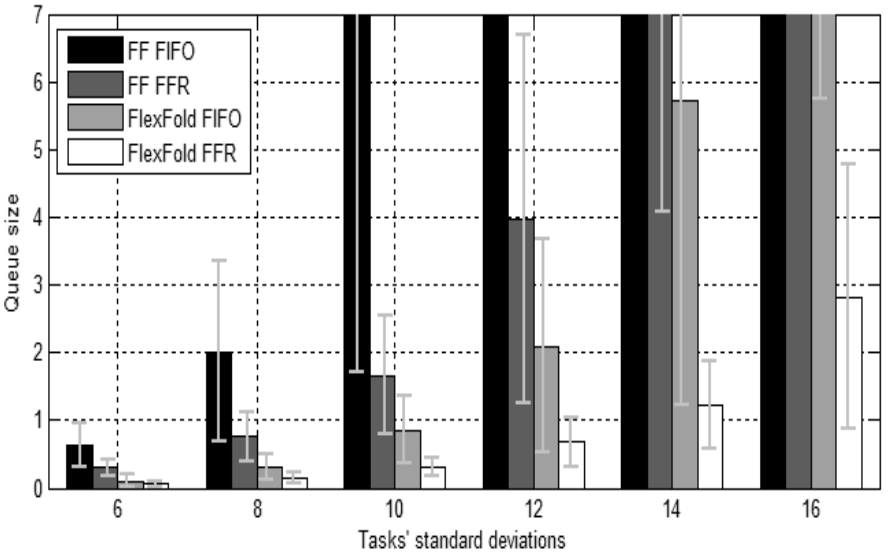


Fig. 7. Average queue size as function of queue type and tasks' size standard deviation

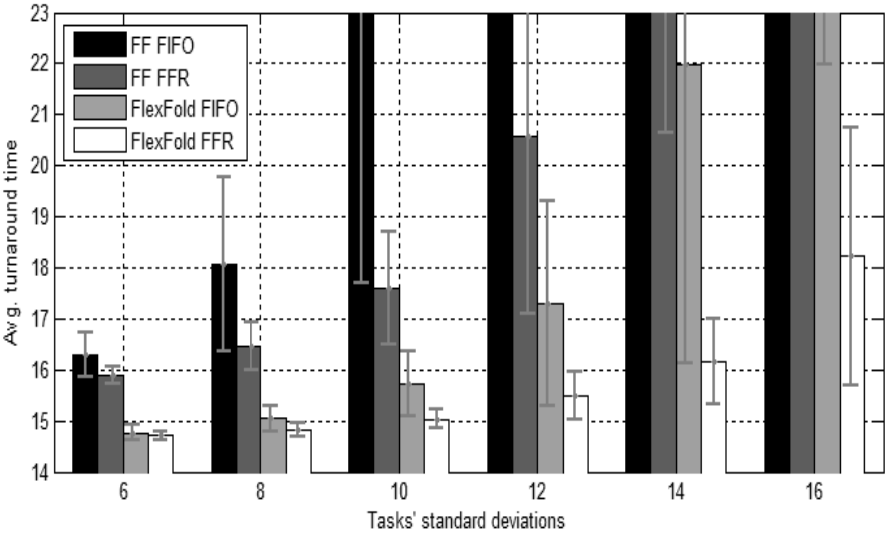


Fig. 8. Average turnaround in relation to queue type and task's size standard deviation

Experiment 2. The purpose of this experiment was to find how task size variation affects the proposed queuing scheme. The justification for this experiment was that one might be interested whether it is beneficial to run dissimilar task in terms of size on same machine, or in contrary it is better to stack them in similar sizes and run those groups alternatively. The task size and duration time was set to be generated with normal distribution, whereas mean value for size was 20 and N(15, 4) for the

duration time. The simulation studies were made for both shape preserving (FF) and non-preserving allocation (FlexFold) algorithm using FIFO and FFR queue. The obtained results are shown in Fig. 7 and Fig. 8.

6 Discussion of Results of Experiments

The results of Experiment 1 show that indeed, the shape manipulation technique allows increasing system performance. Moreover, the FFR queue scheme seems to decrease average turnaround time for same system load.

As long the system load is low, there is no noticeable difference between the FIFO and the FFR, since most of tasks are instantly allocated. Moreover, the percentage of improvement also decreases when system approaches unstable state – the queue length does not converge. The average improvement which can be achieved for the stable state is approximately 5%. Although, it may seem that the improvement is rather insignificant, it can be obtained almost without any additional computational cost, since its complexity is constant (in terms of mesh size).

Furthermore, because it is queuing scheme it can be easily applied to almost any mesh architecture without any modifications, which is no longer true for allocation algorithms, where additionally algorithm time complexity increases with mesh dimension. Moreover, the improvement seems to be independent from used algorithm, hence queuing scheme can probably successfully be applied to any allocation algorithm.

The simulation shows that L-shaping Algorithm performed comparable to the FlexFold. Diversity between obtained results and those in [9] are probably caused by the different incoming task generation scheme.

The results of Experiment 2 show that joining task in similar size groups might be advantageous, despite the fact that the higher tasks' size variation (for stable system), results in higher FFR performance. For that system, in which for various reasons it is impossible to group tasks, the FFR performance would be approximately up to 30% higher.

Using the FFR scheme can ensure smaller standard deviation for both mean queue size and mean turnaround time; hence system's behavior becomes more independent from the task size distribution, which might be unknown.

7 Conclusion

The created FFR queuing scheme is based on a simple idea, nevertheless, it allows to further increase of utilization regardless of network structure or used allocation algorithm, without additional computational overhead, making proposed strategy very promising.

The possible area for further studies is in determining the performance of the FFR into the multidimensional mesh structures. Moreover, the presented FFR assumed that all task has initially same priority, which might not always be a case, hence suitable modification would be needed to handle tasks with priorities or systems with the preemption.

Recently, the experimentation system (with the four implemented allocation algorithms) is serving as a tool to aid teaching graduate students and preparing projects in computer science and telecommunications areas in Faculty of Electronics, Wroclaw University of Technology.

References

1. The New York Times, http://www.nytimes.com/ (accessed June 20, 2011)
2. Koszalka, L.: Static and Dynamic Allocation Algorithms in Mesh Structured Networks. In: Madria, S.K., Claypool, K.T., Kannan, R., Uppuluri, P., Gore, M.M. (eds.) ICDCIT 2006. LNCS, vol. 4317, pp. 89–101. Springer, Heidelberg (2006)
3. Byung, S., Das, C.R.: A Fast and Efficient Processor Allocation Scheme for Mesh-Connected Multicomputers. IEEE Trans. on Computers 1, 46–59 (2002)
4. Sharma, D.D., Pradhan, K.: Submesh Allocation in Mesh Multicomputers Using Busy-List: A Best Fit Approach with Complete Recognition Capability. Journal of Parallel and Distributed Computing 36(2), 106–118 (1996)
5. Bani-Mohammad, S., Ababneh, I., Hamdan, M.: Comparative Performance Evaluation of Non-Contiguous Allocation Algorithms in 2D Mesh-Connected Multicomputers. In: CIT, pp. 2933–2939 (2010)
6. Parallel Machines and Topologies, https://charm.cs.uiuc.edu/ (accessed June 29, 2011)
7. Gabrani, G., Mulkar, T.: A Quad-Tree Based Algorithm for Processor Allocation in 2D Mesh-Connected Multicomputers. Computer Standards & Interfaces 27(2), 133–147 (2005)
8. Gupta, V., Jayendran, A.: A Flexible Processor Allocation Strategy for Mesh Connected Parallel Systems. In: Proceedings to Parallel Processing Conference, pp. 166–193 (1996)
9. Seo, K.-H.: Fragmentation-Efficient Node Allocation Algorithm in 2D Mesh-Connected Systems. In: Proceedings of the 8th International Symposium on Parallel Architecture, Algorithms and Networks, ISPAN 2005, pp. 318–323. IEEE Computer Society Press, Washington, DC (2005)
10. Li, K., Cheng, K.H.: A Two-Dimensional Buddy System for Dynamic Resource Allocation in a Partitionable Mesh Connected System. Journal of Parallel and Distributed Computing 12(1), 79–83 (1991)
11. Zhu, Y.: Efficient processor allocation strategies for mesh-connected parallel computers. Journal of Parallel and Distributed Computing 16(4), 328–337 (1992)
12. Koszalka, L., Lisowski, D., Pozniak-Koszalka, I.: Comparison of Allocation Algorithms for Mesh Structured Networks with Using Multistage Simulation. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Kenneth Tan, C.J., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 58–67. Springer, Heidelberg (2006)
13. Kmiecik, W., Wojcikowski, M., Koszalka, L., Kasprzak, A.: Task Allocation in Mesh Connected Processors with Local Search Meta-heuristic Algorithms. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS 2010. LNCS (LNAI), vol. 5991, pp. 215–224. Springer, Heidelberg (2010)
14. Barnett, M., Littlefield, R.J., Payne, D.G., van de Geijn, R.A.: Global Combine Algorithms for 2-D Meshes with Wormhole Routing. J. Parallel Distributed Computing 24(2), 191–201 (1995)
15. Seo, K.-H., Kim, S.-H.: Improving System Performance in Contiguous Processor Allocation for Mesh Connected Parallel Systems. Journal of Systems and Software 67(1), 45–54 (2003)
16. Zydek, D., Selvaraj, H., Koszalka, L., Pozniak-Koszalka, I.: Evaluation scheme for NoC-based CMP with integrated processor management system. International Journal of Electronics and Telecommunications 56(2), 157–167 (2010)

Reachability of Cyclic Steady States Space: Declarative Modeling Approach

Grzegorz Bocewicz[1], Robert Wojcik[2], and Zbigniew A. Banaszak[3]

[1] Dept. of Electronics and Computer Science,
Koszalin University of Technology, Koszalin, Poland
bocewicz@ie.tu.koszalin.pl
[2] Institute of Computer Engineering, Control and Robotics,
Wrocław University of Technology, Wrocław, Poland
robert.wojcik@pwr.wroc.pl
[3] Dept. of Business Informatics, Warsaw University of Technology, Warsaw, Poland
Z.Banaszak@wz.pw.edu.pl

Abstract. The paper presents a new modeling framework enabling to evaluate the cyclic steady state of a given system of concurrently flowing cyclic processes (SCCP) sharing common system resources while interacting on the base of a mutual exclusion protocol. Assuming a given topology of cyclic routes passing on by subsets of system resources, a set of dispatching rules aimed at recourses' conflicts resolution, operation times as well as the given frequencies of mutual appearance of local processes the main objective is to provide the declarative modeling framework enabling to refine conditions guaranteeing the cyclic steady state space reachability.

Keywords: cyclic processes, declarative modeling, constraints programming, state space, dispatching rules.

1 Introduction

Operations in cyclic processes are executed along sequences that repeat an indefinite number of times. In everyday practice they arise in different application domains (such as manufacturing, time-sharing of processors in embedded systems, digital signal processing, and in compilers for scheduling loop operations for parallel or pipelined architectures) as well as service domains (covering such areas as workforce scheduling (e.g., shift scheduling, crew scheduling), timetabling (e.g., train timetabling, aircraft routing and scheduling), and reservations (e.g., reservations with or without slack, assigning classes to rooms) [5], [8], [10], [11], [12]. Such cyclic scheduling problems belong to decision ones, i.e. aimed at searching for answering whether a solution possessing the assumed features exists or not [12]. Moreover because of their integer domains the problems considered belong to a class of Diophantine problems as well; that means that some classes of cyclic scheduling problems can be seen as non-decidable (undecidable) ones [1].

Therefore, taking into account non decidability of Diophantine problems one can easily realize that not all the behaviors (including cyclic ones, i.e. encompassing cyclic steady states space) are reachable under constraints imposed by system's structure.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 233–243, 2012.

The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural constraints. That means, since system constraints determine its behavior, hence both system structure configuration and desired cyclic schedule have to be considered simultaneously. So, the problem solution requires that the system structure configuration must be determined for the purpose of processes scheduling, yet scheduling must be done to devise the system configuration. In that context, our contribution provides discussion of some solubility issues concerning cyclic processes dispatching problems, especially the conditions guaranteeing solvability of the cyclic processes scheduling. Their examination may replace exhaustive searching for solution satisfying required system functioning.

Many models and methods have been proposed to solve the cyclic scheduling problem [6]. Among them, the mathematical programming approach [12], max-plus algebra [7], constraint logic programming [1], [4] evolutionary algorithms and Petri nets [8] frameworks belong to the more frequently used. Most of them are oriented at finding of a minimal cycle or maximal throughput while assuming deadlock-free processes flow. The approaches trying to estimate the cycle time from cyclic processes structure and the synchronization mechanism employed (i.e. mutual exclusion instances) are quite unique. In that context our main contribution is to propose a new modeling framework enabling to evaluate the cyclic steady state of a given system of concurrent cyclic processes (SCCP). The following questions are of main interest: Does the assumed system behavior can be achieved under the given system's structure constraints? Does there exist the system's structure such that an assumed system behavior can be achieved?

So, the paper's objective is to provide the observations useful in the course of cyclic steady states generation in a system composed of concurrently flowing cyclic processes interacting between oneself through mutual exclusion protocol. This objective regards of quite large class of digital and/or logistics networks that share common properties even though they have huge intrinsic differences. The most important property concerns of different sub networks infrastructure enabling to schedule multimodal processes executed through connected parts of different local networks [2]. The passenger's itinerary including different metro lines encompass a plan of multimodal process execution within a considered metro network.

This study aims to present a declarative approach to reachability problem that can be used further to assist decision-makers in generation, analyzing and evaluating of cyclic steady states reachable in a given SCCP structure.

The rest of the paper is organized as follows: Section 2 introduces to the systems of concurrently flowing cyclic processes and problem formulation. The cases of generation of cyclic steady states space are introduced in Section 3. Conclusions are presented in Section 4.

2 Systems of Concurrent Cyclic Processes

Consider the digraph shown in Fig. 1. The distinguished are three cycles specifying routes of cyclic processes P_1, P_2, P_2, P_3, P_4 and P_6 respectively. Each process route specified by sequence of resources passed on among its execution can interact with other processes through so-called system common resources. Their routes are specified as follows:

$p_1 = (R_5, R_6, R_1, R_2, R_3, R_4)$, $p_2 = (R_3, R_4, R_5, R_6, R_1, R_2)$, $p_3 = (R_{10}, R_7, R_1, R_8, R_{19}, R_9)$, $p_4 = (R_{12}, R_{13}, R_{15}, R_{14}, R_5, R_{11})$, $p_5 = p_6 = (R_{17}, R_{18}, R_{19}, R_{20}, R_{15}, R_{16})$,

where the resources R_1-R_6, R_{15}-R_{20}, are shared resources, since each one is used by at least two processes, and the resources R_7-R_{14}, are non-shared because each one is exclusively used by only one process. Note that streams of processes, e.g. p_1, p_2, following the same route (workpieces flow in a production line) are modeled as well.

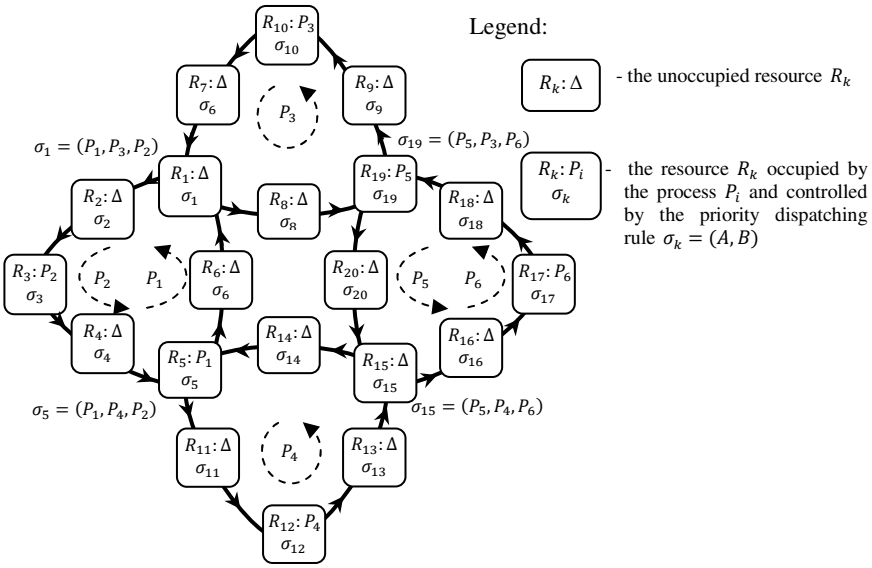


Fig. 1. Process routes structure of SCCP owning three processes

Processes sharing common resources interact each other on the base of mutual exclusion protocol. The possible resource conflicts are resolved with help of assumed priority dispatching rules determining the order in which processes make their access to common shared resources (for instance, in case of the resource R_1, $\sigma_1 = (P_1, P_3, P_2)$ – the priority dispatching rule determines the order in which processes can access to the shared resource R_1, i.e. at first to the process P_1, then to the process P_3, next to P_2 and once again to P_1, and so on). The process P_i occurs the same number of times in each dispatching rule associated to resources appearing in its route. So, the SCCP shown in Fig. 1 is specified by the following set of dispatching rules $\Theta = \{\sigma_1, \dots, \sigma_6, \sigma_{15}, \dots, \sigma_{20}\}$, and $f_1(P_3) = f_{19}(P_3)$, $f_{15}(P_6) = f_{19}(P_6)$,..., $f_5(P_1) = f_1(P_1)$, where $f_c(P_i)$ – a number the i-th process occurs in the c-th priority dispatching rule.

Besides of resource conflicts resolution the priority rules determine the given frequencies of mutual appearance of local processes sharing the resource at hand. For instance, in case of $\sigma_1 = (P_1, P_3, P_2, P_1, P_2, P_1)$ it means that for each two executions of P_2 fall three executions of P_1. In general case, the set of dispatching rules Θ

implies the sequence of relative frequencies of local processes mutual executions denoted by $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$, where: $\psi_i \in \mathbb{N}$.

$$\psi_i = |\{k \mid crd_k\sigma_c = P_i;\ k \in \{1, \ldots, lp(c)\}\}|,\ \forall i \in \{1, \ldots, n\},\ \forall \sigma_c \in \Theta_i, \tag{1}$$

where: Θ_i – the set of dispatching rules associated to resources occurring in the route followed by P_i,

$crd_k\sigma_c$ – the k-th entry of the sequence σ_c, $lp(c)$ – the length σ_c.

So, the SCCP shown in Fig. 1 is specified by the following sequence: $\Psi = (1,1,1,1,1,1)$. That means one execution of each local process falls on one execution of other one.

Since the sequence of relative frequencies of local processes mutual executions Ψ does not necessary encompass cyclic steady state of the SCCP considered, hence a new parameter describing the number of Ψ occurrences within a cyclic steady state is introduced and denoted by $\Xi \in \mathbb{N}$. For example considered $\Xi = 1$, that correspond to the cyclic steady state cycle time of which is equal to 11, see Fig. 4. The introduced variables Ψ and Ξ extend the model proposed in [4]. They allow to considering multiple executions (determined by Ξ) of processes in one cycle.

In general case, each process P_i (where: $P_i \in P = \{P_1, P_2, \ldots, P_i, \ldots, P_n\}$, n - a number of processes) executes periodically while following the route $p_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,lr(i)})$, where: $lr(i)$ - a length of cyclic process route, $p_{i,j} \in R$, $R = \{R_1, R_2, \ldots, R_m\}$, m - a number of resources.

Let us assume that: $o_{i,j}$ - denotes the j-th operation executed by the process P_i^k along the route $p_{i,j}$, and $t_{i,j}$, $t_{i,j} \in \mathbb{N}$, denotes the operation time of the operation $o_{i,j}$ execution. In case of SCCP from Fig. 1 the operation times are shown in the Tab. 1. Therefore, the sequence $T_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,lr(i)})$ describes the operation times required by P_i. Let us assume also, to each shared resource $R_c \in R$ the priority dispatching rule $\sigma_c = (P_{j_1}, P_{j_2}, \ldots, P_{j_{lp(i)}})$, $j_k \in \{1,2, \ldots, n\}$, $lp(c)$ - a number of processes dispatched by σ_c. P_{j_k} executes on the resource R_c. Introduced notations let us specify a descriptive model of a $SCCP$ as the following triple:

$$SC = (ST, BE, SE) \tag{2}$$

where:

$ST = (M, R, T)$ - the variables describing SCCP structure

$M = \{p_1, p_2, \ldots, p_n\}$ – the set of local process routes,

$R = \{R_1, R_2, \ldots, R_m\}$ – the set of resources,

$T = \{T_1, T_2, \ldots, T_n\}$ – the set of local process routes operations times,

$BE = (\Theta, \Psi, \Xi)$ - the variables describing SCCP behaviour

$\Theta = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ – the set of dispatching priority rules, $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$ – the sequence of relative frequencies of local processes mutual executions, Ξ - the number of Ψ occurrences within a one cycle,

$SE = \{eq_{i,j,k}(ST, BE) \mid i = 1, .., n;\ j = 1, \ldots, lr(i); k = 1, \ldots, \Xi\}$ - the set of constraints (equations) linking ST and BE. Each $eq_{i,j,k}(ST, BE)$ describes, the time relation between the moments $x_{i,j,k}$ of operations $o_{i,j}$ beginning for its k-th execution.

Table 1. Operation times of SCCP's (Fig. 1)

P	i	$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$	$t_{i,5}$	$t_{i,6}$
P_1	1	1	1	1	2	1	3
P_2	2	1	3	1	1	1	1
P_3	3	1	1	1	3	1	1
P_4	4	1	3	1	1	1	1
P_5	5	1	3	1	1	1	3
P_6	6	1	2	1	1	1	4

Table 2. Changed operation times of SCCP's (Fig. 1)

P	i	$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$	$t_{i,5}$	$t_{i,6}$
P_1	1	1	1	1	2	1	3
P_2	2	1	3	1	1	1	1
P_3	3	1	1	1	3	1	1
P_4	4	1	3	1	1	1	2
P_5	5	1	3	1	1	1	3
P_6	6	1	2	1	1	1	4

In other words, the constraints $eq_{i,j,k}(ST, BE)$ specify the relation linking X, Tc while encompassing SCCP's cyclic behavior, where: $X = \{X_1, X_2, \dots, X_i, \dots, X_n\}$, $X_i = (x_{i,1,1}, \dots, x_{i,lr(i),1}, \dots, x_{i,1,\Xi}, \dots, x_{i,lr(i),\Xi})$, $x_{i,j,k}$ - the moment of operation $o_{i,j}$ beginning in the l-th cycle: $x_{i,j,k}(l) = x_{i,j,k} + l \cdot Tc$, $l \in \mathbb{Z}$, ($x_{i,j,k}(\mathrm{l}) \in \mathbb{Z}$ – the moment the operation $o_{i,j}^k$ starts its execution in the l-th cycle). Tc denotes the SCCP periodicity: $Tc = x_{i,j,k}(l+1) - x_{i,j,k}(l)$.

For illustration let us consider an example (see Fig. 2) describing the operation $o_{1,3}$ (executed by P_1 on the resource R_1) which can be started (i.e., its first execution; $k = 1$) only if the preceding operation $o_{1,2}$ (executed by P_1 on R_6) has been completed $(x_{1,2,1} + t_{1,1})$ and the resource R_1 has been released, i.e. if the process P_2 occupying the resource R_1 starts its subsequent operation at $x_{2,6,1} - Tc + 1$. So, the relation considered $eq_{1,3,1}(ST, BE)$ can be specified by the following formulae:

$$x_{1,3,1} = max\{(x_{2,6,1} - Tc + 1);\ (x_{1,2,1} + t_{1,2})\} \tag{3}$$

where: $x_{i,j,k}$ – the moment of the operation $o_{i,j}$ beginning in the k-th execution

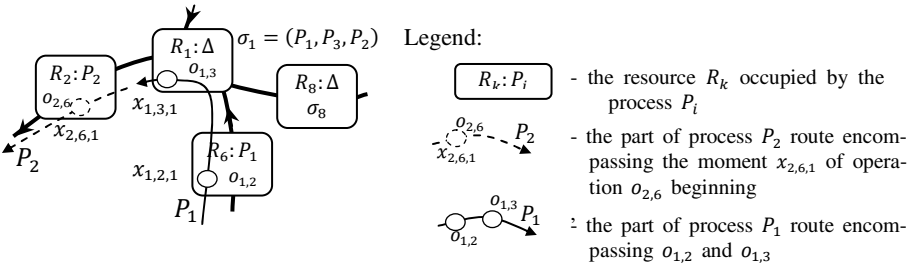


Fig. 2. Illustration of the $x_{1,3,1} = max\{(x_{2,6,1} - Tc + 1);\ (x_{1,2,1} + t_{1,2})\}$ calculation

$T_i = (t_{i,1}, t_{i,2}, \dots, t_{i,lr(i)})$ describes the operation times required by P_i. $\sigma_i = (P_{j_1}, P_{j_2}, \dots, P_{j_{lp(i)}})$, $j_k \in \{1,2,\dots,n\}$, $lp(i)$ - a number of processes dispatched by σ_i the priority dispatching rule assigned to shared resource $R_i \in R$ where P_{j_k} executes on the resource R_i. Therefore a descriptive model of a $SCCP$ as the following triple:

$$SC = (ST, BE, SE) \tag{4}$$

where: $ST = (M, R, T)$ - the variables describing SCCP structure
$M = \{p_1, p_2, \ldots, p_n\}$ – the set of local process routes,
$R = \{R_1, R_2, \ldots, R_m\}$ – the set of resources,
$T = \{T_1, T_2, \ldots, T_n\}$ – the set of local process routes operations times,
$BE = (\Theta, \Psi, \Xi)$ - the variables describing SCCP behaviour
$\Theta = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ – the set of dispatching priority rules, $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$ – the sequence of relative frequencies of local processes mutual executions, Ξ - the number of Ψ occurrences within a cycle,
$SE = \{eq_{i,j,k}(ST, BE) \mid i = 1, .., n;\ j = 1, \ldots, lr(i); k = 1, \ldots, \Xi\}$ - the set of constraints (equations) linking ST and BE. Each $eq_{i,j,k}(ST, BE)$ describes, the time relation between the moments $x_{i,j,k}$ of operations $o_{i,j}$ beginning.

In other words, the constraints $eq_{i,j,k}(ST, BE)$ specify the relation linking X, Tc while encompassing SCCP's cyclic behavior, where: $X = \{X_1, X_2, \ldots, X_i, \ldots, X_n\}$, $X_i = (x_{i,1,1}, \ldots, x_{i,lr(i),1}, \ldots, x_{i,1,\Xi}, \ldots, x_{i,lr(i),\Xi})$, $x_{i,j,k}$ - the moment of operation $o_{i,j}$ beginning in the l-th cycle: $x_{i,j,k}(l) = x_{i,j,k} + l \cdot Tc$, $l \in \mathbb{Z}$, $(x_{i,j,k}(\mathrm{l}) \in \mathbb{Z}$ – the moment the operation $o_{i,j}^k$ starts its execution in the l-th cycle). Tc denotes the SCCP periodicity: $Tc = x_{i,j,k}(l+1) - x_{i,j,k}(l)$.

For illustration let us consider an example (see Fig. 2) describing the operation $o_{1,3}$ (executed by P_1 on the resource R_1) which can be started (i.e., its first execution; $k = 1$) only if the preceding operation $o_{1,2}$ (executed by P_1 on R_6) has been completed $(x_{1,2,1} + t_{1,1})$ and the resource R_1 has been released, i.e. if the process P_2 occupying the resource R_1 starts its subsequent operation at $x_{2,6,1} - Tc + 1$. So, the relation considered $eq_{1,3,1}(ST, BE)$ can be specified by the following formulae:

$$x_{1,3,1} = max\{(x_{2,6,1} - Tc + 1);\ (x_{1,2,1} + t_{1,2})\} \tag{5}$$

Besides of (5) the tab. 3 contains the rest of constraints SE describing the SCCP from Fig. 1. For all the constraints the following principle holds: the moment of the operation $o_{i,j}$ beginning states for a maximum of the completion time of operation $o_{i,j-1}$ preceding $o_{i,j}$ and the release time of the resource $p_{i,j}$ awaiting for $o_{i,j}$ execution.

$$\begin{gathered} eq_{i,j,k}(ST, BE)\text{: moment of operation } o_{i,j} \text{ begining} = \\ max\{\text{moment of } p_{i,j} \text{ release, moment of operation } o_{i,j-1} \text{completion}\}, \\ i = 1, .., n;\ j = 1, \ldots, lr(i);\ k = 1, \ldots, \Xi \end{gathered} \tag{6}$$

The constraints (6) taking into account multiple executions of process in one cycle can be seen as extension of constraints presented in [4]. Moreover, they can be also illustrated in terms of operations precedence digraph [4], vertices of which (placed along the time axis) correspond to the moments $x_{i,j,k}$ of operation $o_{i,j}$ beginning and the arcs determine their execution order. An example of the operations precedence digraph for the SCC from Fig. 1 is shown in Fig. 3.

Problem Formulation: Given the model (2) describing the SCCP specified by the set of routes M, the set of resources R, the operation times T, the set of priority

dispatching rules Θ, the sequence of relative frequencies of local processes mutual executions Ψ, the number of Ψ occurrences within a cyclic steady state Ξ, and the set of constraints linking above mentioned variables. The main question regards of cyclic steady state reachability and can be stated as follows: Does there exist the set X following the all constraints SE? or What is the cycle Tc of the SCCP behavior?

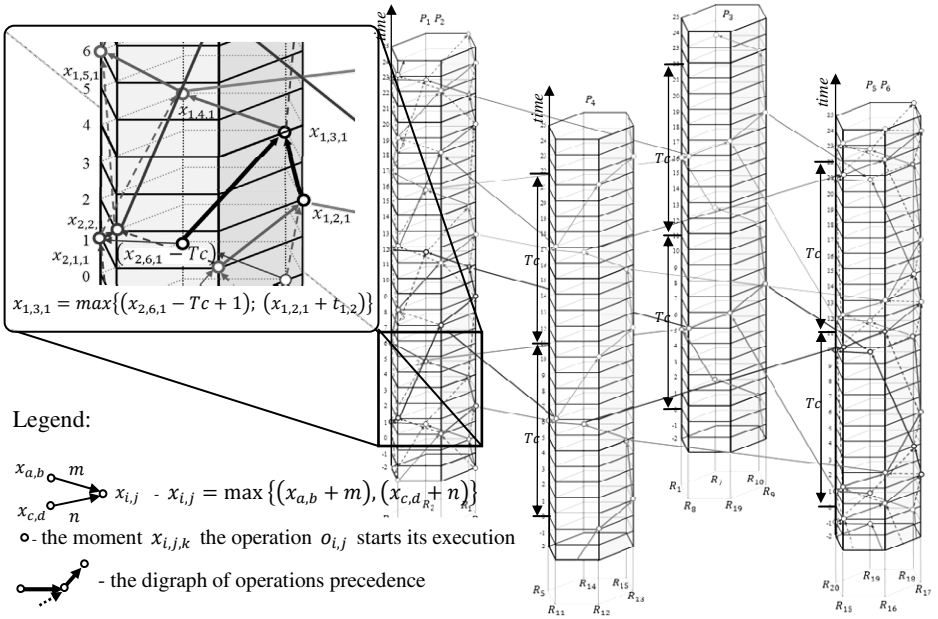


Fig. 3. The operations precedence digraph for SCCP from Fig. 1

3 Cyclic Steady States Space

Constraints satisfaction problem (SCP) can be used as a formal representation of the above stated problem. Consider CSP (7):

$$CS = (\{X, Tc\}, \{D_x, D_{Tc}\}, SE)) \tag{7}$$

where: $x_{i,j,k}$ – the moment of the operation $o_{i,j}$ beginning in the k-th execution where: X, Tc – decision variables, $D_x = \{D_{x_{i,j,k}} | D_{x_{i,j,k}} = \mathbb{N}, i = 1, .., n;\ j = 1, \dots, lr(i);\ k = \{1, \dots, \Xi\}$, $D_{Tc} = \mathbb{N}$, – domains of decision variables X, Tc SE - the set of constraints linking ST and BE (e.g. see Table. 3)

In order to illustrate an approach proposed let us consider the SCCP from Fig. 1 while assuming $\Psi = (1,1,1,1,1,1)$, $\Xi = 1$ and SE specified by Table 3. The solution obtained in the OzMozart environment consists of (X^i, Tc^i):

$$X^1 = \{X_1 = (0,1,2,3,5,8), X_2 = (0,1,7,8,9,10), X_3 = (10,0,1,2,5,9,),$$
$$X_4 = (1,2,8,9,10,0), X_5 = (10,0,4,5,6,7), X_6 = (1,0,3,4,5,6)\}, Tc^1 = 11.$$

Table 3. Constraints for the SCCP (from Fig. 1)

R_3:	$x_{2,1,1} = \max\left(x_{1,6,1} - Tc + 1; x_{2,6,1} - Tc + x_{2,6,1}\right)$ $x_{1,5,1} = \max\left(x_{2,1,1} + 1;\ x_{1,4,1} + t_{1,4,1}\right)$	R_4:	$x_{2,2,1} = \max\left(x_{1,1,1} + 1; x_{2,1,1} + t_{2,1,1}\right)$ $x_{1,6,1} = \max\left(x_{2,3,1} + 1; x_{1,5,1} + t_{1,5,1}\right)$
R_5:	$x_{1,1,1} = \max\left(x_{2,4,1} - Tc + 1;\ x_{1,6,1} - Tc + t_{1,6,1}\right)$ $x_{4,5,1} = \max\left(x_{1,2,1} + 1; x_{4,4,1} + t_{4,4,1}\right)$ $x_{2,3,1} = \max\left(x_{4,6,1} + 1;\ x_{2,2,1} + x_{2,2,1}\right)$	R_6:	$x_{1,2,1} = \max\left(x_{2,5,1} - Tc + 1; x_{1,1,1} + t_{1,1,1}\right)$ $x_{2,4,1} = \max\left(x_{1,3,1} + 1; x_{2,3,1} + t_{2,3,1}\right)$
R_1:	$x_{1,3,1} = \max\left(x_{2,6,1} - Tc + 1;\ x_{1,2,1} + t_{1,2,1}\right)$ $x_{3,3,1} = \max\left(x_{1,4,1} + 1;\ x_{3,2,1} + t_{3,2,1}\right)$	R_2:	$x_{1,4,1} = \max\left(x_{2,1,1} + 1;\ x_{1,3,1} + t_{1,3,1}\right)$ $x_{2,6,1} = \max\left(x_{1,5,1} + 1;\ x_{2,5,1} + t_{2,5,1}\right)$
R_{17}:	$x_{6,1,1} = \max\left(x_{5,6,1} - Tc + 1;\ x_{6,6,1} - Tc + t_{6,6,1}\right)$ $x_{5,5,1} = \max\left(x_{6,2,1} + 1; x_{5,4,1} + t_{5,4,1}\right)$	R_{18}:	$x_{5,6,1} = \max\left(x_{6,3,1} + 1;\ x_{5,5,1} + t_{5,5,1}\right)$ $x_{6,2,1} = \max\left(x_{5,1,1} + 1;\ x_{6,1,1} + t_{6,1,1}\right)$
R_{19}:	$x_{5,1,1} = \max\left(x_{6,4,1} - Tc + 1;\ x_{5,6,1} - Tc + t_{5,6,1}\right)$ $x_{3,5,1} = \max\left(x_{5,2,1} + 1; x_{3,4,1} + t_{3,4,1}\right)$	R_{20}:	$x_{5,2,1} = \max\left(x_{6,5,1} - Tc + 1;\ x_{5,1,1} + t_{5,1,1}\right)$ $x_{6,4,1} = \max\left(x_{5,3,1} + 1;\ x_{6,3,1} + t_{6,3,1}\right)$
R_{15}:	$x_{5,3,1} = \max\left(x_{6,6,1} - Tc + 1;\ x_{5,2,1} + t_{5,2,1}\right)$ $x_{4,3,1} = \max\left(x_{5,4,1} + 1;\ x_{4,2,1} + t_{4,2,1}\right)$	R_{16}:	$x_{5,4,1} = \max\left(x_{6,1,1} + 1;\ x_{5,3,1} + t_{5,3,1}\right)$ $x_{6,6,1} = \max\left(x_{5,5,1} + 1; x_{6,5,1} + t_{6,5,1}\right)$
R_{12}:	$x_{4,1,1} = \max\left(x_{4,6,1} - Tc + 1; x_{4,6,1} - Tc + t_{4,6,1}\right)$	R_{13}:	$x_{4,2,1} = \max\left(x_{4,1,1} + 1; x_{4,1,1} + x_{4,1,1}\right)$
R_{14}:	$x_{4,4,1} = \max\left(x_{4,3,1} + 1; x_{4,3,1} + t_{4,3,1}\right)$	R_{11}:	$x_{4,6,1} = \max\left(x_{4,5,1} + 1; x_{4,5,1} + t_{4,5,1}\right)$
R_{10}:	$x_{3,1,1} = \max\left(x_{3,6,1} - Tc + 1; x_{3,6,1} - Tc + t_{3,6,1}\right)$	R_7:	$x_{3,2,1} = \max\left(x_{3,1,1} + 1; x_{3,1,1} + t_{3,1,1}\right)$
R_8:	$x_{3,4,1} = \max\left(x_{3,3,1} + 1; x_{3,3,1} + t_{3,3,1}\right)$	R_9:	$x_{3,5,1} = \max\left(x_{3,5,1} + 1; x_{3,5,1} + t_{3,5,1}\right)$

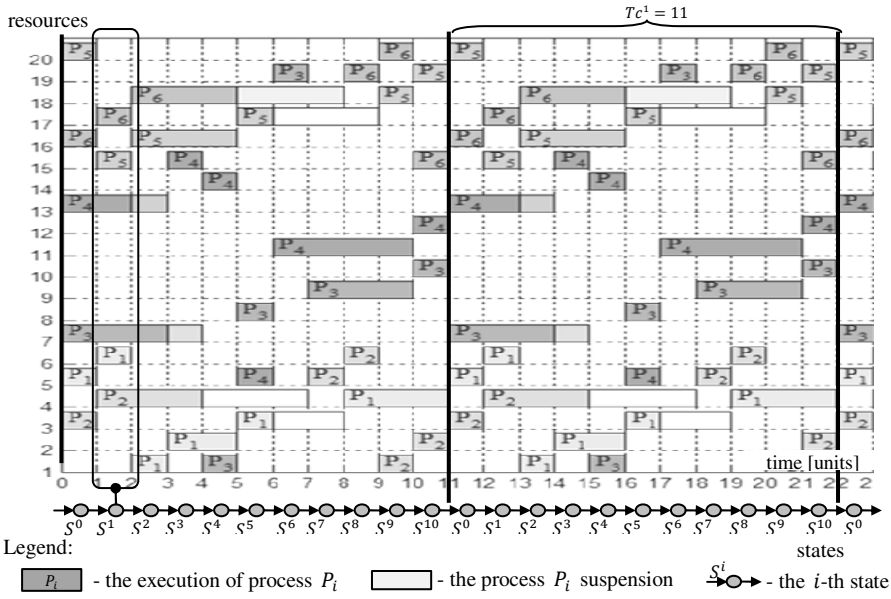


Fig. 4. Gantt's diagram for SCCP from Fig. 1 ($\Psi = (1,1,1,1,1,1)$, $\Xi = 1$)

The graphical illustration of solution obtained, i.e. encompassing cyclic behavior of the SCCP modeled, is shown in Fig. 4. Note that each case $\Xi \in \{1,2,3\}$ results in the same solution, see the Fig. 5a).

However, in case of slight change of an operation time (e.g., see the Table 2), the results differ each other dramatically. In case $\Xi \in \{1,3\}$ does not exist any cyclic steady state. In turn, for $\Xi = 2$ the newly obtained cyclic steady state $\left(M, R, T', (\Theta, \Psi, 2)\right)$ has the cycle $Tc^2 = 23$, see the Fig. 5b).

The solution obtained guarantees the same processes allocation, and execution of the same operations repeats with the cycle Tc. That means the same states (e.g. encompassing processes allocation, resources usability, resources availability) repeat as well. An assumption imposing the operation times are multiples of a unit period time implies that number of states STc creating the cyclic steady state has to follow: $STc \leq Tc$.For the sake of illustration let us consider an admissible state S of the SCCP defined as the triple [3]:

$$S = (A, Z, Q) \tag{8}$$

where: $A = (a_1, a_2, \ldots, a_m)$ – the processes allocation,
$Z = (z_1, z_2, \ldots, z_m)$ – the sequence of semaphores determining the resources availability, $Q = (q_1, q_2, \ldots, q_m)$ – the sequence of semaphore indices.
Z and Q encompassing the assumed Θ enable to define a next state function, which in turn allows one to generate a state space for assumed initial state S^0.

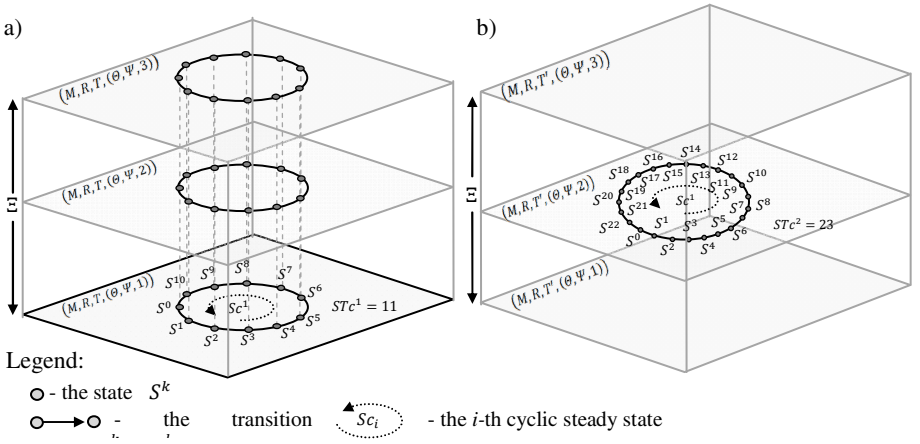


Fig. 5. Graphical illustration of cyclic steady states space for T (for operation times from Tab. 1), $\Psi = (1,1,1,1,1,1), \Xi \in \{1,2,3\}$ a) and T' (Tab. 2), $\Psi = (1,1,1,1,1,1), \Xi \in \{1,2,3\}$ b).

In order to summarize the above comments, a solution to the problem (7) provides the state space $\mathbb{S}^1$ (determined by the pair (X^1, Tc^2)) that can be reachable from any of its states $S \in \mathbb{S}^1$. Of course, state spaces generated in this way are cyclic steady

states. That means, the possible cyclic steady states following different solutions of the problem (7) create the reachable cyclic steady states space $\{\mathbb{S}^1, \mathbb{S}^2,..., \mathbb{S}^{ls}\}$ encompassing the all possible cyclic behaviors of the SCCP modeled.

4 Concluding Remarks

The main advantages to using a declarative framework are the availability of existing techniques and the expendability of constraint-based representations. In case considered such approach has been employed for modeling of SCCP systems and then for studying of their cyclic steady states space reachability. Searching for a set of possible cyclic steady states encompassing potential cyclic behaviors of the SCCP at hand can be useful in many tasks aimed at cyclic scheduling.

Each cyclic steady state characterized by its cycle time specifies the local processes repeatability, i.e. their periodicity. In that context study of systems composed of local cyclic processes lead to two fundamental questions: Does there exist a control procedure (i.e. a set of dispatching rules and an initial state) guaranteeing an assumed steady cyclic state subject to SCCP's structure constraints? Does there exist the SCCP's structure such that an assumed steady cyclic state can be achieved? Response to these questions determines our further works.

References

1. Bocewicz, G., Wójcik, R., Banaszak, Z.: On Undecidability of Cyclic Scheduling Problems. In: Karagiannis, D., Jin, Z. (eds.) KSEM 2009. LNCS, vol. 5914, pp. 310–321. Springer, Heidelberg (2009)
2. Bocewicz, G., Wójcik, R., Banaszak, Z.A.: Toward Cyclic Scheduling of Concurrent Multimodal Processes. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS (LNAI), vol. 6922, pp. 448–457. Springer, Heidelberg (2011)
3. Bocewicz, G., Wójcik, R., Banaszak, Z.A.: Cyclic Steady State Refinement. In: Abraham, A., Corchado, J.M., González, S.R., De Paz Santana, J.F. (eds.) International Symposium on DCAI. AISC, vol. 91, pp. 191–198. Springer, Heidelberg (2011)
4. Bocewicz, G., Banaszak, Z.: Declarative approach to cyclic scheduling of multimodal processes. In: Golińska, P. (ed.) EcoProduction and Logistics, vol. 1. Springer, Heidelberg (in print, 2012)
5. Liebchen, C., Möhring, R.H.: A case study in periodic timetabling. Electronic Notes in Theoretical Computer Science 66(6), 21–34 (2002)
6. Levner, E., Kats, V., de Pablo, D.A.L., Cheng, T.C.E.: Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers & Industrial Engineering 59(2), 352–361 (2010)
7. Polak, M., Majdzik, P., Banaszak, Z., Wójcik, R.: The performance evaluation tool for automated prototyping of concurrent cyclic processes. Fundamenta Informaticae 60(1-4), 269–289 (2004)
8. Song, J.-S., Lee, T.-E.: Petri net modeling and scheduling for cyclic job shops with blocking. Computers & Industrial Engineering 34(2), 281–295 (1998)

9. Heo, S.-K., Lee, K.-H., Lee, H.-K., Lee, I.-B., Park, J.H.: A New Algorithm for Cyclic Scheduling and Design of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 42(4), 836–846 (2003)
10. Trouillet, B., Korbaa, O., Gentina, J.-C.K.: Formal Approach for FMS Cyclic Scheduling. IEEE SMC Transactions, Part C 37(1), 126–137 (2007)
11. Wang, B., Yang, H., Zhang, Z.-H.: Research on the train operation plan of the Beijing-Tianjin inter-city railway based on periodic train diagrams. Tiedao Xuebao/Journal of the China Railway Society 29(2), 8–13 (2007)
12. Von Kampmeyer, T.: Cyclic scheduling problems, Ph.D. Dissertation, Fachbereich Mathematik/Informatik, Universität Osnabrück (2006)

Caption Text and Keyframe Based Video Retrieval System

Dung Mai and Kiem Hoang

University of Information Technology
Thu Duc Dist., Ho Chi Minh City, Viet Nam
{dungmt,kiemhv}@uit.edu.vn

Abstract. In this paper, we present a framework for video retrieval using caption text and keyframe similarity. To extract caption text, we applied methods detecting and extracting image areas contain caption text and we used Tesseract-OCR engine to convert into plain text, then use Hunspell library for spell words. Next, we used Clucene search engine index and query on this text. We applied shape descriptors APR and ECM to descript keyframes of the video shots and use those descriptors as a feature vector of video shots. From the feature vectors were obtained, we used ANN library to index and search. The system which is built on the web-based application using ASP.NET support keyword-based and keyframe-based query. The results obtained from experiments produced very promising.

Keywords: caption text, keyframe, shape descriptor, video retrieval.

1 Introduction

Recently, a tremendous increase of multimedia information has raised the need for automatic information indexing and retrieval systems. To enable search and retrieval of video, we need a good description of video content. There are many approaches and technologies for automatically parsing video, audio, and text to identify meaningful composition structure and to extract and represent content attributes of any video sources [1–4]. [5] and [6] are good surveys about content-based video retrieval.

In this work, we focus on caption text in video shots and its keyframes for built a video retrieval system. There are some reasons to chose: first, caption text is usually closely related to video content, it enables both keyword and free-text based search, e.g., we can identify video frames on specific topics of discussion from an educational video if the frames display corresponding text information; second, keyframes is a good representation of all the frames in the video shot, it can be used to distinguish videos from each other, and to provide access points into them.

The contributions of this paper are:

– Propose video retrieval system using caption text and keyframe.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 244–252, 2012.

– We innovatively combined the *Tesseract-OCR engine*[1] and *Hunspell SpellCheck*[2] into the process text extraction to improve performance and accuracy.

Our system allows users to query in two ways: enter text for the query with their choice of keywords or upload an external image in our system and then execute the query. The system will return relevant video shots.

The paper is organized as follows. The next section reviews related works. The third section describes our video retrieval system. In this section, we detail the system architecture and caption text extraction. The results are presented in section 4 and we conclude in the last section of the paper.

2 Related Work

Text in video is usually related to video content. Hence, text is often considered to be a strong candidate for use as a feature in indexing and content-based retrieval. Recently, many methods have been developed for extracting text from still images and videos. In this section, we give an overview of some significant related works:

Jawahar et al present an approach that enables search based on the textual information present in the video [2]. The authors proposed an approach that enables matching at the image-level and thereby avoiding an OCR. Results are shown from video collections in English, Hindi and Telugu.

Lienhart and Jung et al surveyed large number of techniques to address problem of text information extraction, the purpose is to classify and review these algorithms, discuss benchmark data and performance evaluation, and to point out promising directions for future research [3,7].

Anthimopoulos et al have presented a two-stage system for text detection in video images [8]. The system consists of a very efficient, edge-based, first stage with high recall and a second machine learning refinement stage which improves performance by reducing the false alarms, is based on a sliding window, an SVM classifier and a saliency map.

Langlois et al performed simultaneous analysis of video, subtitles and audio streams is performed in order to index, visualize and retrieve excerpts of video documents that share a certain emotional or semantic property [9].

In addition, there are many video shots in a video clip. A shot is considered as the basic unit of video. Keyframe(s) can be used to describe an entire video shot. In the following paragraph, we present a brief overview of some methods based keyframe for video retrieval.

Peng et al presented a keyframe based video summary system. Their method based on solely the visual saliency performs well on surveillance videos [4].

Pickering et al generated a large numbers of features for each of the key frames. The retrieval performance of two learning methods: boosting and k-nearest neighbors, is compared against a vector space model [10]. Browne et al

[1] `http://code.google.com/p/tesseract-ocr/`
[2] `http://hunspell.sourceforge.net/`

introduced Fischlar-Simpsons system which provides retrieval from an archive of video using any combination of text searching, keyframe matching, shot-level browsing, as well as object-based retrieval [11].

Sze et al proposed an optimal keyframe representation scheme based on global statistics for video shot retrieval [12].

3 Video Retrieval System

3.1 Overview of System Architecture

Our proposed system consists of two main stages: feature extracting and indexing, and video retrieval. The first stage is illustrated in figure 1.

Firstly, videos are collected in the *video collections* step then shots are detected in *shot detecting* component. From here, a set of *video shots* are can be processed in parallel: i) *Text extraction*: caption from video shots are extracted into *text files* then indexed using *Text Indexing*; ii) *Keyframe extraction*: in this step, *keyframes*

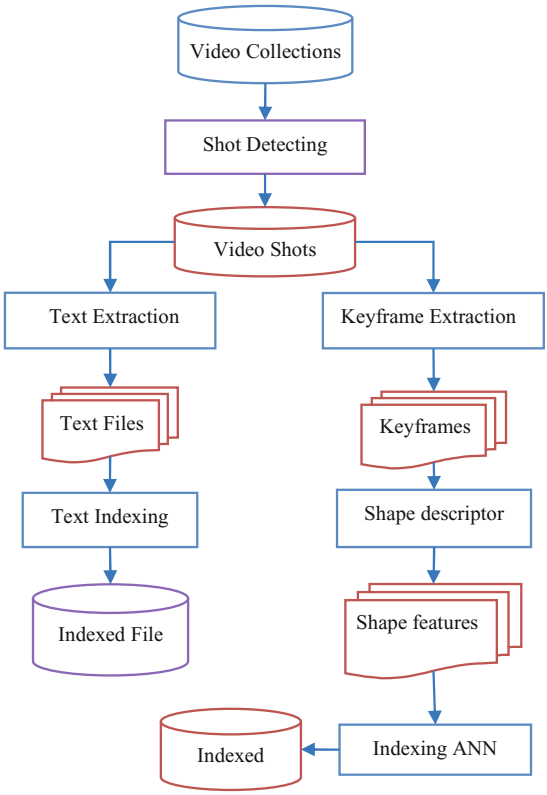


Fig. 1. The architecture of proposed system

are extracted then put into *shape descriptor* to produce a set of *shape features*. At this point, shape features are indexed using *Indexing ANN*.

The second stage, users can access video shots in system by keyword or image query. The user enters text query or choose a photo that contains the query information. System search and return expected results.

3.2 Extracting Caption Text

We know that text in video can be used to represent the content of video. Basically, text in video is divided into two kinds: scene text and caption text [7] (e.g. in figure 2). Scene text is the text that appears in a scene unintentionally, such as street names, road signs, store names. Therefore, it is difficult to detect and correct identification. In contrast, caption text (or overlay text, artificial text) is inserted in the video processing stage in order to illustrate a specific purpose or describe the content of the video or give additional information related to it. So they contain important information, suitable for indexing and retrieval.

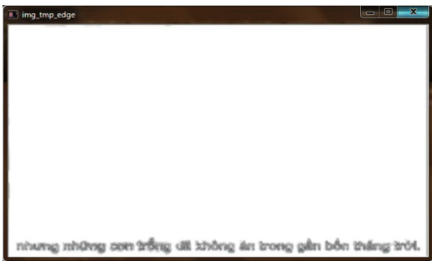
Fig. 2. Example of scene text and caption text

To detect caption text in frame, we use Sobel edge detector in order to find the edge magnitude of the image intensity (Fig. 3(a)).and then uses smooth filter to remove small edge and smooth the shape of the candidate text areas (Fig. 3(b)).

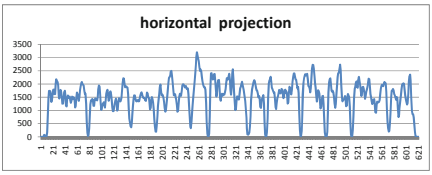
To detect line text, we used the horizontal and vertical projection. In horizontal projection, lines with projection values below a threshold are discarded. In this way boxes with more than one text line are divided and some lines with noise are also discarded (Fig. 3(c)). Similar approach, the vertical projection breaks every text line in parts only if the distance between them is greater than a threshold which depends on the height of the candidate text line (Fig. 3(d)). Then, initial bounding boxes contain (Fig. 3(e)) will be refined in order to the final bounding boxes (Fig. 3(f)). The final bounding boxes contains text are transformed into plain text using Tesseract-OCR engine. To improve the accuracy of words in plain text, we use the spell checker library Hunspell, which is a state of the art spell checker for languages with complex word compounding and rich morphology. Then, plain texts in same video shot will be stored in text file. We use Clucene search engine to index and query on this text file.[3]

[3] `http://clucene.sourceforge.net/`

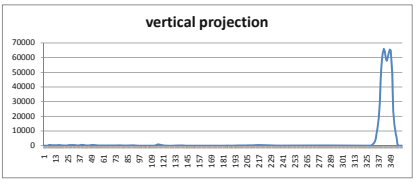
(a) Edge map



(b) Smooth & Binomial Direction



(c) horizontal projection



(d) Vertical projection

(e) Bounding boxes

(f) Final Bounding box

Fig. 3. Example of caption text extraction

3.3 Extract Keyframe

Keyframes play an important role in video. It is used to distinguish videos from each other, to summarize videos, and to provide access points into them. In this work, we use the technique proposed by [13] can determine any number of keyframes by clustering the frames in a video shot and by selecting a representative frame from each cluster. Temporal constraints determine which representative frame from each cluster is chosen as a keyframe of video.

In consequence, we apply shape descriptor to describe this keyframe and use them as feature vector of video shot.

3.4 Shape Descriptor

Shape is one of the fundamental visual features in the multimedia applications. The methods for shape description can be classified into two categories: contour-based and region-based methods. Contour-based method needs extraction of

boundary information which in some cases may not available. However, region-based method does not necessarily rely on shape boundary information, but they do not reflect local features of a shape.

In this section, we briefly describe two shape descriptors, ECM and ARP, which have been common used in image retrieval applications by its performance, and invariant to translation, rotation and scaling [14]. After the extracting, the resulting descriptors can be employed to index for video shot using ANN library.[4]

Edge Co-occurrence Matrix (ECM). The edge co-occurrence matrix was proposed by [15], contains second order statistics of edge directions in an image. The ECM is an extension of the co-occurrence matrix of edge directions that was proposed by [16]. The first step, we produce an edge image from the original gray level image. Next, the edges are detected in 8 directions and the direction of the strongest edge is selected for each pixel location.

Since the edges were detected in 8 directions, the size of the ECM is 8x8. The ECM entries are then used as elements in a feature vector.

Angular Radial Partitioning (ARP). Chalechale et al [17] proposed an approach for image representation based on geometrical distribution of edge pixels, this method yields the best retrieval performance when are compared with ART, HED, EHD and the Zernike moment invariants methods [14,18]

The algorithm uses the surrounding circle of image I for partitioning it to $M \times N$ sectors, where M is the number of radial partitions and N is the number of angular partitions.

The angle between adjacent angular partitions is $\theta = 2\pi/N$ and the radius of successive concentric circles is $\rho = R/M$; where R is the radius of the surrounding circle of the image. The number of edge points in each sector of I is chosen to represent the sector feature.

4 Experimental Results

4.1 Data Set

For experimental purpose, we have created our own data set as there is no standard data set available in the literature. In this data set, we have collected a variety of videos, including data set used in [19] contains four video files, and Youtube video files contain caption text in English and Vietnamese language. We have divided them into two categories: news and movies. There are 17,230 out of 21,052 video shots contain caption text.

4.2 Evaluation Methods

In this section, we present quantitative results on the performance of the caption text extraction system. In order to evaluate the performance of caption text extraction, we have used measure in terms of words. Let TW is words identified

[4] `http://www.cs.umd.edu/~mount/ANN/`

Table 1. Shows recall and precision of caption text extraction

	English		Vietnamese	
	News	Movies	News	Movies
TW	5729	6254	2761	3752
FW	552	1449	952	832
MW	821	1733	1179	950
Recall	0.87	0.78	0.70	0.80
Precision	0.91	0.81	0.74	0.82

correctly, FW is words identified incorrectly and MW is words missed in same video shot. Precision and recall are defined as:

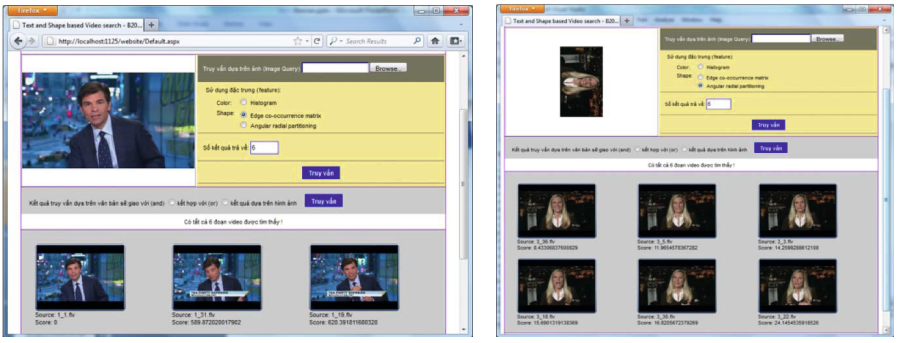
$$Precision = TW/(TW + FW) \text{ and } Recall = TW/(TW + MW)$$

In order to evaluate, we have chosen 100 video shots contain caption text in English language and 100 video shots contain caption text in Vietnamese language for two categories: news and movies. We have implemented a tool to compare the results of caption text extraction system with ground-truth data obtained by a manual method. The results are shown in Table 1. The performance of the proposed caption text extraction system is good enough to be used for indexing.

The results show the precision of caption text in English language is higher Vietnamese language, because Hullspell library and Tesseract-OCR are better performance in English. In addition, the data contains caption text in English are collected from news program in Vietnam television and movies from different sources. In contrast, Vietnamese data are selected from high quality movies, while the news is collected from different sources. Therefore, the caption text has different quality.

4.3 Results

We have built a web-based video retrieval system. Our system allows users to query in two ways: i) The user enters text for the query with his choice of



(a) Normal query image (b) scaled, rotated query image

Fig. 4. The screenshot show results retrieval with given query image

keywords. The system will return video shots contain text. Query results based on the results of search engine Clucene; ii) The user uploads an external image in our system and execute the query. The system will return video shots whose keyframe approximate nearest neighbor basis with the query image (based on shape descriptors). The results based on query results of the ANN library. The results are shown in Figure 4.

5 Conclusion and Future Work

We proposed a video retrieval system using caption text and keyframe of video shots with automatically extracting caption text from video shots and shape descriptor of keyframe for indexing and retrieval. Our experimental work produced very promising results. However, caption texts not always reflect the content of the video, such as: highlight news, advertising or some caption texts appear in last frames of shot i and first frames of shot $i + 1$. Those issues are difficult in text extraction. For future research, we will evaluate performance of our system to other similar systems and improve performance of caption text extraction and focus on the visual features and localization of semantic concepts for indexing and retrieving.

Acknowledgments. This research is partially supported by VNU-HCM R&D project: B2009-29-04, C2011-04-07.

References

1. Zhang, H., Wu, J., Zhong, D., Smoliar, S.W.: An integrated system for content-based video retrieval and browsing. Pattern Recognition, 643–658 (1997)
2. Jawahar, C.V., Chennupati, J.B., Paluri, B., Jammalamadaka, N.: Video retrieval based on textual queries. In: The 13th Intl Conference on Advanced Computing and Communications (2005)
3. Jung, K.: Text information extraction in images and video: a survey. Pattern Recognition 37(5), 977–997 (2004)
4. Peng, J., Xiao-Lin, Q.: Keyframe-based video summary using visual attention clues. IEEE Multimedia 17, 64–73 (2010)
5. Smoliar, S.W., Zhang, H.: Content-based video indexing and retrieval. IEEE MultiMedia 1(2), 62–72 (1994)
6. Dimitrova, N., Zhang, H.J., Shahraray, B., Sezan, I., Huang, T., Zakhor, A.: Applications of video-content analysis and retrieval. IEEE MultiMedia 9(3), 42–55 (2002)
7. Lienhart, R.: Video ocr: A survey and practitioner's guide. In: Video Mining, pp. 155–184. Kluwer Academic Publisher (2003)
8. Anthimopoulos, M., Gatos, B., Pratikakis, I.: A two-stage scheme for text detection in video images. Image Vision Comput. 28(9), 1413–1426 (2010)
9. Langlois, T., Chambel, T., Oliveira, E., Carvalho, P., Marques, G., Falcão, A.: Virus: video information retrieval using subtitles. In: Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2010, pp. 197–200. ACM, New York (2010)

10. Pickering, M.J., Rüger, S.: Evaluation of key frame-based retrieval techniques for video. Comput. Vis. Image Underst. 92(2-3), 217–235 (2003)
11. Browne, P., Smeaton, A.F.: Video retrieval using dialogue, keyframe similarity and video objects. In: ICIP (3), pp. 1208–1211 (2005)
12. Sze, K.W., Lam, K.M., Qiu, G.: A new key frame representation for video segment retrieval. IEEE Transactions on Circuits and Systems for Video Technology 15(9), 1148–1155 (2005)
13. Girgensohn, A., Boreczky, J.: Time-constrained keyframe selection technique. Multimedia Tools Appl. 11(3), 347–358 (2000)
14. Veltkamp, R.C., Latecki, L.J.: Properties and performances of shape similarity measures. In: Content-Based Retrieval (2006)
15. Rautkorpi, R., Iivarinen, J.: A Novel Shape Feature for Image Classification and Retrieval. In: Campilho, A.C., Kamel, M.S. (eds.) ICIAR 2004, Part I. LNCS, vol. 3211, pp. 753–760. Springer, Heidelberg (2004)
16. Brandt, S., Laaksonen, J., Oja, E.: Statistical shape features for content-based image retrieval. J. Math. Imaging Vis. 17(2), 187–198 (2002)
17. Chalechale, A., Mertins, A., Naghdy, G.: Edge image description using angular radial partitioning. IEE Proceedings Vision, Image and Signal Processing 151(2), 93–101 (2004)
18. Bober, M.: Mpeg-7 visual shape descriptors. IEEE Trans. Cir. and Sys. for Video Technol. 11(6), 716–719 (2001)
19. Anselmi, N.: Shot boundary detection in opencv. Wiki (2011), http://mmlab.disi.unitn.it/wiki/index.php/Shot_Boundary_Detection_in_OpenCV

E-Commerce Video Annotation Using GoodRelations-Based LODs with Faceted Search in Smart TV Environment

Trong Hai Duong[1], Ahmad Nurzid Rosli[2], Visal Sean[2], Kee-Sung Lee[2], and Geun-Sik Jo[1]

[1] Dept. of Computer and Information Engineering, Inha University, Korea
[2] School of Computer and Information Engineering, Inha University, Korea
haiduongtrong@gmail.com, {nurzid,seanvisal,lks}@eslab.inha.ac.kr, gsjo@inha.ac.kr

Abstract. TV-commerce is a new form of shopping that allows consumer to view, select and buy products from Smart TV. To do so, sellers annotate videos and associate it with information from online e-commerce systems in a semantic manner. In this work, we propose an e-commerce information derivation mechanism for video annotation using Linked Open Data (LOD) with faceted search. Annotation information is derived from e-commerce LODs, which linked distributed data across e-commerce web. We incorporated faceted search to allow consumer to easily make a information derivation query defined by GoodRelations ontology. The derived information is displayed as a faceted graph facilitating information choice.

Keywords: Linked open data, ontology, video annotation, e-commerce.

1 Introduction

With the blooming of TV-commerce, TV broadcasting companies will be the first to reap its benefits. As TV-commerce prospectively promises many economical, commercial and technical returns, it has gain the attention of a lot of companies and researchers, especially from the semantic web community. The organization and presentation of information to the buyers sitting in front of the TV will be one of the key to stand out in the TV-commerce and win the buyers. There had been many research activities with this focus. Silva et al. [2] in his research have pointed out on his research about the needs of information integration between buyer and seller. They have proposed a semantic information integration approach for agent-based electronic market based-on ontology by exploiting mapping paradigm which aligning consumer needs and market capacities. One of the main motivations is to build a rich, consistent and reusable semantics to handle increasingly complex e-commerce application. There is another approach

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 253–263, 2012.

by Chen et al. [3] through customer-oriented automatic cataloging construction model. They have developed an application called E-catalog. The application mainly to resolve the condition where e-catalog has to be constructed manually, which is of course a tedious job, time consuming and error prone. With the gradual increase of products information on TV-commerce, buyers will find it harder to make up their mind to choose the best item. This is because there is no direct interaction between the buyer and the seller on the products information, and the buyers are left alone to filter all the information bombarded at him/her. Therefore, Zhai et al. [4], incorporated fuzzy theory into ontology, where they have proposed a Fuzzy Domain Ontology Model to handle this uncertainty of information and knowledge. Mata et al. [5], have adopt a meshed-up technology to integrate Amazon, e-Bay and Google API to assist the buyer to find the best product according to different criteria such as product name, price, product evaluation and customer opinion. In order to infer the goods appear on the TV shows, they are few method have been discovered by few researchers. Li et al. [7] proposed a novel annotation framework based-on video object called Object-based Video Annotation. They have refined the detection results using Context Based Concept Fusion (CBCF) strategy that will allow online annotation from the Internet. They key factor that motivate them to construct this framework is because of the limited annotation vocabulary due to small training scale. So, what they do is divided the video into three types of annotations; 1) Human-based annotation, 2) web-based annotation and 3) Video analysis-based annotation. However, in previous year, Jeong et al. [8] proposed an automatic video annotation and summarization system called OLYVIA. Since traditional video indexing has a lot of problems, making it is difficult to search for a video content using high-level concept. To solve this problem, they employed the ontologies and semantic rules to facilitate video retrieval. High level concept of shot, group, scene and video level are automatically extracted by applying semantic rules to video annotation ontology and object ontology. There are various approaches in video annotation research works. One of the approaches is to develop and expended the multimedia ontology such as Large Scale Concept Ontology (LSCOM) with 1000 lexical concepts. It contains a list of concepts associated with multimedia, including images and videos to describe semantic media content [10]. Another approach was demonstrated in Video Event Representation Language (VERL), by describing video event representation and annotation based on ontology suitable for video content [11]. In this work, they addressing semantic retrieval and reasoning issues, where the description in the annotation does not define the data stream content adequately. Similar work in [12–14], eliminates the semantic gap in order to create associative value using semantic technologies. On the other hand, Bai et al. [23] presented a video semantic content analysis framework based on ontology. In this work, the key elements in video content analysis and support the detection process of the corresponding domain specific semantic content. The OWL language is used for knowledge representation for video analysis ontology and domain ontology.

Linked Data based video annotation is to enables users to mark up video with annotations using Semantic Web and Linked Data approaches [12–14, 19–21]. In the previous works, the annotator can creates the annotation by simply leaving the appropriate Linked Data URI. Their system is a frame based video annotation which only allows the annotation on the video frames. One of the major problems in the video content is lack of annotation on the objects appearances. Moreover, although the interface is provided for finding appropriate URIs to annotate, but still it is not effective and efficient enough since the results are derived from keyword search is huge and necessary to be filtered. In our proposed work, we provide a user friendly interface based-faceted search technique to let use filter and eliminate the amount of results while searching for appropriate product to annotate the object in the e-commerce video content.

2 Methodology

Faceted search recently gained a lot of attention to creates efficient interfaces [24].The user can interact with the facet interface by adding or removed the filter for information search. There are several efforts in faceted search research field [25, 26],but each have different focuses, assumptions, and limitations. A common point among existing methods involves manual facets creation that leads to the difficulty to generate information across LODs. The main focus for in our research is to automatically generate facets from GoodRelations ontology by matching concepts in LODs to concepts in GoodRelations in order to identify object properties and their values for facets of the corresponding concepts. Facets are mainly to filter the instances belonging to concepts to derive annotation information.

2.1 Facet Search Using Ontology-Based LOD

We assume a real world ($\boldsymbol{A}$, $\boldsymbol{V}$) where $\boldsymbol{A}$ is a finite set of attributes and $\boldsymbol{V}$ is the domain of $\boldsymbol{A}$. Also, $\boldsymbol{V}$ can be expressed as a set of attribute values, and $\boldsymbol{V} = \bigcup_{a \in \boldsymbol{A}} V_a$ where V_a is the domain of attribute a. In this chapter, we accept the following assumptions [1, 28, 29, 27]:

Definition 1 (Ontology). *An ontology is a triplet:*

$$O = (C, \sum, R) \tag{1}$$

where,

- **C***:is a set of concepts (the classes).*
- **R***: is a set of binary relations between the concepts from C.*
- $< C, \sum >$*: is the taxonomic structure of the concepts from C where* $\sum$ *is the collection of subsumption* ($\sqsubseteq$)*, equivalence* ($\equiv$)*, and disjointness* ($\coprod$) *relationships between two concepts from C.*

Definition 2 (Concept). *A concept c of an (***A***,* **V***)-based ontology is defined as a pair:*

$$c = (A^c, V^c) \tag{2}$$

where c is the unique identifier for instances of the concept. $A^c \subseteq \mathbf{A}$ *is a set of attributes describing the concept and* $V^c \subseteq \mathbf{V}$ *is the attributes' domain:* $V^c = \bigcup_{a \in A^c} V_a$.

Pair (A^c, V^c) is called the possible world or the structure of the concept c. Notice that within an ontology there may be two or more concepts with the same structure.

Definition 3 (Instance). *An instance of a concept* c *is described by the attributes from set* A^c *with values from set* V_c. *Thus, an instance of a concept* c *is defined as a pair:*

$$\text{instance} = (\text{id}, \text{v}) \tag{3}$$

where id *is a unique identifier of the instance in world* (A, V) *and* v *is the value of the instance, which is a tuple of type* A^c *and can be presented as a function:*

$$\text{v} : A^c \to V^c \tag{4}$$

such that $\text{v}(a) \in V^a$ *for* $a \in A^c$. *All instances of the same concept in an ontology are different with each other.*

By $\text{Ins}(O, c)$ *we denote the set of instances belonging to concept c in ontology O. We have*

$$\boldsymbol{I} = \bigcup_{c \in C} \text{Ins}(O, c) \tag{5}$$

In LOD, a description of a statement is represented by triples. A triple (s, p, o) where, s, p, and o stand for subject, predicate and object respectively. The subject of a triple is the URI identifying the described resource. The object can be the URI of another resource that is somehow related to the subject. The predicate indicates a specific relation exists between subject and object. A set of LOD triples is defined as a LOD graph.

Definition 4 (LOD Graph). *LOD graph is defined as a direct graph:*

$$G = (N, E) \tag{6}$$

where,

- $N = \{v_i | v_i \in I \cup C \cup V\}$*: the set of vertices in G.*
- $E = \{e(v_i, v_j) | v_i \in I \cup C, v_j \in N, e \in A \cup R \cup \sum\}$*: the set of directed edges presented the predicates of all LOD triples. Each LOD triple* (v_i, e, v_j)*, where* v_i*,* e*, and* v_j *are the subject, predicate and object in the LOD triple, respectively.*

To access to information organizing by an ontology - based LOD SPARQL-endpoint, the ontology should be provided for agents to be understandable its corresponding LOD. A SPARQL is created based on the ontology. A conjunctive query denoted as $Q(N_q, E_q)$ that is processed as a graph pattern [18]. Vertices of Q are comprising a set of variable vertices N_q^v and constant vertices (N_q^c), $N_q = N_q^v \cup N_q^c \subseteq N$. Edges of Q are formulae $e(v_i, v_j)$, with $vi \in N_q^v, v_j \in N_q$. A result of Q on the graph is a mapping from vertices of Q to vertices of G. The result set for Q which is a subgraph of G is denoted by $R(N_r, E_r)$. Similar to [18], we define Facets and Facet value as follows:

Definition 5 (Facets). *Let $R(N_r, E_r)$ be the result set for the query $Q(N_q, E_q)$ and $N_r^x \subseteq N_r$ be the particular set of vertices being obtained by bindings for variable $x \in N_q^v$. We have:*

- *Facets for the variable x,*

$$F(x) = \{e(v_i, v_j) | e(v_i, v_j) \in E, v_i \in N_r^x, v_i \neq v_j, v_j \in N_r^x \cup N\} \quad (7)$$

- *The set of facets for Q,*

$$F(Q) = \{F(x) | x \in V_q^v\} \quad (8)$$

Facets $F(x)$ (for the variable x) are labels of edges, which capture direct connections between elements in N_r^x and other elements of the LOD graph.

Definition 6 (Facet value). *Let $R(V_r, E_r)$ be the result set for the query $Q(N_q, E_q)$ and $V_r^x \in V_r$ be the particular set of vertices being obtained by bindings for variable $x \in V_q^v$. A set of faceted values is defined:*

$$- F(x) = \{v_j | v_i \in N_r^x, e(vi, vj) \in E\} \quad (9)$$

2.2 Video Annotation Using GoodRelations-Based LODs with Faceted Search

GoodRelations Ontology. There are two kinds of ontologies used to annotate videos. The first one describes visual feature media objects such as color, texture, shape, motion, and position (i.e. visual feature media ontology such as MPEG-7 [15]). Another one provides a knowledge-base of a specific domain for annotating video content (domain ontology). In this work, we employ the GoodRelations [17] ontology as domain ontology. GoodRelations is the most powerful vocabulary for publishing all of the details of the products and services. It serves conceptual model for a consolidated view on commerce data on the Web [16]. The ontology is flexible, moderate in size, and supports value interval plus existential quantification while posing minimal requirement on the reasoning support of the ontology management infrastructure. There are numbers of tools developed to exploit the GoodRelations ontology [17], such as; 1) GoodRelations Annotator - a form based tool that can be used by any business to create detailed description including products, payment and delivery option,

store locations, opening hours, and eligible customer types and eligible regions; 2) osCommerce Shop and Extensionprovides the GoodRelations4osCommerce extension that activates RDFa and RDF/XML export of the offer and item details. It also helps avoiding duplicate output of data in Web pages. GoodRelations have been used by more than 10,000 small and large shops world-wide including Google, Yahoo!, BestBuy, sears.com, and kmart.com. Many linked open commerce dataspaces (SPARQL endpoints) used GoodRelations as their schema (i.e. http://uriburner.com/sparql and http://loc.openlinksw.com/sparql).

Information Derivation Algorithm for Video Annotation. The most basic way to create an annotation is simply to pause the video at the appropriate frame, and select the media object in the frame. Each media object in a specific video can be considered as an instance of a domain concept belonging to the GoodRelations ontology. The annotator would choose a relevant concept to describe the media object. This concept is used to create a simple query inputting to selected SPARQL-endpoints. The relevant information of the concept are derived from the SPARQL-endpoints and displayed as a faceted graph. The properties and the values related to the concept are extracted from GoodRelation ontology to make facets. User can use the facets to filter the annotation information and choose a most relevant instance to add to the media object.

We apply faceted search to facilitate selecting the relevant instance. Facets are generated for each variable or concept using *Equation* 7. We can use facet functions to filter the result set to get the relevant results. According to Wagner et al [18], there are three functions that user can use to construct queries, change the faceted values and in order to modify the result set R:

- Focus selection: Using a focus selection operation, user can change the facets in the current query Q to other facets, which generates more relevant result. For example, users changes the facets from x to y. It is to focus on facets contained in $F(y)$ instead of $F(x)$: $F(Q') = F(Q') \setminus F(x) \cup F(y)$ then $R_w(V'_r, E'_r) \setminus (V^x_r, E^x_r) \cup (V^y_r, E^y_r)$.
- Refinement: Assuming the current query Q is performing on a variable x, users can add a facet belonging to $F(x)$ with a corresponding variable or faceted value to the query Q. This means user can add a predicate query $f(x, y) \in F(x)$ to current query Q where y is a variable or a constant. However, user can also modify a existing predicate query $f(x, y)$ by changing current variable or constant y. This operation would limit the result set or generate a more relevant result.
- Expansion: Users can expand a result set by expansion operation. This operation removes a facet or a faceted value such as replacing a constant with a variable.

To illustrate the aforementioned algorithm, we annotate a TV product in a video. To annotate a TV as an object in the video, a list of concepts related to the TV concept from GoodRelations ontology are generated as shown in part (2) of *Fig.*1. We assume that the LCD_TV is a most relevant concept and it is used to make a query to input to the SPARQL-endpoints for instance generation.

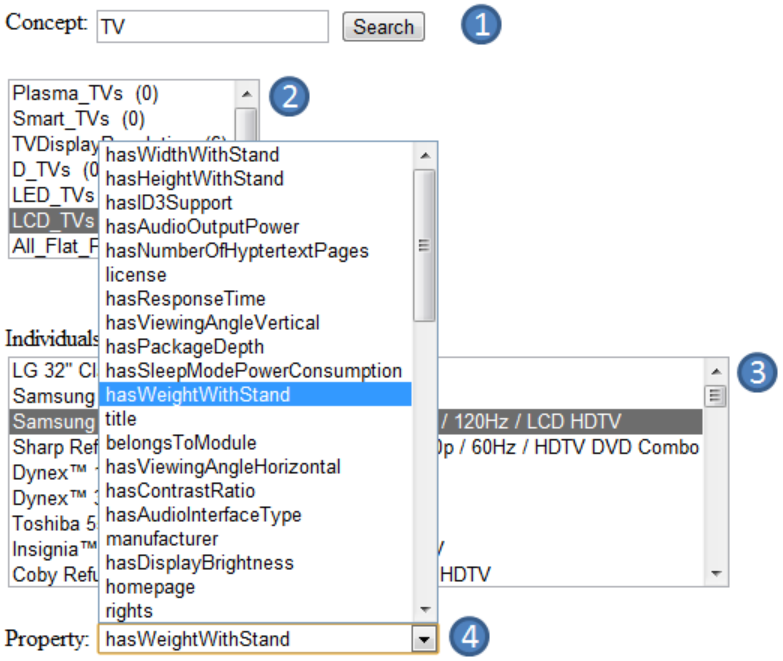


Fig. 1. An Annotation Process

All instances of the concept LCD_TV are given as shown in part (3). The annotator will choose an instance belonging to LCD_TV, add then this instance to the selected media object of TV product as its annotation information. However, it is difficult to select a relevant instance in the case of huge instances are returned. Therefore, the annotator can use above three functions to filter most relevant instance. Facets are generate from GoodRelations ontology as the concept's properties with values (see part (4) of *Fig.*1).

3 Implementation and Discussion

3.1 Implementation

We used BestBuy[1] and SPARQL-endpoints[2] as linked open commerce dataspaces. The GoodRelations ontology is used to access the commerce dataspaces and to create facets. We also used jena framework[3] and Tomcat server[4] for our

[1] `http://www.bestbuy.com/`
[2] `http://uriburner.com/sparql` and `http://loc.openlinksw.com/sparql`
[3] `http://jena.apache.org/`
[4] `http://tomcat.apache.org/`

implementation. In order to annotate the object in a video content, it is necessary to select an appropriate instances. The instances are selected from LOD through the GoodRelations ontology. The process on how to select the appropriate instance for annotation is described as follows. Our system let user first search the concept of the product object that he or she wants to annotate. After searching for the concept name of the product, our system shows the results as the list of all concepts that related to the query. Each concept is shown with the number of available instances within each concept as shown in *Fig* 2. By seeing the list of concepts, user can easily choose the appropriate concept. After adding select a specific concept, another list of individuals that belongs to that concept will be shown for user to select as shown in *Fig* 2. Individuals appeared in the list is shown using only the name property. User can click on each of them to see all the detail information and properties of that individual. The numbers of individuals that belong to each concept is huge. Therefore, in order to support user to easily find the right individual, our system provides user an option to filter and narrow down the number of individuals. This is done by using the available data or object properties of the concept. All the concept properties are retrieved based on GoodRelations ontology. The user can start to filter the results by selecting one or more properties. For the object property, user may simply select the existing value from the list. Meanwhile, for data property, they need to insert the input in order to allow them to proceed with filtering process. Hence,numbers of individual is reduced and user can easily select to right individual for the annotation. User can finalize the annotation by click on the save button and our system will save all the annotated information and other related metadata to our local repository.

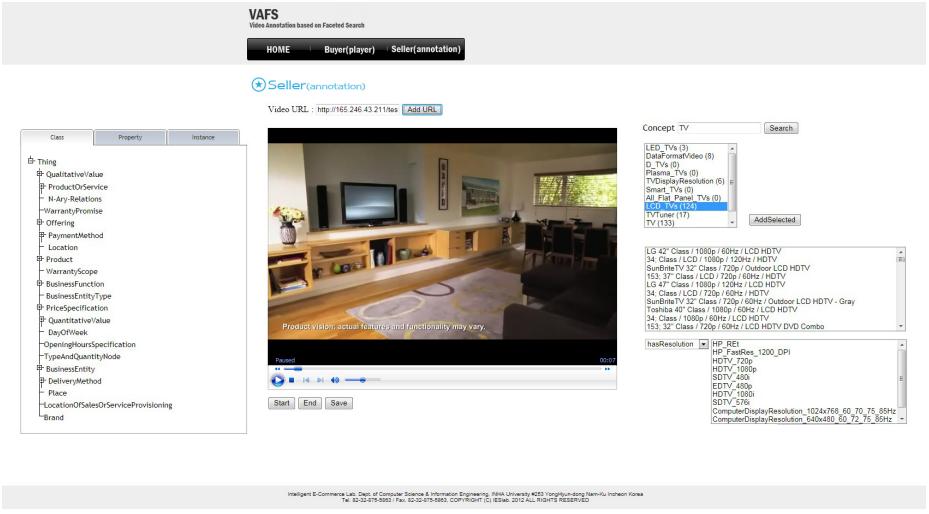


Fig. 2. Interface of Annotation System Using LOD

3.2 Discussion

Lambert and Yu [19] proposed a semantic video annotation system based on Linked Data which, known as Annomation. The system is used for annotating and publishing video resources using Linked Data identifier. It is developed as a web based application that enables users to watch a video and at the same time collaboratively participate to add annotations to the video content. The Annomation takes advantages of Linked Data, where things can be inferred via their unique Uniform Resource Identifier(URI)and more information can be acquired from them including the links. This system provides the most basic way to create an annotation. Users simply pause the video at the appropriate point, enter duration and add a Semantic Web/Linked Data URI. However, annotations can be created only limited to frame and not to the specific objects that appeared in the video content. Moreover, although the interface is provided for finding appropriate URIs to annotate, but still it is not effective and efficient enough since the results are derived from keyword search is huge and necessary to be filtered. In our proposed work, we provide a user friendly interface based-faceted search technique to let use filter and eliminate the amount of results while searching for appropriate product to annotate the object in the e-commerce video content.

Yuen and et al. [20] on the other hand, designed a video annotation system called LabelMe to allow users to creates annotations on the objects appeared in the video content. Furthermore, users are also able to annotate moving or static objects by outlining their shape using a drawing tool presented in their previous study [12]. However, LabelMe does not utilize a Semantic Web/Linked Data technique for annotation information, instead only a plain text to be annotated with the object in the video content. We address this problem by employing Linked Open data and GoodRelations to annotate the object.

Our work is comparable with Waitelonis and Sack [21] proposed an integrated database for the open academic video search platform known as Yovisto. They have introduced the LOD cloud and augmentation Yovisto video collection. The video data are also enrich with DBpedia dataset. Since Yovisto resources are connected with LOD, all video resources are connected amongst each other. With LOD, they are able to locate the interrelationships between authors or novelist of the videos, such as influences and influenced (by). This means Linked Data is used to complement Yovisto video data and to uncover implicit cross connections within them in order to improve user experience by enabling a semantically supported exploration search.

4 Conclusion

In this work, we proposed a methodology for annotation information derivation from e-commerce Linked Open Data (LOD) which linked distributed data across e-commerce web. Use of faceted search enables consumer to easily make query for exploring annotation information. The derived information is displayed as a faceted graph facilitating annotator to select most relevant information that is used to annotate media objects.

Acknowledgement. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korean government(MEST) (No. 2012-0005500).

References

1. Duong, T.H., Jo, G.S., Jung, J.J., Nguyen, N.T.: Complexity Analysis of Ontology Integration Methodologies: A Comparative Study. Journal of Universal Computer Science 15(4), 877–897 (2009)
2. Silva, N., Viamonte, M.J., Maio, P.: Agent-Based Electronic Market With Ontology-Services. In: Proceedings of IEEE International Conference on e-Business Engineering, pp. 51–58. IEEE Computer Society (2009)
3. Chen, D., Li, X., Liang, Y., Zhang, J.: Research on the Theory of Customer-Oriented E-Catalog Ontology Automatic Construction. In: The Smart Internet, pp. 2961–2964. IEEE Computer Society (2010)
4. Zhai, J., Shen, L., Liang, Y., Jiang, J.: Application of Fuzzy Ontology to Information Retrieval for Electronic Commerce. In: ISECS 2008, pp. 221–225. IEEE Computer Society (2008)
5. Mata, F., Pimentel, A., Zepeda, S.: Integration of heterogeneous data models: A Mashup for electronic commerce. In: International Conference on Management of eCommerce and eGovernment, pp. 168–171. IEEE Computer Society (2010)
6. Feng, Y., Xu, H., Fang, X.: An Intelligent Recommendation Method of E-Commerce Based on Ontology. In: BIFE 2009, pp. 592–594. IEEE Computer Society (2009)
7. Li, Y., Lu, J., Zhang, Y., Li, R., Zhou, B.: A Novel Video Annotation Framework Based on Video Object. In: JCAI 2009, pp. 572–575. IEEE Computer Society (2009)
8. Jeong, et al.: OLYVIA: Ontology-based Automatic Video Annotation and Summarization System using Semantic Inference Rules. In: Third International Conference on Semantics, Knowledge and Grid, pp. 170–175. IEEE Computer Society (2008)
9. Chen, D.L., Nie, G.H., Liu, P.F.: An Exploration of the Recommendation Systems and Knowledge Grid in E-Commerce. Journal of Information (4), 33–35 (2007)
10. Naphade, M., Smith, J.R., Tesic, J., Chang, S.F., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J.: Large-scale concept ontology for multimedia. IEEE Multimedia 13(3), 86–91 (2006)
11. Francois, A.R.J., Nevatia, R., Hobbs, J., Bolles, R.C., Smith, J.R.: VERL: an ontology framework for representing and annotating video events. IEEE Multimedia 12(4), 76–86 (2005)
12. Naphade, M.R., Smith, J.R.: On the detection of semantic concepts at TRECVID. In: Proc. ACM Multimedia, New York, NY, pp. 660–667 (2004)
13. Park, K.W., Lee, J.H., Moon, Y.S., Park, S.H., Lee, D.H.: OLYVIA: Ontology-based Automatic Video Annotation and Summarization System Using Semantic Inference Rules. In: Proceedings of the Third International Conference on Semantics, Knowledge and Grid, SKG 2007, pp. 170–175. IEEE Computer Society (2007)
14. Jeong, J.W., Hong, H.K., Lee, D.H.: Ontology-based Automatic Video Annotation Technique in Smart TV Environment. IEEE Transactions on Consumer Electronics 57(4), 1830–1836 (2011)
15. Sikora, T.: The MPEG-7 Visual standard for content description - an overview. IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on MPEG-7 11(6), 696–702 (2001)

16. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008)
17. Hepp, M., Radinger, A., Wechselberger, A., Stolz, A., Bingel, D., Irmscher, T., Mattern, M., Ostheim, T.: GoodRelations Tools and Applications. Poster and Demo Proceedings of the 8th International Semantic Web Conference, ISWC 2009, Washington, DC, USA, October 25 (2009)
18. Wagner, A., Ladwig, G., Tran, T.: Browsing-Oriented Semantic Faceted Search. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 303–319. Springer, Heidelberg (2011)
19. Lambert, D., Yu, H.Q.: Linked Data Based Video Annotation and Browsing for Distance Learning. In: Proceedings of the Second International Workshop on Semantic Web Applications in Higher Education (2010)
20. Yuen, J., Russell, B., Liu, C., Torralba, A.: LabelMevideo: building a video database with human annotations. In: Proceedings of IEEE International Conference on Computer Vision, pp. 1451–1458 (2009)
21. Waitelonis, J., Sack, H.: Augmenting video searchwith linked open data. In: Proc. of Int. Conf. on Semantic Systems 2009, i-Semantics 2009 (2009)
22. Yamamoto, D., Masuda, T., Ohira, S., Nagao, K.: Collaborative Video Scene Annotation Based on Tag Cloud. In: Huang, Y.-M.R., Xu, C., Cheng, K.-S., Yang, J.-F.K., Swamy, M.N.S., Li, S., Ding, J.-W. (eds.) PCM 2008. LNCS, vol. 5353, pp. 397–406. Springer, Heidelberg (2008)
23. Bai, L., et al.: Video semantic content analysis based on ontology. In: International Machine Vision and Image Processing Conference, IMVIP 2007, September 5, pp. 117–124 (2007)
24. Bonino, D., Corno, F., Farinetti, L.: FaSet: A Set Theory Model for Faceted Search. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT 2009, pp. 474–481 (2009)
25. Lown, C., Hemminger, B.M.: Extracting user interaction information form the transactions logs of a faceted navigation opac (June 2009), http://journal.code4lib.org/articles/1633
26. Yee, K.-P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Ft. Lauderdale, Florida, USA, April 05-10 (2003)
27. Danilowicz, C., Nguyen, N.T.: Consensus-based partitions in the space of ordered partitions. Pattern Recognition 21(3), 269–273 (1988)
28. Nguyen, N.T.: Inconsistency of Knowledge and Collective Intelligence. Cybernetics and Systems 39(6), 542–562 (2008)
29. Nguyen, N.T.: A Method for Ontology Conflict Resolution and Integration on Relation Level. Cybernetics and Systems 38(8) (2007)

Nearest Feature Line Discriminant Analysis in DFRCT Domain for Image Feature Extraction

Lijun Yan[1,⋆], Cong Wang[2], and Jeng-Shyang Pan[2]

[1] Harbin Institute of Technology
92 West Da-Zhi Street, Harbin, 150001, China
yanlijun@126.com
[2] Harbin Institute of Technology Shenzhen Graduate School
Xili University Town, NanShan, Shenzhen, China
jengshyangpan@gmail.com

Abstract. A novel subspace learning algorithm based on nearest feature line in time-frequency domain is proposed in this paper. The proposed algorithm combines neighborhood discriminant nearest feature line analysis and fractional cosine transform to extract the local discriminant features of the samples. A new discriminant power criterion based on nearest feature line is also presented in this paper. Some experiments are implemented to evaluate the proposed algorithm and the experimental results demonstrate the efficiency of the proposed algorithm.

Keywords: Nearest Feature Line, Image Feature Extraction, Fractional Cosine Transform.

1 Introduction

Image classification and related technology [1] have a variety of potential applications in information security, smart card, access control, etc. For instance, in the face recognition task, the number of training images points per person is smaller than that of the dimensionality of face images. High-dimensional face images lead to high computational complexity and overfitting. Dimensionality reduction is an effective way to alleviate it, and subspace learning algorithms have been widely used.

Principal Component Analysis (PCA)[2, 3], linear discriminant analysis (LDA)[4], and maximum margin criterion (MMC)[5] are among most popular subspace learning algorithms. PCA projects the original data to a low dimensional space, which is spanned by the eigenvectors associated with the largest eigenvalues of the covariance matrix of all samples. PCA is the optimal representation of the input samples in the sense of minimizing the mean squared error. However, PCA is an unsupervised algorithm, which may impair the recognition accuracy. LDA finds a transformation matrix U that linearly maps a high-dimensional sample to a low-dimension data, where $n < m$. LDA can calculate an optimal discriminant projection by maximizing the ratio of the trace

⋆ Corresponding author.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 264–272, 2012.

of the between-class scatter matrix to the trace of the within-class scatter matrix. LDA takes consideration of the labels of the input samples and improves the classification ability. However, LDA suffers from the small sample size (SSS) problem. Many effective approaches have been proposed to solve the problem.

Nearest feature line (NFL)[6] is a new classification tool, proposed by Li in 1998, firstly. In particular, it performs better when only limited samples are available for training. The basic idea underlying the NFL approach is to use all the possible lines consisting of any pair of feature vectors in the training set to encode the feature space in terms of the ensemble characteristics and the geometric relationship. As a simple yet effective algorithm, the NFL has shown its good performance in face recognition, audio classification, image classification, and retrieval. The NFL takes advantage of both the ensemble and the geometric features of samples for pattern classification. In contrast to a nearest neighbor (NN) classifier, the NFL makes better use of the ensemble information for classification[7–9].

While NFL has achieved reasonable performance in data classification, most existing NFL-based algorithms just use the NFL metric for classification and not in the learning phase. While classification can be enhanced by NFL to a certain extent, the learning ability of existing subspace learning methods remains to be poor when the number of training samples is limited. To address this issue, a number of enhanced subspace learning algorithms based on the NFL metric have been proposed, recently. For example, Zheng et al. proposed a nearest neighbour line nonparametric discriminant analysis (NNL-NDA)[10] algorithm, Pang et al. presented a nearest feature line-based space (NFLS)[11] method, and Lu et al. put forward an uncorrelated discriminant nearest feature line analysis (UDNFLA)[12]. Neighborhood discriminant nearest feature line analysis (NDNFLA)[13] is proposed to extract the local discriminant features of prototype samples. However, most of current algorithms runs in the time domain or space domain. A novel feature extraction algorithm in time-frequency domain is proposed in this paper.

The rest of the paper is organized as follows. In section 2, some preliminaries are given. In section 3, we give an introduction of the proposed methods. In section 4, a number of experiments are implemented to justify the superiority of the proposed algorithms. Conclusions are made in section 5.

2 Outline of DFRST and NDNFLA

2.1 DFRST

DFRST is a generalized form of the DST. The definition of DST kernel matrices is as follows.

$$S_{N-1} = \sqrt{\frac{2}{N}}[\sin(\frac{mn\pi}{N})] \tag{1}$$

for $m, n = 1, 2, \cdots, N-1$.

k_m in the above four definition is defined as follows.

$$k_m = \begin{cases} \frac{1}{\sqrt{2}}, m = 0 \text{and} m = N \\ 1, \qquad \text{others} \end{cases} \tag{2}$$

The DST kernel is symmytric and periodic with period 2. This means that repeated application of DST-I will get the original signal. Because of its good properties, DST-I was chosen to develop DFRST, as shown in formula (3).

$$S_N^I = \sqrt{\frac{2}{N+1}} \begin{bmatrix} \sin\frac{\pi}{N+1} & \sin\frac{2\pi}{N+1} & \cdots & \sin\frac{(N-1)\pi}{N+1} & \sin\frac{N\pi}{N+1} \\ \sin\frac{2\pi}{N+1} & \sin\frac{4\pi}{N+1} & \cdots & \sin\frac{2(N-1)\pi}{N+1} & \sin\frac{2N\pi}{N+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sin\frac{(N-1)\pi}{N+1} & \sin\frac{2(N-1)\pi}{N+1} & \cdots & \sin\frac{(N-1)^2\pi}{N+1} & \sin\frac{N(N-1)\pi}{N+1} \\ \sin\frac{N\pi}{N+1} & \sin\frac{2N\pi}{N+1} & \cdots & \sin\frac{N(N-1)\pi}{N+1} & \sin\frac{N^2\pi}{N+1} \end{bmatrix} \tag{3}$$

The N-point DFRST kernel is defined as

$$\begin{aligned} S_{N,\alpha} &= V_N D_N^{2\alpha/\pi} V_N^T \\ &= V_N \begin{bmatrix} 1 & \cdots & \cdots & 0 \\ \vdots & e^{-2j\alpha} & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & e^{-2j(N-1)\alpha} \end{bmatrix} V_N^T \end{aligned} \tag{4}$$

where $V_N = [v_1, v_3, \cdots, v_{2N-1}]$. v_k denotes the DST-I eigenvector.

Given an input signal $x \in R^N$, let D_x^α be the DFRST of x with order α, then

$$D_x^\alpha = S_{N,\alpha} x = V_N D_N^{2\alpha/\pi} V_N^T x \tag{5}$$

Discrete fractional cosine transform displays a signal varying from the time domain to the DST domain with order parameter changing from 0 to 1. It is a powerful time-frequency analysis tool. The detail properties of DFRST can be found in [14].

2.2 Nearest Feature Line

Nearest feature line is a classifier. It is first presented by Stan Z. Li and Juwei Lu. Given a training samples set, $X = \{x_n \in R^M : n = 1, 2, \cdots, N\}$, denote the class label of x_i by $l(x_i)$, the training samples sharing the same class label with x_i by $P(i)$, and the training samples with different label with x_i by $R(i)$. NFL generalizes each pair of prototype feature points belonging to the same class: $\{x_m, x_n\}$ by a linear function $L_{m,n}$, which is called the feature line. The line $L_{m,n}$ is expressed by the span $L_{m,n} = sp(x_m, x_n)$. The query x_i is projected onto $L_{m,n}$ as a point $x_{m.n}^i$. This projection can be computed as

$$x_{m,n}^i = x_m + t(x_n - x_m) \tag{6}$$

where $t = [(x_i - x_n)(x_m - x_n)]/[(x_m - x_n)^T(x_m - x_n)]$.

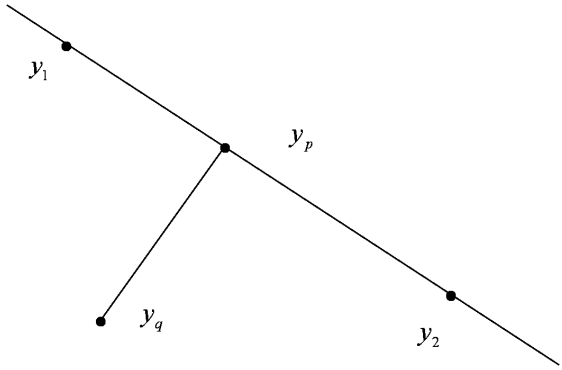


Fig. 1. Feature Line Distance

The Euclidean distance of x_i and $x^i_{m,n}$ is termed as FL distance. The less the FL distance is, the bigger probability that x_i belongs to the same class as x_m and x_n is. Fig. 1 shows a sample of FL distance. In Fig. 1, the distance between y_p and the feature line $L_{m,n}$ equals to the distance between y_q and y_p, where y_p is the projection point of y_q to the feature line $L_{m,n}$.

2.3 NDNFLA

Let's introduce two definitions firstly.

Definition 1. *Homogeneous neighborhoods: For a sample x_i, its k nearest homogeneous neighborhood N^o_i is the set of k most similar data which are in the same class with x_i.*

Definition 2. *Heterogeneous neighborhoods: For a sample x_i, its k nearest Heterogeneous neighborhoods N^e_i is the set of k most similar data which are not in the same class with x_i.*

In NDNFLA approach, the optimization problem is as follows:

$$\begin{aligned}\max J(W) = &(\sum_{i=1}^{N} \frac{1}{NC^2_{|N^e_i|}} \sum_{x_m,x_n\in N^e_i} \left\|W^T x_i - W^T x^i_{m,n}\right\|^2 \\ &- \sum_{i=1}^{N} \frac{1}{NC^2_{|N^o_i|}} \sum_{x_m,x_n\in N^o_i} \left\|W^T x_i - W^T x^i_{m,n}\right\|^2)\end{aligned} \tag{7}$$

Using matrix computation,

$$\begin{aligned}&\sum_{i=1}^{N} \frac{1}{NC^2_{|N^e_i|}} \sum_{x_m,x_n\in N^e_i} \left\|W^T x_i - W^T x^i_{m,n}\right\|^2 \\ &= \sum_{i=1}^{N} \frac{1}{NC^2_{|N^e_i|}} \sum_{x_m,x_n\in N^e_i} \mathrm{tr}[W^T (x_i - x^i_{m,n})(x_i - x^i_{m,n})^T W] \\ &= \mathrm{tr}\{W^T \sum_{i=1}^{N} \frac{1}{NC^2_{|N^e_i|}} \sum_{x_m,x_n\in N^e_i} [(x_i - x^i_{m,n})(x_i - x^i_{m,n})^T] W\}\end{aligned} \tag{8}$$

where tr denotes the trace of a matrix. Similar with the above,

$$\begin{aligned}&\sum_{i=1}^{N}\frac{1}{NC^2_{|N_i^o|}}\sum_{x_m,x_n\in N_i^o}\left\|W^Tx_i-W^Tx_{m,n}^i\right\|^2\\&=\mathrm{tr}\{W^T\sum_{i=1}^{N}\frac{1}{NC^2_{|N_i^o|}}\sum_{x_m,x_n\in N_i^o}[(x_i-x_{m,n}^i)(x_i-x_{m,n}^i)^T]W\}\end{aligned}\tag{9}$$

Then the problem becomes

$$\max J(W)=\mathrm{tr}[W^T(A-B)W]\tag{10}$$

where

$$A=\sum_{i=1}^{N}\frac{1}{NC^2_{|N_i^e|}}\sum_{x_m,x_n\in N_i^e}[(x_i-x_{m,n}^i)(x_i-x_{m,n}^i)^T]\tag{11}$$

$$B=\sum_{i=1}^{N}\frac{1}{NC^2_{|N_i^o|}}\sum_{x_m,x_n\in N_i^o}[(x_i-x_{m,n}^i)(x_i-x_{m,n}^i)^T]\tag{12}$$

A length constraint $w^Tw=1$ is imposed on the proposed NDNFLA. Then, the optimal projection W of NDNFLA can be obtained by solving the following eigenvalue problem.

$$(A-B)w=\lambda w\tag{13}$$

Let $w_1,w_2,\cdots,w_q$ be the eigenvectors of formula(13) corresponding to the q largest eigenvalues ordered according to $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_q$. An $M\times q$ transformation matrix $W=[w_1,w_2,\cdots,w_q]$ can be obtained to project each sample $M\times 1$ x_i into a feature vector $q\times 1$ y_i as follows:

$$y_i=W^Tx_i,\qquad i=1,2,\cdots,N\tag{14}$$

3 Proposed Algorithm

Most of current existing NFL-based methods are performed to extract the features of images in the time domain or space domain. As a two-dimensional signal, the image also has some specific properties, for example, the directionality, frequency etc. In this paper, a feature extraction method in frequency domain is proposed. This method is based on Discrete Fractional Cosine Transform (DFRCT). DFRCT is a powerful time-frequency tool in the signal processing. It can show how an image is transformed from space domain to frequency domain continuously. In this section, a novel image feature extraction algorithm based on DFRCT is proposed.

Suppose there are c pattern classes. N is the total number of training samples, and N_i is the number of the samples in the ith class. X_i^j denotes the jth sample in the ith class. $\bar{X}_i$ is the mean matrix of training samples in the ith class. $\bar{X}$ is the mean matrix of all training samples.

Firstly, discriminant power criterion based on NFL is proposed in this section. Let l_i^j denote the number of FLs in the same class with x_i^j among its k nearest feature lines. Then, let $L = \sum_{i=1}^{c} \sum_{j=1}^{n_i} l_i^j$. At last, let

$$J_{DP} = \frac{L}{k * N} \tag{15}$$

According to the formula(15), it is clear that the bigger J_{DP} is, the more discriminant features are. So using the discriminant power criterion based on NFL, which order of DFRCT can be found.

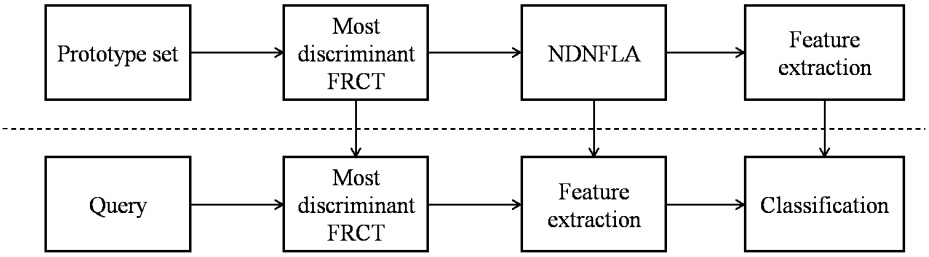


Fig. 2. The procedure of proposed method

The main idea of the proposed feature extraction algorithm, is to extract the local discriminant features from the most discriminant DFRCT. The procedure of the proposed algorithm is given by Fig. 2. The detailed procedure of proposed method is as follows:

Training stage:

Step 1, using the discriminant power criterion based on NFL, find the most discriminant DFRCT;

Step 2, transforming all the prototype samples to the most discriminant DFRCT domain;

Step 3, applying NDNFLA to find the optimal transformation matrix W_0;

Step 4, Extracting the feature of prototype samples following formula(14).

Classification stage:

Step 1, transforming the query to the most discriminant DFRCT domain;

Step 2, Extracting the feature of query following formula(14);

Step 3, Classification with NFL.

4 Experimental Results

4.1 Experiments on ORL Face Database

In this section, a number of experiments on ORL face database[15] and AR face database[16] are implemented to evaluate the effectiveness of the proposed

Fig. 3. Some samples of ORL face database

algorithm, which is also compared with some conventional subspace learning methods, including PCA, as well as the latest NFL-based subspace learning algorithms, such as NFLS, UDNFLA and NDNFLA. The experiments are implemented on a PC with 1.6-GHz CPU and 1G RAM. NFL classifier is used for classification on the features extracted by the NFL-based learning algorithms. To reduce the computation complexity, PCA is used before the NFL-based learning algorithms. All the energy is retained in the PCA phase.

In ORL face database, there are 10 images for each of the 40 human subjects, which were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to $20°$. These images have size of 112×92. In this experiment, 5 images per person from the ORL database are selected randomly for training and the rest are used for testing. The system runs 20 times. Fig.3 shows some samples of ORL face database. Table 1 tabulates the maximum average recognition rate (MARR) of these algorithms on ORL face database. Clearly, MARR of the proposed algorithm is higher than other popular approaches.

Table 1. MARR of different algorithms on ORL face database

Algorithms	MARR	Feature dimension
fisherface	0.9154	39
PCA+NN	0.9271	40
PCA+NFL	0.9449	80
UDNFLA	0.8870	180
NFLS	0.9284	190
NDNFLA	0.9690	150
Proposed algorithm	0.9797	150

4.2 Experiments on AR Face Database

AR face database was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B. It contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different illumination conditions, facial expressions, and occlusions (sun glasses and scarf). The pictures were taken at the CVC under strictly controlled conditions. Each person participated in two sessions, separated by two weeks (14 days) time. The same pictures were taken in both sessions. In the following experiments, only nonoccluded images of AR face database are selected. Five images per person are randomly selected for training and the other images are for testing. This system also runs 20 times. Some samples of AR face database are shown in Fig. 4. Table 2 tabulates the maximum average recognition rate (MARR) of these algorithms on AR face database. Clearly, MARR of the proposed algorithm is higher than other popular approaches.

Fig. 4. Some samples of AR face database

Table 2. MARR of different algorithms on AR face database

Algorithms	MARR	Feature dimension
fisherface	0.9481	120
PCA+NN	0.7604	120
PCA+NFL	0.8521	190
UDNFLA	0.9353	120
NFLS	0.9126	190
NDNFLA	0.9690	150
Proposed algorithm	0.9793	140

5 Conclusion

A novel feature extraction scheme based on NDNFLA and DFRCT is proposed in this paper. NDNFLA can extract the local discriminant feature of the samples. Combining DFRCT, the proposed algorithm can extract the discriminant

information in frequency domain and preserve the neighborhood of samples. Compared with UDNFLA, NDNFLA and so on, the proposed algorithm has the higher recognition accuracy. Experimental results confirm the efficiency of the proposed NDNFLA.

References

1. Zhou, X., Nie, Z., Li, Y.: Statistical Analysis of Human Facial Expressions. Journal of Information Hiding and Multimedia Signal Processing 1, 241–260 (2010)
2. Wu, J., Zhou, Z.H.: Face Recognition with One Training Image Per Person. Pattern Recognition Letters 23, 1711–1719 (2002)
3. Chen, S., Zhang, D., Zhou, Z.H.: Enhanced $(PC)^2A$ for Face Recognition with One Training Image Per Person. Pattern Recognition Letters 25, 1173–1181 (2004)
4. Belhumenur, P.N., Hepanha, J.P., Kriegman, D.J.: Eigenfaces vs Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Analysis Machine Intelligence 19, 711–720 (1997)
5. Li, H., Jiang, T., Zhang, K.: Efficient and Robust Feature Extraction by Maximum Margin Criterion. IEEE Trans. Neural Networks 17, 157–165 (2006)
6. Li, S.Z., Lu, J.: Face Recognition Using the Nearest Feature Line Method. IEEE Trans. Neural Networks 10, 439–443 (1999)
7. Chien, J.T., Wu, C.C.: Discriminant Waveletfaces and Nearest Feature Classifiers for Face Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 1644–1649 (2002)
8. Chen, K., Wu, T.Y., Zhang, H.J.: On the Use of Nearest Feature Line for Speaker Identification. Pattern Recognition Letters 23, 1735–1746 (2002)
9. Gao, Q.B., Wang, Z.Z.: Using Nearest Feature Line and Tunable Nearest Neighbor Methods for Prediction of Protein Subcellular Locations. Computational Biology and Chemistry 29, 388–392 (2005)
10. Zheng, Y.-J., Yang, J.-Y., Yang, J., Wu, X.-J., Jin, Z.: Nearest Neighbour Line Nonparametric Discriminant Analysis for Feature Extraction. Electronics Letters 42, 679–680 (2006)
11. Yang, Y., Yuan, Y., Li, X.: Generalised Nearest Feature Line for Subspace Learning. Electronics Letters 43, 1079–1080 (2007)
12. Lu, J., Tan, Y.P.: Uncorrelated Discriminant Nearest Feature Line Analysis for Face Recognition. IEEE Signal Processing Letter 17, 185–188 (2010)
13. Yan, L., Pan, J.S., Chu, S.C., Roddick, J.F.: Neighborhood Discriminant Nearest Feature Line Analysis for Face Recognition. In: Second International Conference on Innovations in Bio-inspired Computing and Applications (IBICA), pp. 344–347. IEEE Press, Shenzhen (2011)
14. Pei, S.C., Yeh, M.H.: The Discrete Fractional Cosine and Sine Transforms. IEEE Transactions on Signal Processing 49, 1198–1207 (2001)
15. Olivetti, Olivetti and Oracle Research Laboratory Face Database of Faces, http://www.cam-orl.co.uk/facedatabase.html
16. Martinez, A.M., Benavente, R.: The AR Face Database. CVC Technical Report 24 (1998)

Adaptive Scheduling System Guaranteeing Web Page Response Times

Krzysztof Zatwarnicki

Department of Electrical, Control and Computer Engineering
Opole University of Technology, Opole, Poland
k.zatwarnicki@gmail.com

Abstract. The problem of guaranteeing Quality of Web Services (QoWS) is now crucial for farther development and application in new areas of internet services. In this paper we present WEDF (Web Earliest Deadline First) adaptive, an intelligent Web system which guarantees the quality of service in Web systems with one Web server. The proposed system keeps the page response time within established boundaries, in such a way that at a heavy workload, the page response time both for small and complex pages, would not exceed the imposed time limit. We show in experiments conducted with the use of a real modern Web server that the system can be used to guarantee a higher quality of service than other referenced and widely used in practice scheduling systems.

Keywords: Quality of Web services, guaranteeing Web page response time, HTTP request scheduling.

1 Introduction

For a long time the problem of developing Web-based systems guaranteeing and ensuring the quality of service was not in the mainstream of issues connected with the problem of designing web systems, especially when it comes to ensuring the response times for services on the Internet. The problem of designing Web systems guaranteeing QoWS is connected with the fact that most of the services operating nowadays on the Internet offer a quality of service on the best-effort level [9]. Additionally the Internet is managed by a group of independent operators whose strategic goals are different and only few of them provide guaranteed services.

Due to the fact that the Internet becomes progressively one of the main communication channels and many areas of life are transferred to the Web platform or exist only in this platform it is very important to provide not only best-effort services but also to guarantee the quality of service. It can be even said that the problem of guaranteeing quality of service is now crucial for the farther development and application in new areas of internet services. Therefore, at the present time, there is a significant increase in interest, not only in the problem of improving but also of guaranteeing the quality of service.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 273–282, 2012.

In our previous works we have already presented proposition of systems improving quality of service in a locally distributed Web cluster system [3], globally distributed Web cluster system with broker [4] and globally distributed Web cluster system without broker [13]. In our later works we deal with the problem of guaranteeing quality of service in Web systems with one Web server [14], locally distributed cluster of servers [12], and globally distributed Web cluster system with broker [15].

In this paper we present new a version of WEDF adaptive, an intelligent Web system which guarantees the quality of service in Web systems with one Web server. Our system guarantee that the page response time does not exceed a demanded time under the assumptions that the number of requests simultaneously serviced by the system is lower then the maximal capacity of the system. We show the quality of operation of the system in experiments conducted which use of real modern Web server. In the experiments we used a prototype of WEDF scheduling server, and a request generator simulating the behavior of real modern Web browsers. In the paper we describe in detail how the scheduling server and a request generator are constructed and implemented.

There are many works on how to guarantee Web service quality. Most of them concern maintaining the quality of the service for individual HTTP requests [1], [2], [6], [7]. Very few papers were dedicated to the problem of designing Web service, which would guarantee servicing all WWW pages within a limited time. In one of these papers [11] the proposed solution is, however, similar to others, keeping the quality of the Web service at a given level only for a limited group of users. Among many papers devoted to scheduling requests in the Web systems with no admission control, most of them are concentrated on improving the quality e.g. [10]. Guarantying quality of service is almost always connected with rejecting requests of users not belonging to a privileged group. In the system proposed we use an adaptive, intelligent method enabling the service of all of the users with guaranteed quality.

The paper is divided into five sections. Section 2 presents the problem formulation. Section 3 contains a design of a scheduling server controlling the WEDF system. Section 4 describes the testbed used in the experiments and the research results. The final Section 5 contains concluding remarks.

2 Problem Formulation

The WEDF system is composed of clients, a scheduling server and an executor. Fig. 1a presents the overall view of the system. The scheduling server receives HTTP requests sent by clients, schedules them and transfers them to the service responsible for executing the request. The HTTP response is transferred back from the executor to the client through the scheduling server.

The scheduling server should operate in this way to not let the page response time $\dot{t}_{p_i}$ exceed the demanded time $t_{\max}$ under the assumptions that the number of requests simultaneously serviced by the system is lower then the maximal capacity of the system, and it is possible to achieve time $t_{\max}$ for every Web page serviced in the Web system.

The response time t_{p_i} of the Web page p_i is measured from the moment of receiving by the scheduling server the first HTTP request, concerning the given web page, up to receiving an 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 from the same client, concerning the Web page, was processed within the aforementioned time interval.

The executor consists of the Web server and other backend servers like application and database servers involved in servicing HTTP requests. The executor can service many HTTP requests concurrently.

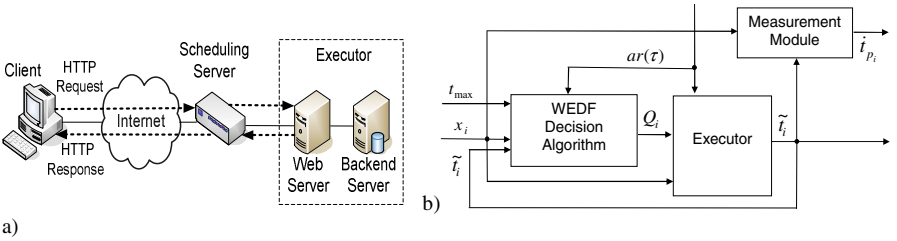


Fig. 1. WEDF system: a) overall view, b) decision process

The scheduling server should operate in this way to not decrease the efficiency and capacity of the Web system. It is also assumed that the scheduling server should allow to service concurrently only an adopted number $ar_{\max}$ of HTTP requests by the executor. The value $ar_{\max}$ should be the smallest number of concurrently serviced requests for which the throughput $X(ar)$ (number of request serviced in the time unit) is roughly maximal.

In order to describe the way the scheduling server operates, the request service time definition should be presented. The request service time $\tilde{t}_i$ is a time measured from the moment the scheduling server starts sending the HTTP request to the executor, to the moment the scheduling server receives the last byte of the HTTP response.

Let us introduce the following designations: x_i – HTTP request, $ar_{\max}$ – maximum number of requests allowed to be serviced by the executor; ar_i – number of requests serviced by executor at the moment of arrival of the i th request; ara_i – number of all requests in the WEDF system at the monet of arrival of the i th request; $ara_{\max}$ –maximum capacity of the system, number of requests in the system, above which request response times increase unacceptably; $ar(\tau)$ – number of requests serviced by the executor at the τ moment; $ara(\tau)$ – number of all requests in the WEDF system at the τ moment; d_i – moment, when the service of the request should end; λ – concurrency factor; M_i – parameters of the executor; p_i – identifier of the Web page to which the i th object belong; W_i – vector of classes of objects not yet downloaded by the client within page p_i, Q_i – queue of HTTP

requests in the scheduling server; $\tilde{t}_i$ – service time to the i th request; $\hat{t}_s$ – estimated service time to request x_i; tp_{p_i} – page response time measured from the moment of arrival of the first object belonging to the page to the moment of arrival of i th request; $\tau_i^{(1)}$ – moment of arrival of i th request; $\tau_i^{(2)}$ – moment the i th request leaves the queue Q_i.

The main task is to design the scheduling server, which for each incoming request x_i, under condition $ar(\tau_i^{(1)}) = ar_{\max}$, designates schedule for the queue Q_i. The queue should schedule incoming requests on the base of deadlines d_i assigned to requests x_i, $i = 1,2,...$, where deadlines d_i are determined in such a way to satisfy the condition $\dot{t}_{p_i} \le t_{\max}$, assuming that $\neg \exists_{\dot{t}_{p_i}} \left[\left(\dot{t}_{p_i} > t_{\max} \right) \wedge \left(\forall_{i=1,2,3...} ar_i = ar_{\max} \right) \right]$ and $\forall_{x_i} \left(ara_i < ara_{\max} \right)$. In the case of $ar_i < ar_{\max}$, request x_i should not be placed in the queue.

Fig. 1b presents general schema of the decision process in the WEDF system. It is worth mentioning here that the problem of the schedule designation for the queue Q_i is trivial, the real problem is to designate the deadlines d_i, $i = 1,2,...$.

3 Scheduling Server

The scheduling server consists of: an analysis module, a queue module and a service module. A schema of the scheduling server is presented in fig. 2.a.

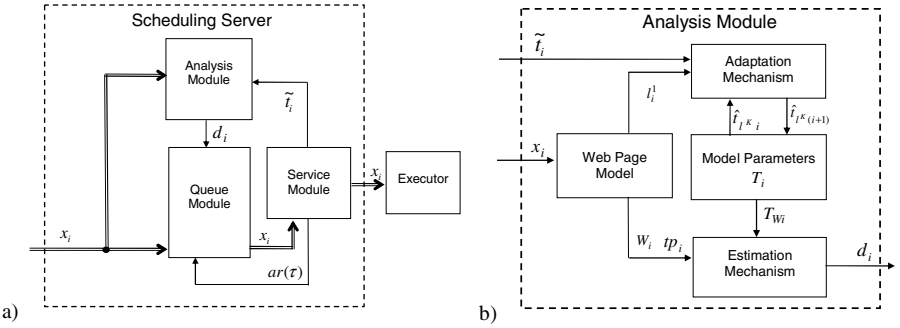


Fig. 2. Scheduling server: a) schema, b) analysis module

The incoming x_i request is recognized and analyzed in the analysis module. This module also determine the deadlines d_i. The deadline is transferred to the queue module which collects requests and schedules them on the base of deadlines. If the number of requests concurrently serviced $ar(\tau)$ by the executor decreases then the HTTP request being in front of the queue is fetched and transferred to the service

module. The service module sends the request to the Web server, supervises its' execution, and sends the response back to the client.

The analysis module is composed of a Web page model, an estimation mechanism, model parameters, and an adaptation mechanism. The Web page model retrieves from the incoming HTTP request x_i, address of requested object u_i, the page identifier p_i, and the user identifier j_i. Both identifiers can be found in the cookie field of the request and were provided to the users' Web browser in the response to the first request. The Web page model also collects information referring to HTTP objects provided in the Web service: sizes of the objects, classes to which they belong, membership of the Web pages and information of objects being downloaded by given users. On the output of the model the following information is provided: vector $W_i = \left[l_i^1,...,l_i^k,...,l_i^K\right]$ of classes of objects belonging to the page p_i and not yet downloaded by the user j_i (l_i^1 is the class of requested object), and the time tp_{p_i}. Objects belonging to the same class have similar request response times. The class of object can be determined on the base of the size of the object in the case of a static object (file stored on the server). Each dynamic object (object created on demand at the moment of the request arrival) can belong to separate class.

The model parameters module T_i, where $T_i = \left[\hat{t}_{1i},...,\hat{t}_{li},...,\hat{t}_{Li}\right]$, stores information concerning request service times for all the classes of requests. The parameter $\hat{t}'_{li}$ is an estimated request service time for an object belonging to l th class and under the load of the executor equal to $ar_{\max}$, where $l = 1,...,L$, and L is the number of distinguished classes. The model parameter module transfers to the estimation mechanism service times $T_{Wi} = \left[\hat{t}_{l^1 i},...,\hat{t}_{l^K i},...,\hat{t}_{L^K i}\right]$ connected with the object pointed out in W_i.

The estimation mechanism determines the deadline d_i according to formula $d_i = \tau_i^{(1)} + \Delta d_i - \hat{t}_{k^1 i}$. The value of Δd_i is a time which can be spent in the queue and on servicing the request by the executor. Δd_i is determined on the base of the following formula $\Delta d_i = \hat{t}_{l^1 i}\left(t_{\max} - tp_i\right) \Big/ \lambda \sum_{k=1}^{K} \hat{t}_{l^k i}$.

The concurrency factor λ determines the number of requests sent by the same user and being serviced simultaneously by the executor. The value of λ should be designated in preliminary experiments. According to our experiments the value of λ is 0.267.

The adaptation mechanism updates times stored in the model parameters module T_i. Adaptation is provided at the end of the service of the request. The new value of service time is calculated on the base of the old value and measured service time $\tilde{t}_i$ according to formula $\hat{t}_{l(i+1)} = \hat{t}_{li} + \hat{\eta}\left(\tilde{t}_i - \hat{t}_{li}\right)$, where $\hat{\eta}$ is the adaptation factor and its value should be equal to 0.1 according to our preliminary experiments. Adaptation is preceded only if the request was placed in advance in the queue Q_i.

The queue module in the scheduling server contains the queue Q_i, and is responsible for placing the request in the queue according to the adopted policy. The request is placed in the queue only if $ar(\tau_i^{(1)}) = ar_{\max}$ otherwise the request is transferred directly to the execution module. At the front of the queue the requests with the shortest deadlines are placed, and at the end with the longest in this way that $Q_i = \left(x_{(i-m_1)},...,x_{(i-m_g)},...,x_{(i-m_G)}\right)$, $Q_i \in D$, $i = 1,2,...$, $\forall_{g\in\{1,...,G\}} m_g \in \{0,...,i-1\}$, where $D = \left\{ \left(x_{(i-m_1)},...,x_{(i-m_g)},...,x_{(i-m_G)}\right): \forall_{d_{(i-m_{(g-f)})},d_{(i-m_g)}, \text{ where } g\in\{1,...,G\}, f\geq 0,\ g-f\in\{1,...,G\}} d_{(i-m_{g-f})} \leq d_{(i-m_g)} \right\}$, $x_{(i-m_g)}$ is the request placed at the g th position in the queue, $d_{(i-m_g)}$ is a deadline assigned to $x_{(i-m_g)}$, and G is the number of requests in the queue.

The request is retrieved from the queue and transferred to the service module at the moment $\tau_i^{(2)}$ when the number of requests serviced concurrently in the executor decreases and $ar(\tau_i^{(2)}) < ar_{\max}$. The service module supervises the service of the request in the executor and measures the service time $\tilde{t}_i$. After finishing the service of the request the service time is transferred to the adaptation mechanism of the analysis module.

4 Testbed and Experiments

The experiments have been conducted with use of a real Web server and simulated Web clients. Three computers and a network switch were used in the experiments. The first computer with an Intel Core 2 Duo 2.0 GHz the processor was acting as a generator of the HTTP requests. The second computer with a similar processor hosted the scheduling server software, and the last computer with an Intel Pentium 4 2.0 GHz processor hosted the Web server Apache 2.2.20 and the MySQL Server 5.1. The network switch was a gigabyte Repotec RP-G3224V switch. The operating system working on each of the computers was the Linux Fedora 15.

The computer chosen for the Web server had the lowest computational power so it was easy to check and reach the maximal capacity of the server without overloading other elements of the system. The Web server hosted five different Web pages. Table 1 presents the structure of the pages.

The pages were static and dynamic. All of the pages contained from 10 to 30 embedded objects of the size from 1 to 100 KB. Dynamic pages used PHP as the script language generating the content of the page. One of the pages used also SQL requests to the MySQL database containing 3 tables of a size of 10 000 rows each.

The scheduling server software was written in the C++ language with the use of *libsoup* [8] and a *boost* [5] libraries supporting the supervision of HTTP requests and queues. The *gcc* compiler was used to create the executable file.

Table 1. Web pages used in the experiments

Name	Type and size of the frame object of the page	Embedded objects number and sizes	MySQL Database
Static 10	Static 1 KB	10, size 1-100 KB, sum 477 KB	—
Static 30	Static 1 KB	30, size 1-100 KB, sum 1.39 MB	—
Dynamic 30	Dynamic, PHP	30, size 1-100 KB, sum 1.39 MB	—
Dynamic MySQL 30	Dynamic, PHP	30, size 1-100 KB, sum 1.39 MB	requests to database containing 3 tables, 10 000 rows each

In order to compare the WEDF scheduling method with other well known and often applied methods the scheduling server had implemented four different scheduling policies:

- Direct – the incoming request was sent to the Web server immediately after its arrival without queuing,
- FIFO (First In Firtst Out) – the requests were queued according to the order of arrival,
- SRPT (Shortest Remaining Processing Time) – the requests were queued according to the estimated request service time (the time was calculated in the same way as in WEDF method),
- WEDF – the requests were scheduled according to WEDF method described above.

The software of the HTTP request generator was also written in the C++ language with use of *LibcURL* library, which enables the creation of HTTP requests and the supervision of the process of sending requests and receiving the responses. The request generator generated requests in a similar way to modern Web browsers. It was creating a given number of virtual clients. Each client, at first, was opening a TCP connection to send the requests concerning the frame of the page, after that it was sending in next the TCP connections requests concerning objects pointed out in the header of the HTML document. After receiving the frame as a whole, the client was opening up to 6 TCP connections to download the embedded objects. Immediately after finishing downloading the Web page as a whole the client started to download the next page.

The software of the generator was also collecting information concerning the quality of the operation of the system working under the control of different methods. The mean value of request response time and the satisfaction were collected. The satisfaction is often used to evaluate the effect of operation of the real time soft systems, it depends on the page response time. Fig. 3 presents the satisfaction in the page response time function. The value of satisfaction is equal to 1 if the page response time is shorter then $t^s_{\max}$ (where $t^s_{\max} = t_{\max}$), and decreases to 0, when its' value is bigger then $t^h_{\max}$.

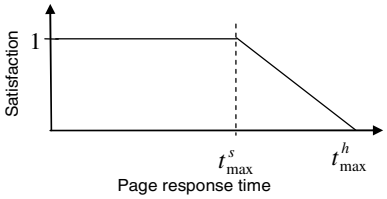


Fig. 3. Satisfaction function

The experiments have been conducted for a different number of simulated clients and different adopted values of $t^s_{\max}$ and $t^h_{\max}$ (the value of $t^h_{\max}$ was $t^h_{\max} = 2t^s_{\max}$).

Fig. 4 presents the mean value of HTTP request response times in the function of load (number of clients), and Fig. 5 presents the mean value of satisfaction in the load function.

As one can notice the request response time (Fig. 5) for the SRPT policy is the shortest. However the satisfaction is the highest for the WEDF policy in each of the experiments, especially in a heavy load and for a short demanded time $t_{\max}$. It should be indicated here that obtaining short request response time does not always increase the quality of service and the users experience in working with the Web service. The obtained results confirm results of previously conducted simulation experiments [14].

In conclusion it can be said that WEDF system can provide the quality of the system at the desired level.

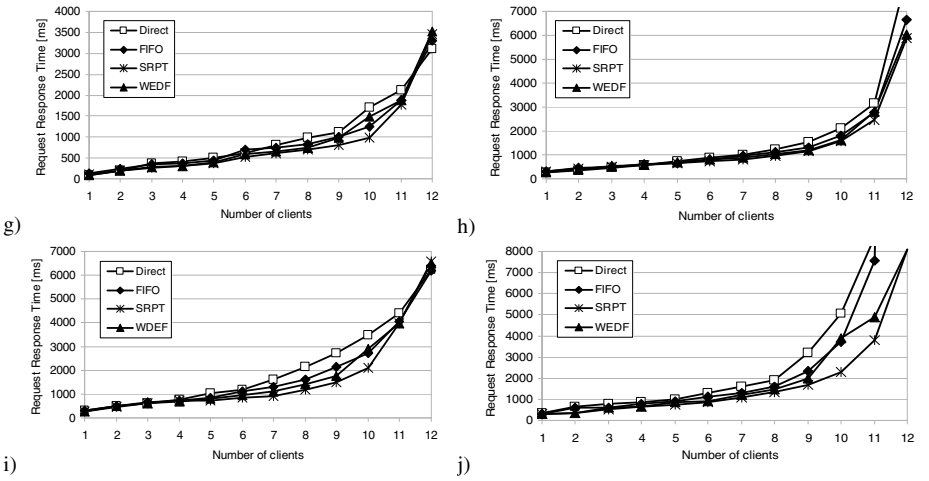


Fig. 4. Mean value of request response time for $t^s_{\max} = 800$ ms and different Web pages: a) Static 10, b) Static 30, c) Dynamic 30, d) Dynamic MySQL 30

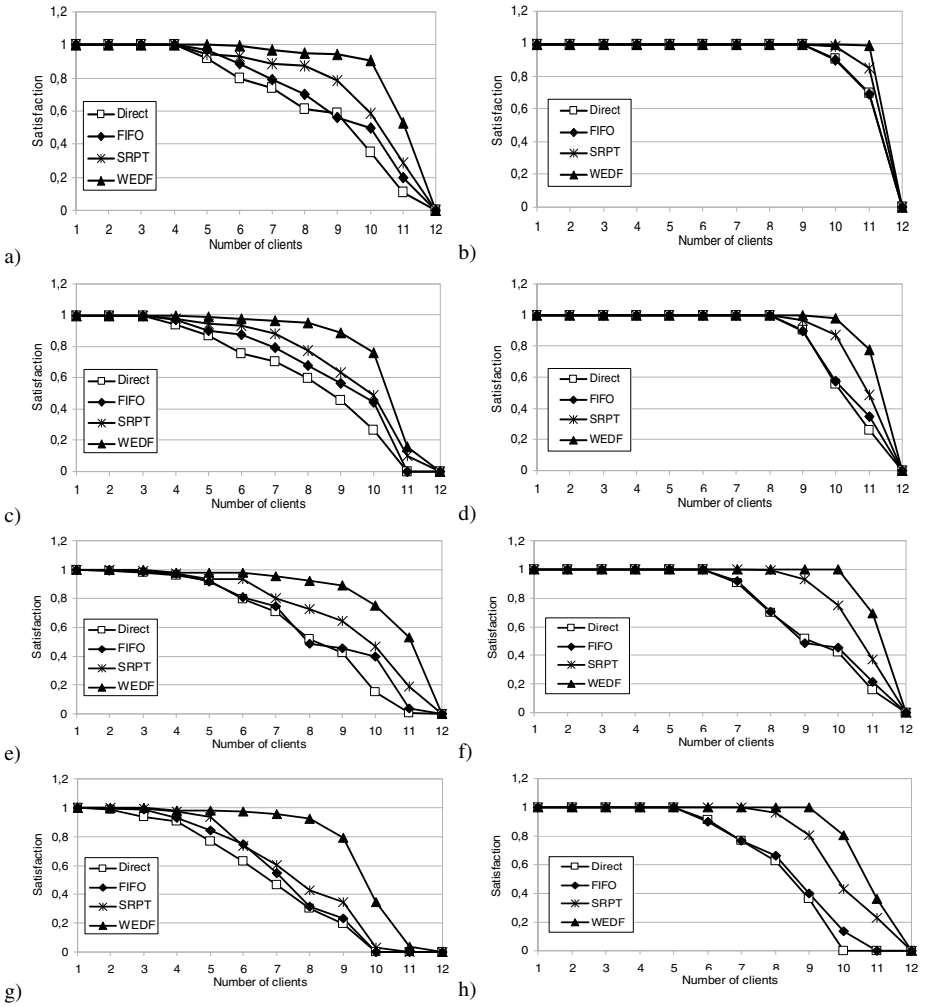


Fig. 5. Mean value of satisfaction in the load function for pages: a) Static 10 $t^s_{\max} = 800$ ms, b) Static 10 $t^s_{\max} = 2000$ ms, c) Static 30 $t^s_{\max} = 800$ ms, d) Static 30 $t^s_{\max} = 2000$ ms, e) Dynamic 30 $t^s_{\max} = 800$ ms, f) Dynamic 30 $t^s_{\max} = 2000$ ms, g) Dynamic MySQL 30 $t^s_{\max} = 800$ ms, h) Dynamic MySQL 30 $t^s_{\max} = 2000$ ms

5 Summary

In the paper the HTTP request scheduling method enabling guaranteeing quality of the Web service was presented. The proposed WEDF method applies adaptive algorithms. According to the method, requests are scheduled at the front of the Web service in the way that the page response time does not exceed a demanded time under the assumptions that the number of requests simultaneously serviced by the

system is lower then the maximal capacity of the system. Thanks to the new method the clients using a loaded service will obtain similar Web page request response times for both small simple pages as well as complex ones requiring retrieving data from databases. The experiments conducted with the use of real modern Web servers show that the method can be used to guarantee a higher quality of service than other reference and widely used in practice scheduling methods.

References

1. Abdelzaher, T.F., Shin, K.G., Bhatti, N.: Performance Guarantees for Web Server End-Systems: A Control-Theoretical Approach. IEEE Trans. Parallel and Distributed Systems 13(1), 80–96 (2002)
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.: Performance Evaluation of Fuzzy-Neural HTTP Request Distribution for Web Clusters. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 192–201. Springer, Heidelberg (2006)
4. Borzemski, L., Zatwarnicka, A., Zatwarnicki, K.: Global Adaptive Request Distribution with Broker. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part II. LNCS (LNAI), vol. 4693, pp. 271–278. Springer, Heidelberg (2007)
5. Boost C++ libraries, `http://www.boost.org/` (access September 10, 2011)
6. Harchol-Balter, M., Schroeder, B., Bansal, N., Agrawal, M.: Size-based scheduling to improve web performance. ACM Trans. Comput. Syst. 21(2), 207–233 (2003)
7. Kamra, A., Misra, V., Nahum, E.: Yaksha: A Self Tubing Controller for Managing the Performance of 3-Tiered Websites. In: Proc. Int'l Workshop Quality of Service, pp. 47–56 (2004)
8. Libsoup library description, `http://developer.gnome.org/libsoup/stable/` (access September 10, 2011)
9. McCabe, D.: Network analysis, architecture, and design. Morgan Kaufmann, Boston (2007)
10. Schroeder, B., Harchol-Balter, M.: Web servers under overload: How scheduling can help. In: 18th International Teletraffic Congress, Berlin, Germany (2003)
11. 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)
12. Zatwarnicki, K.: A cluster-based Web system providing guaranteed service. System Science 35(4), 68–80 (2009)
13. Zatwarnicki, K.: Neuro-Fuzzy Models in Global HTTP Request Distribution. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010, Part I. LNCS, vol. 6421, pp. 1–10. Springer, Heidelberg (2010)
14. Zatwarnicki, K.: Providing Web Service of Established Quality with the Use of HTTP Requests Scheduling Methods. In: Jędrzejowicz, P., Nguyen, N.T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2010, Part I. LNCS (LNAI), vol. 6070, pp. 142–151. Springer, Heidelberg (2010)
15. Zatwarnicki, K.: Guaranteeing Quality of Service in Globally Distributed Web System with Brokers. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 374–384. Springer, Heidelberg (2011)

A Smart and Tangible AR Dress Fitting System

Heien-Kun Chiang, Long-Chyr Chang, Feng-Lan Kuo, and Hui-Chen Huang

Department of Information Management & Graduate Institute of Digital Content Technology
National Changhua University of Education, Changhua, Taiwan
{hkchiang,lcchang,laflkuo}@cc.ncue.edu.tw,
hirney2395@gmail.com

Abstract. The human-computer interaction plays a critical role between the communication of the machine and the human beings. The ability of 3D visualization system to realistically and quickly render the appearance of products or architectures makes it an attractive and affordable solution for product demonstration. Furthermore, the tangible interface of augmented technology, allowing users to interact with products of interest via hand gestures, offers immersive, joyful, and lifelike product interaction experience to users. By integrating the technologies of 3D visualization and augmented reality, this paper proposes a smart, tangible, and gesture-based visualization system for feminine dress-fitting. To evaluate the usability of the proposed system, a field study experimental method is adopted where every participant is asked to experiment with the proposed system first, and fill out a usability questionnaire afterwards. The questionnaire consists of six constructs: effectiveness, ease of use, interactivity, joyfulness, lifelikeness, and satisfaction. The subjects consist of 34 females, with age ranging from 20 to 29 years old. The results of the questionnaire indicate that the subjects show positive attitude toward the proposed system.

Keywords: Tangible interface, augmented reality, dress-fitting system.

1 Introduction

The life quality of people has changed dramatically because of the growth of economy or the change of social structure. Although the cultures might be different from one country to another, the increase of the life pace is an undeniable fact. As indicated by Levine [1], the comparison of cultures of countries seems to always center around "time", the working time and the leisure time of each individual. That is, time management becomes an important factor in deciding the ratio of a person's leisure time to working time. This is especially true for modern working females who routinely need to spend a certain amount of time in dressing themselves up before going out to work. A recent survey of 2,491 women by clothing giant Matalan [2] indicated that females, on average, spend 16 minutes deciding what to wear on weekday morning and 14 minutes on weekend morning. For holidays, America girls spend about 36 minutes to 52 minutes for their leisure activities. If a female has to spend so much time in choosing one of the known clothes from her wardrobe, it is arguable that the time spent on choosing a new cloth while shopping is going to be much longer.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 283–293, 2012.

In the past, a female has to go to a physical department store such as Macy's, Bloomingdale, or Fortnum & Mason to search clothes of interest, try them on, and then decide if the clothes are suitable for her or not. The emergence of Internet and World Wide Web, Web for short, changes the way of buying clothes. Internet and Web enable the buying and selling of products over the network, referred to as electronic commerce or e-commerce for short. Because of the popularity of e-commerce, a special form of e-commerce where consumers buy goods directly from sellers using their computing devices without an intermediary service is called online shopping or online retailing. The process is also called business to consumer (B2C). Currently, there are many online shopping stores offering virtual dressing room allowing shoppers at home to virtually try on dresses or fashions online. This provides a shopper a chance to measure if the style and the fit of the clothes of interest are an ideal match before making a purchase decision. This also helps online shopping stores to reduce the return rate of merchandise due to the style dislike or the size mismatch.

However, there are several shortcomings of current online virtual dressing rooms. Firstly, operations on their systems still rely on the use of mouse and thus a shopper has to periodically move forwards to operate the mouse and move backwards to see the results of the fit. Secondly, most of them only provide 2D front view of the cloth on-trial, totally ignoring the other view angles of the cloth; this might potentially lead a shopper to believe the style and the fit are good but only know they are wrong upon receiving the cloth. Last but not the least, because of the above two factors, the effectiveness, ease of use, interactivity, joyfulness, lifelikeness, and immersive experiences might be greatly reduced.

Fortunately, the current development of technologies of webcam video, augmented reality (AR), and 3D visualization offers a potential solution for problems of current online virtual dressing rooms. The webcam video technology allows a shopper to visually see the real-time live view of the apparel of interest on her/his body. The AR, an extension of virtual reality, technology augments the real world environment with 3D virtual information to provide an immersive and lifelike experience. By integrating the technologies of webcam video, AR and 3D visualization, this study proposes a smart, tangible, and gesture-based visualization system for feminine dress-fitting. That is, the proposed system allows a online shopper to realistically view the cloth on her body in every 30 degree interval of the 360 degree view angles. It also provides a shopper with hand-based gestures to operate the dress-fitting process without having to move her/his feet forward or backward. Lastly, the AR and 3D visualization technologies allow a shopper to see the real-time, virtual, and lifelike 3D cloth superimposed on the image of her body on computer screen.

2 Related Work

AR is a computer simulation technology whose synthesized images are composed of real-life entities and computer-generated virtual artifacts [3]. Before AR, Milgram and Kishino in 1994 proposed mixed reality, the merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects

co-exist and interact in real-time [4]. Figure 1 shows the conceptual diagram of mixed reality which indicates that AR is closer to real environment but still with features of virtual reality. Figure 2 shows an AR example: MagicBook [5]. Azuma [3] considered AR a variation of virtual reality and proposed every AR application should meet the following three conditions: (1) it is a mixture of real entities and virtual objects, (2) it provides real-time interaction, and (3) it must be presented in 3D space. The proposed system meets the conditions defined by Amuza.

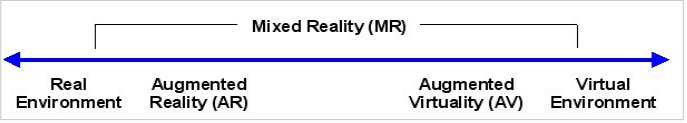


Fig. 1. Conceptual diagram of mixed reality [3]

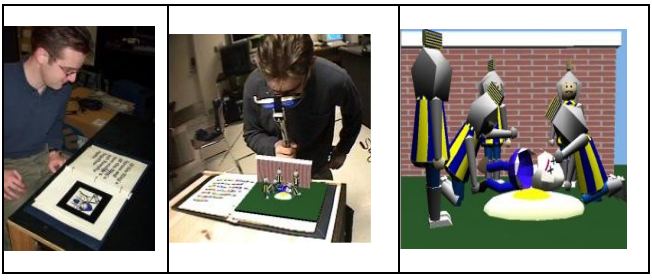


Fig. 2. An example of AR visualization from MagicBook [5]

Recently, AR technology has been applied to many different research fields, including entertainment, industry, medication, sports, and education [6]. For example, famous automobile manufacturer BMW applied augmented technology to its automobile maintenance process where a mechanist, wearing an AR head-mounted display, performed the maintenance of a vehicle by following the voice commands and the instructions on the 3D animation superimposed on real vehicle. The latest PS2 video games "The Eye of Judgment" and "EyePet" from Sony incorporation are two examples of AR entertainment applications [7]. In the field of education, Kaufmann and Schmalstieg applied AR technology to the teaching/learning of high school mathematics and geometry and found the AR system offered a user friendly and interesting learning environment to students [8]. Pan et al. used AR technology to the teaching/learning of primary school students with age of seven to ten and found that carefully designed AR system could satisfy children curiosity and promote their learning motivation [9].

3D representation improves the usability of data and 3D visualization allows the big picture of data to be seen visually. In many applications such as architecture modeling, interior design, and design review [10], 3D visualization are superior to 2D graphics in its ability to render visually realistic, stunning, and richly detailed

information for viewers. Furthermore, 3D models in a visualization system can be used as an intuitive, user-friendly interface to permit users to interact or manipulate the model directly. Nowadays, the 3D rendering speed of a new PC has increased dramatically, and 3D instructions are being embedded into video graphic card's chip-set. Thus, a webcam-based, real-time 3D visualization approach for product demonstration is becoming feasible.

Virtual dressing room is a brand new concept and is slowly becoming a trend on various fashion websites. In 2009, Zuagara [11] claimed itself to be the first company integrating AR technology into a virtual dressing room. To use functions of Zugara's virtual dressing room, a shopper first prints out the designated AR marker cards from the website. She then stands at a certain position in front of her computer screen, and puts a marked card on her upper body to allow the webcam to scan, decode and send the decoded information to the computer to generate virtual cloth on her body, as shown in Figure 3. In 2010, the coalition of Metaio and Hearst Magazines Digital Media [12] announces the first marker-less augmented reality virtual dressing room, as shown in Figure 4, where a shopper is instructed to stand at a certain position to fit the body template provided, and then she can try many different clothes just like the steps used by Zugara's system. Both Zugara and Metaio systems use 2D graphic and a shopper can only see a flat 2D virtual cloth image. In 2010, the department store Macy's [13] launched Magic Fitting Room, as shown in Figure 5, where a customer stands in front of a webcam-equipped large display for her photo to be taken. After that, she uses the touch-panel to choose and resize a cloth of interest.

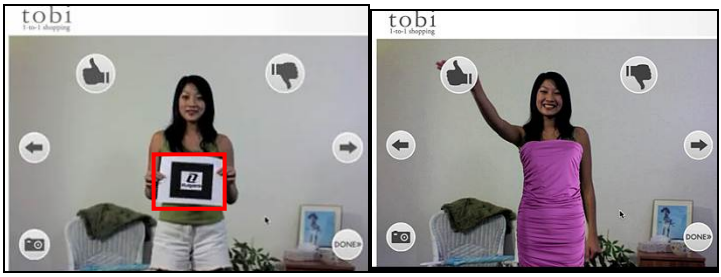


Fig. 3. A shopper uses the Zugara virtual dressing room [11]

Fig. 4. A shopper uses the marker-less virtual dressing room [12]

Fig. 5. A shopper uses the magic fitting room [13]

A comparison of the three virtual dressing rooms discussed above to the proposed system is shown in Table 1. The advantages of the proposed AR virtual dress-fitting system include (1) 3D visualization support, (2) real marker-less recognition, and (3) 360 degree view angles.

Table 1. A comparison of different virtual dressing room systems

System	Model	Technology	Interaction	Interaction
Zugara	2D	AR marker	image-based	A user stands at a fixed position to fit the chosen cloth.
Metaio	2D	AR marker-less	image-based	A user stands at a fixed position to fit the chosen cloth.
Macy	2D	image-based	touch-panel	After the photo is taken, a user can leave her position and uses the touch-panel to control dress-fittings.
This study	3D	AR markerless	image-based and touch-panel	3D cloth fitting model follows with the movement of a user in real-time.

3 System Framework and Experimental Design

Figure 6 shows the system architecture of the proposed 3D AR visualization dress-fitting system which consists of 7 modules: webcam video input module, image recognition module, 3D model generation module, direct manipulation module, touch-screen operation module, the 3D AR visualization user interface module, and dress management module. They are briefly described below.

Webcam video input module: This is usually a webcam which continuously capture the image stream of the surrounding environment, what is sent to the encoder/decoder unit of the image recognition module for analysis. Factors influencing the captured image include camera lens resolution, distance between camera and the object of interest, and the size of object to be recognized. The webcam used in this study is Logitech HD Pro C901 which offers full HD 1080 resolution. The distance between Logitech webcam and the object is about 150cm, determined after several trial-and-errors.

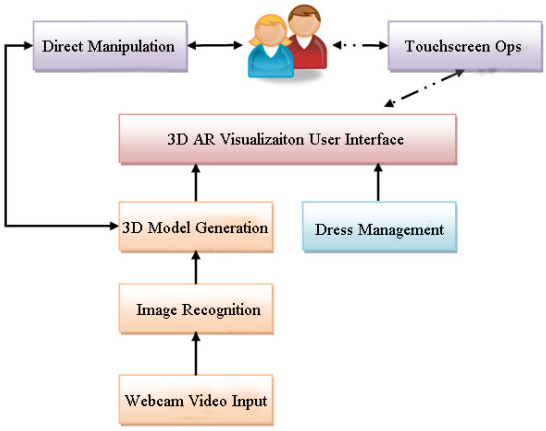


Fig. 6. The system architecture

Image recognition module: This module contains image encoding and image decoding units which are responsible for marker-less objects; encoding and decoding. After receiving the image stream from a webcam, the encoder/decoder unit first analyzes to see if the incoming images are recognizable. Once the target image is identified, the unit searches the system database to retrieve 3D objects representing the identified image (the body shape of a female subject for virtual dress-fitting application). In this study, all marker-less clothes are encoded using D'Fusion Studio AR software from Total Immersion [14]. Figure 7a shows the identified marker-less shape of a user.

3D model generation module: This module is responsible for rendering 3D objects of marker-less objects. Once a 3D object is located, it is rendered using the GLUT library in OpenGL. All 3D objects representing cloth artifacts are designed using 3D Studio Max modeling tool from Autodesk. Figure 7b shows the 3D model (blue T-shirt) generated on a female's body. Once the operations (move or resize) on the 3D model by a user are done, the movement of 3D model follows the user's movement.

Direct manipulation module: This module supports the human hand-gesture operations. A user can use her hand to issue commands to the proposed system. That is, once the system captures the gesture event of a user, it reacts to that event by performing pre-defined operations. Figure 7b shows a user using her hand to issue the move-up command. The most important component of interactive module is event handling which detects events from the system or the user, and reacts to the occurring event correspondingly.

Touch-screen operation module: Touch technology has been used widely at the consumer electronics such as iPhone, iPad, or the interactive weather report large display. In addition to the hand-gesture support, the proposed system uses the touch technology for cloth management because the touch operations on computer screen are intuitive and natural for most users. Figure 7c shows the touch screen dress management interface.

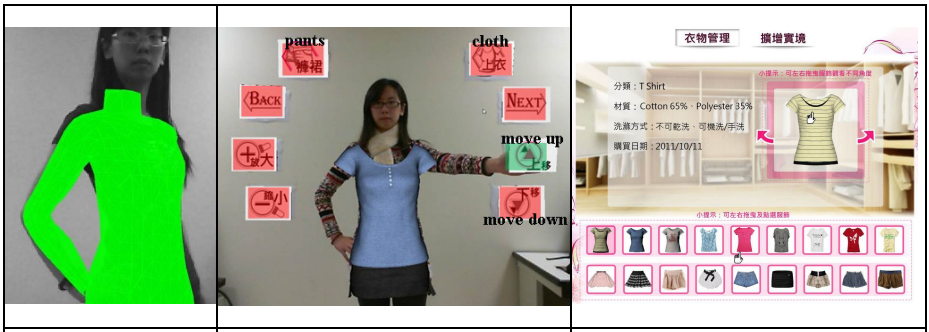
3D AR visualization user interface module: This module is the core of the proposed system which offers a 3D AR visualization environment for a user to try clothes on her body. It displays the 3D model generated after a user's body shape

(marker-less) is identified. It also responds to a user's hand-gesture command to modify the 3D model. Once the size and position of the 3D model on a user's body is set, the 3D model is programmed to perform translation or rotation according to the user's movement. Furthermore, this module coordinates the communication between the touch-screen operation module and the dress management module for retrieving and storing dresses.

Dress management module: This module, as shown in Figure 7c, is implemented using animated Flash for easy operations. All identified marker-less clothes and their corresponding 3D models are stored here. The modified (re-sized) 3D models are stored alongside their original models. That is, the modified 3D models will be retrieved once the same user come back and use the proposed system again.

Implmentation

The system is implemented using D'Fusion AR toolkit for all modules except the dress management module. All programs are written using LuaScript. The dress management module is implemented using Adobe Flash environment and the ActionScript scripting language. The web server for providing the dress interface is implemented using PHP server-side language and the database is implemented using the MySQL database system. All 3D models are built using the 3D Studio Max from Autodesk, exported to be used by D'Fusion.



(a) identified shape (b) hand-gesture operations (c) touch screen operations

Fig. 7. Screen shots of the functions of the proposed system

Experimental Design

The experiment design uses the goal-oriented approach where missions are assigned to participants who have to complete in a certain given time. After the experiment, participants fill out a usability questionnaire, a five-point Likert scale with 1-point indicating strongly disagree and 5-point indicating strongly agree. The following describes the procedures of the experiment.

Subjects: 34 female volunteers with ages between 20 and 29 years old.

Mission: Participants have to finish a pre-defined goal of trying out 5 clothes and 5 skirts/pants. After the initial trial and screening of the clothes and the skirts/pants, they have to find the best combination of cloth/pant (skirt) of their like. The total

experimental time for each participant is 30 minutes. After the experiment, every participant fills out a questionnaire and open questions for 15 to 20 minutes.

Setting: A room is setup for this experiment and it is equipped with a laptop with 15 inch LCD monitor, three webcams, and a set of drawings hang on the wall, as shown in Figure 8. The three webcams are positioned at 30 degree apart. A user is asked to rotate 4 times in a 90 degree clockwise for the system to capture her 360 degree views.

Time: The total testing is 45 to 50 minutes.

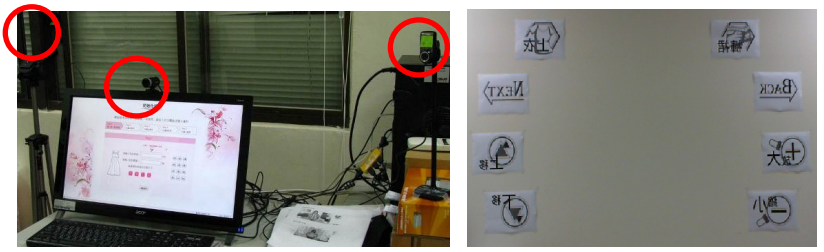


Fig. 8. The experimental setting of the proposed system

4 Usability Study

Subjects of the experiment consist of 34 females, with age ranging from 20 to 29 years old, and they all never use virtual dressing rooms before. Six of them have watched systems similar to the virtual dressing rooms in Internet. Figure 9 shows the statistics for the time spending everyday in dressing up, where 9% of the subjects spend less than 5 minutes, 44% in 5 to 10 minutes, 29% in 10 to 15 minutes, 15% in 15 to 20 minutes, and 3% in 20 to 30 minutes. The average time spending in dressing up for them is about 10 to 15 minutes, which corresponds well to the 14 to 16 minutes survey for 2491 women by Matalan, as mentioned in the introduction section.

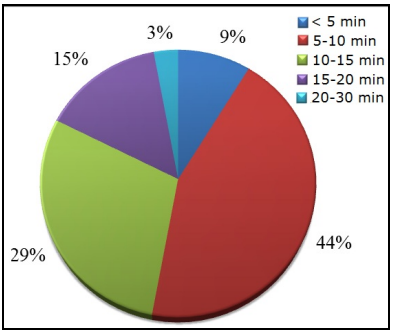


Fig. 9. The distribution of time spent on dressing everyday

Table 2. Questionnaire

Effectiveness construct
1. 360 view of the system fits my need.
*2. 360 view of the system is not necessary.
3. The system improves my efficiency in dressing up.
4. The system speeds me up in finding the right cloth.
5. The system helps me in dressing up through its functions.
Ease of use construct
6. The system is easy to use.
7. The watching and matching the cloth process is smooth and simple.
8. Without reading the manual, I know how to use the system.
*9. I have difficulties in using the system.
10. The user interface of the system is intuitive and natural.
Interactivity construct
11. The system is responsive.
12. The real-time visualization of the 360 degree view of the cloth is dynamic and quick.
*13. The system is slow in response to my operations.
14. The operational process of the system is smooth and easy to understand.
15. The system is able to give adequate and meaningful feedbacks.
Joyfulness construct
*16. The system is boring and not interesting.
17. The artistic design of the user interface is pretty and elegant.
18. Using the system is a wonderful experience.
19. I feel relaxed and pleasant in using the system.
20. I am attracted to and interested in using the system.
Lifelikeness construct
21. I feel lifelike when using the system.
22. The simulation environment provided in the system is vivid and adequate.
23. The 3D cloth model and its texture look like real.
*24. I find it difficult in fitting myself into the scenario of the system.
Satisfaction construct
*25. The system doesn't meet my expectation.
26. I am able to use the system to choose the dress of interest.
27. The overall visual effect and operation of the system satisfies me.
28. I will keep using the system in dressing myself up.
29. I am more than happy in recommending the system to my friends.

* Inversed questions

The questionnaire consists of six constructs: effectiveness, ease of use, interactivity, joyfulness, lifelikeness, and satisfaction. The questions of the questionnaire are shown in Table 2. The questionnaire uses 5-point Liker scale with score of 5 indicating totally agree and score of 1 indicating totally disagree. Results from the questionnaire are shown in Figure 10. The average scores for the constructs of effectiveness, ease of use, interactivity, joyfulness, lifelikeness, and satisfaction are 4.36, 4.25, 4.24, 4.46, 4.22, and 4.42 respectively. All scores are above 4.2 indicating all subjects are satisfied in using the proposed system. The joyfulness construct gains the highest score of 4.46 which might reflects the effort by the authors putting into design an artistic and aesthetics for 3D modeling and user interface is rewarding. The lowest score of 4.22 falls into the lifelikeness construct which is surprising at first but is

understandable after interviewing some of the subjects who state that the 3D models while adequate but still far away from indistinguishable from real artifacts.

Results of the interview with the subjects about the system reveals that (1) the system's 360 degree view angles are the most welcome feature which is not available in the current online virtual dressing rooms; (2) while the textures on the clothes in the system are clear but the response time from the recognition of the cloth to the appearance of its correspondent 3D model is a little bit slow; (3) most subjects feel the system offers an immersive dress-fitting experience which they never encountered before, and (4) some subjects feel the requirement of taking 4 sides of their body in order to generate the 360 degree views takes time and they hope the future system can overcome this shortcoming. All these issues will be addressed in the next undergoing project.

5 Conclusion and Future Research

Nowadays, most online virtual dressing rooms use 2D images and image-based technology to allow shoppers to try the clothes of interest. This kind of presentation is simple and easy to use; however, it usually incurs high chances of mismatch between what a customer feels about the cloth on trial and the quality (shape and style) of the real cloth received afterwards because the 2D presentation of the virtual dressing rooms shows a flat, 2D front view image of the cloth. The recent surging of AR provides a solution for this problem by providing tangible interface and vivid, lifelike 3D visualization of the cloth on trial. The ability of 3D visualization system to realistically and quickly render the appearance of the style and shape of the cloth makes it an attractive and affordable solution for virtual dress-fitting applications.

By integrating the technologies of 3D visualization and augmented reality, this study proposes a smart, tangible, and gesture-based visualization system for feminine dress- fitting. The field study consists of 34 females using the proposed system and the fill out a questionnaire afterwards. From results of the usability questionnaire, consisting of six constructs of effectiveness, ease of use, interactivity, joyfulness, lifelikeness and satisfaction, the proposed 3D AR visualization dress-fitting system proves to be usable and satisfactory.

Limitations of the current study includes the followings: (1) the construction of the 360 degree views takes time which needs to be improved in the near future; (2) the current system only support females, future support for males should be taken into consideration; and (3) the latest Microsoft Kinect device [15] offers an opportunity to build a system capable of automatically scanning and constructing a 3D body mesh. This approach looks promising and might has great potential in the applications of virtual dressing rooms.

References

1. Levine, R.: The Pace of Life in 31 Countries, American Demographics (1997, access from FindArticles.com, May 30, 2010)
2. Matalan, Women Spend One Year of Their Lives on Choosing Clothes, http://geniusbeauty.com/news/women-spend-one-year-choosing-clothes

3. Azuma, R.: A survey of Augmented Reality. Presence: Teleoperators and Virtual Environment 6(4), 355–385 (1997)
4. Milgram, P., Kishino, A.F.: Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information and Systems E77-D(12), 1321–1329 (1994)
5. Billinghurst, M., Kato, H., Poupyrev, I.: The MagicBook: a Transitional AR Interface. Computer & Graphics 25(5), 745–753 (2001)
6. Chiang, H.K., Chou, I.Y., Chang, L.C., Huang, C.Y., Kuo, F.L., Chen, H.W.: An Augmented Reality Learning Space for PC DIY. In: ACM Augmented Human Conference, Tokyo, Japan, March 12-24 (2011)
7. Sony, http://www.sony.com
8. Kaufmann, H., Schmalstieg, D.: Mathematics and Geometry Education with Collaborative Augmented Reality. Computers & Graphics 27, 339–345 (2003)
9. Pan, Z., Cheok, D.A., Yang, H., Zhu, J., Shi, J.: Virtual Reality and Mixed Reality for Virtual Learning Environments. Computers & Graphics 30, 20–28 (2006)
10. GeoSimulation, http://www.geosimulation.org/geosim/3d.html
11. Zugara, http://zugara.com
12. Metaio, http://www.metaio.com
13. Macy, http://www.macys.com
14. Total Immersion, http://www.totalimmesion.net
15. Kinect, http://www.microsoft.com

Consensus as a Tool for RESTful Web Service Identification

Adam Czyszczoń and Aleksander Zgrzywa

Wrocław University of Technology, Division of Computer Science and Management, Institute of Informatics. Wybrzeże Wyspiańskiego 27, 50370 Wrocław, Poland
{adam.czyszczon,aleksander.zgrzywa}@pwr.wroc.pl
http://www.zsi.ii.pwr.wroc.pl/

Abstract. Our recent studies show that the market of RESTful Web services is rapidly growing. However, still there is lack of identification methods of this class of services so they remain invisible for their potential users. In this work we propose a tool for solving above problem using the consensus methods. The identification process is based on recognising URI structural patterns subjected to opinions of different agents with different knowledge. The task of consensus method is to determine version of knowledge which best reflects given versions reflected in agent's opinions on individual components. The research includes defining the structure of knowledge, determining conflict situations, conflict profiles, defining consensus function and assigning distance functions which allow to resolve conflicting views. Moreover, our research is supported with implementations of proposed approach by which we conducted preliminary experiments. Experimental results show high effectiveness and performance of proposed approach in contrast to the other chosen methods.

Keywords: consensus methods, Web services, REST, identification.

1 Introduction

Our recent studies [1] investigating the market of RESTful Web services indicate that this business activity is rapidly growing and by the end of 2012 is projected to reach at least 5,8 billions of dollars. Given such a dynamic market increase, by the end of 2014 it may exceed the size of SOAP services market. The number of RESTful services already surpassed the number of SOAP services. We consider it to be between 22,5 thousand up to 136,3 million. Despite this fact, Web services are meaningful only if their potential users can find sufficient information which allows to use those services. This means that as long as they remain unidentified in private and isolated directories of service providers, their potential users (programmers, web developers, regular Internet users) cannot invoke them. Moreover, currently there is lack of methods of RESTful Web services identification.

In this paper we present a tool for RESTful Web service identification using consensus methods. The process is based on recognising service's URI (Uniform

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 294–303, 2012.

Resource Identifier) structural components. Because of the fact that the structure of RESTful Web service is often ambiguous and difficult to evaluate, the approach utilizes Artificial Neural Networks (ANN) as separate entities in form of autonomous agents. By using different training vectors and training parameters for pattern learning, every agent in the system has different knowledge regarding service's structural components. The task of consensus method is to determine version of knowledge which best reflects given versions reflected in agent's opinions on individual components. The problem is similar to Group Decision Making (GDM) dilemma presented in [2] where set of experts $E = \{e_1, \ldots, e_n\}$ chooses best alternative from set $X = \{x_1, \ldots, x_m\}$. However our approach differs in many general aspects, especially in the consensus reaching process.

Research presented in this paper includes determining the structure of knowledge in the system, defining possible conflict situations resulting from contradictory opinions, defining conflict profiles on conflict subjects, determining consensus function and defining distance functions which allow to resolve conflicting views. Additionally, our research is supported with implementations of proposed approach by which we conducted preliminary experiments. The aim of experiment is to compare effectiveness and performance of different distance functions.

2 Related Work

The concept of RESTful Web service identification is founded on our previous research [3] where we presented uniform and universal RESTful Web service URI structure which allows their identification. Our study also included analysis of resources and their variables in order to create a generic description of particular Web service. Additionally, we proposed engine architecture that allows to effectively identify the services according to their URI structure. Elaborated tools are the basis for service identification and this paper is considered as a continuation to the previously conveyed research by increasing accuracy of the identification process.

Consensus methods for conflict solving are presented in research carried out by N. T. Nguyen [4,5] where author introduces intelligent technologies for resolution of knowledge inconsistency in various computer systems applications, and presents fundamental elements of a consensus system. Research conducted by mentioned author is considered in this paper as a theoretical base for resolving conflicts using consensus system approach. It includes the definition of consensus system, conflict situations and distance functions for consensus determination.

3 RESTful Web Service Identification

The problem of RESTful Web services identification lies in determining if given URI belongs to this class of services or not. The evaluation is based on recognising URI structural patterns of such a service as input parameters for artificial neural network classifier. The RESTful Web service URI structure concern information about service's version, name, access policy (whether service is for

public or *private* use) and service's hostname prefix indicating whether it is part of some API (Application Programming Interface), for example *api.twitter.com* or *ws.audioscrobbler.com*. It also contains information about the location of service's name and resources. By location we mean position in the URI—whether it is its *path* part, *query* or both. The last option concerns resources only. Based on our previous work [3] we define RESTful Web service URI structural components as follows:

Definition 1. *RESTful Web service URI identifier is five-tuple of components* $RWS = \langle$ *ApiIndicator, AccessPolicy, Version, NamePosition, ResourcesPosition* $\rangle$ *where each of them denotes respectively corresponding elements described above.*

Since considered components are not always meaningful, there is certain difficulty in determining which part of the URI represents name, resources or version. The solution to this problem is to differentiate the knowledge of agents by using different input data during network training. Template data can be compiled on the basis of different styles of creating RESTful services from different vendors. During network training, some of the parameters can also change in order to keep possible lowest output error. As a result, agents have different knowledge on how REST complaint Web service should be constructed and generate different output opinions on how particular components conform to its structure. Resolving agent's conflicting opinions on the same subject is the task of our consensus system.

4 Consensus System

According to [4] consensus system describes part of a real-world which is characterized by set of events. Each event is described by set of attributes and their values. Events can be classified into groups based on different subjects. The events are investigated by agents whose job is to make opinions on different subjects of the system. Since agents may speak differently about the same subject, it is necessary to establish a consensus for such a situation. The consensus system is defined as following quadruple [5,6]:

$$ConsensusSystem = \langle A, X, P, Z \rangle \tag{1}$$

where
A – a finite set of attributes, which includes a special attribute *Agent*; each attribute $a \in A$ has a domain V_a (a finite set of elementary values) such that values of a are subsets of V_a; values of attribute *Agent* are 1-element sets, which identify the agents.
X – a finite set of consensus carriers, $X = \{\prod (V_a) : a \in A\}$.
P – a finite set of relations on carriers from X, each relation is of some type T (for $T \subseteq A$ and $Agent \in T$).
Z – a finite set of propositional calculus, for which the model is relation system (X,P)

For RESTful Web services identification purposes and regarding Equation 1, set X defines all objects occurring in consensus system—in our case it represents URI structural elements of a RESTful Web service, described in previous section. Set P represents properties of objects from X and their relations. Those objects are events describing our system denoting actual identification problems analysed by the agents. Set Z represents conditions which have to be satisfied by relations from P.

In order to solve consensus problem for RESTful Web service identification we decomposed the whole task into subproblems presented in the following sections.

4.1 Knowledge Structure

Agents knowledge about service is composed of attributes and their values, and relations and conditions on those attributes, whereas attributes represent service's structural elements presented in Definition 1. Each agent can make an opinion on many services.

Attributes and Values. Opinions made by agents indicate how much particular URI reflects RESTful Web service by determining the membership level of its components. Based on the above the possible attributes of our consensus system are following:

$$A = \{Agent, Service, Api, Access, Version, NamePos, ResourcesPos\} \quad (2)$$

Attribute *Agent* represents agents taking part in voting, *Service* represents all potential services subjected to agents' opinions. The *Api* attribute indicates if service has *ApiIndicator* or not. Similarly *Access* and *Version* represent boolean factors indicating whether particular parameter exists or not. Last two attributes *NamePos* and *ResourcePos* represent possible positions (*path* or *query*) of service's name and resources, or hold information that those parameters do not occur. Based on the above equation the consensus carriers X and resulting values of above attributes are as follows:

$$X = \left\{\prod(V_{Agent}), \prod(V_{Service}), \prod(V_{Api}), \ldots, \prod(V_{ResourcesPos})\right\} \quad (3)$$

where
$V_{Agent} = \{a_1, a_2, a_3, \ldots, a_n\}$, $V_{Service} = \{s_1, s_2, s_3, \ldots, s_n\}$,
$V_{Api}, V_{Access}, V_{Version} = \{0, 1\}$, $V_{NamePos}, V_{ResourcesPos} = \{0, path, query\}$.

Relations and Conditions. We distinguish three fundamental relations representing three types of knowledge about service URI structure. Those relations concern information about service (*ApiIndicator*, *AccessPolicy*, *Version*), and position of service's name and resources. Above relations are defined in equation below.

$$P = \{ServiceInfo, ServiceName, ServiceResources\} \quad (4)$$

where above relations are of following types:

$ServiceInfo : \{Agent, Service, Api, Access, Version\}$,
$ServiceName : \{Agent, Service, NamePos\}$,
$ServiceResources : \{Agent, Service, ResourcesPos\}$.

For example we interpret a tuple $\langle\{a_1\}, \{s_1\}, \{1\}, \{1\}, \{0\}\rangle$ of relation *ServiceInfo* in following way: agent a_1 considers potential service s_1 to contain information about *ApiIndicator*, *AccessPolicy* but not about *Version*. Example tuple $\langle\{a_1\}, \{s_1\}, \{path\}\rangle$ of relation *ServiceName* means that agent a_1 considers s_1 to contain service's name in path. Tuple $\langle\{a_1\}, \{s_1\}, \{query\}\rangle$ of last relation *ServiceResources* means that a_1 states that s_1 has resources in query.

However, in order to make identification process reasonable, agents knowledge must rely on certain rules. Therefore, relations described above have to satisfy the following conditions:

$$\begin{aligned} Z = \{ & \\ & (ServiceInfo(a, s, api, acc, v) \wedge (api = 1 \veebar api = 0)), \\ & (ServiceInfo(a, s, api, acc, v) \wedge (acc = 1 \veebar acc = 0)), \\ & (ServiceInfo(a, s, api, acc, v) \wedge (v = 1 \veebar v = 0)), \\ & (ServiceName(a, s, n) \wedge (n = 0 \veebar n = path \veebar n = query)), \\ & (ServiceResources(a, s, r) \wedge (r = path \vee r = query)) \Rightarrow \\ & \quad \Rightarrow (ServiceResources(a, s, r) \wedge r \neq 0) \\ & \} \end{aligned} \tag{5}$$

In general above conditions constraint agents on choosing only one value from set V of corresponding attribute. For the first three conditions this means that according to agent a if information about service s includes api information, access indicator or version, the same agent cannot say that this service does not contain one of those attributes. For the fourth condition this means that if service name exists and it is in path it cannot be in query, and vice versa. In result one agent may make opinion on one service only. Fifth condition states that agent cannot say that service resources do not exist while they are in path or in query. It also should be noted that it is possible for an agent to conclude that there are resources both in query and in path. Such a situation means that resources are considered as *mixed*.

4.2 Conflict Situations

Conflict situation is formed on the basis of different opinions on the same subject specified by different agents. It consist of: (i) relations which describe type of conflict situation, (ii) subjects of conflict situation on which agents make their opinions, (iii) and content of the conflict which comprises agents information on given subject. Opinion of a single agent on the conflict subject is represented by a single tuple. Based on the above assumptions and regarding to [5] we define conflict situation c as:

$$c = \langle p, A \to B\rangle \tag{6}$$

where

$p \in P$ represents relation, set A represents conflict subjects, set B the content of the conflict.

RESTful Web service identification is based on resolving conflict situations which concern service components described by relations in P. Therefore, the conflict subject are services on which agents make their opinions. The content of the conflicts are opinions on components of corresponding relation type. The information they carry allows to establish a consensus for given conflict. Based on the above for every relation P we define the following conflict situations:

$$c_1 = \langle ServiceInfo, Service \rightarrow \{Api, Access, Version\}\rangle \tag{7}$$

$$c_2 = \langle ServiceName, Service \rightarrow \{NamePos\}\rangle \tag{8}$$

$$c_3 = \langle ServiceResources, Service \rightarrow \{ResourcesPos\}\rangle \tag{9}$$

Resulting conflict situations apply to information about service, location of its name and location of its resources. In the tables below we present examples of defined above conflict situations on two different services s_1, s_2, created by three different agents $a_{1..3}$ (for better readability we skip curly brackets when there is only one value in the set). The conflict situation presented in Table 1 shows

Table 1. Example of conflict situation c_1

Agent	Service	Api	Access	Version
a_1	s_1	1	1	1
a_1	s_2	0	0	1
a_2	s_1	0	1	1
a_2	s_2	0	1	0
a_3	$\{s_1, s_2\}$	0	1	1

agent opinions on information about services contained in their URI structure. Agents a_2 and a_3 consider service s_1 to contain information about *Access* and *Version* only, whereas agent a_1 believes that the same service contains all three parameters. In case of service s_2 there are three conflicting views. Agent a_1 considers this service to include information about version only, a_2 to include information about access policy only, and a_3 to include information about both parameters. Conflict situation in Table 2 presents opinions about service's name

Table 2. Example of conflict situation c_2

Agent	Service	NamePos
a_1	s_1	*path*
a_1	s_2	0
a_2	$\{s_1, s_2\}$	*query*
a_3	s_1	*path*
a_3	s_2	0

location in URI structure. Contradictory opinions relate to agent a_2 that in contrast to other agents believes that names of s_1 and s_2 are in *query*, instead of *path* as in the case of remaining opinions on s_1, and that name does not occur at all as in the case of s_2. Conflict situation in Table 3 presents contradictory

Table 3. Example of conflict situation c_3.

Agent	Service	ResourcesPos
a_1	s_1	$path$
a_1	s_2	$\{path, query\}$
a_2	s_1	$path$
a_2	s_2	0
a_3	$\{s_1, s_2\}$	$\{path, query\}$

opinions concerning service's resources position. In case of s_1 all agents consider it to contain resources in *path*. Additionally, agent a_3 states that those resources are of *mixed* type. As for s_2, agents a_1 and a_3 consider its resources also to be both in *path* and *query*, whereas sceptical agent a_2 believes that the same service has no resources.

4.3 Conflict Profiles

In order to determine consensus from given conflict situations we firstly need to specify conflict profiles for every situation. Conflict profile is represented as a set of different versions of knowledge about the same element. It is defined on subjects of conflict situations—in our case services. For each conflict subject $e \in Service$ we determine conflict profiles $profile(e)$ which contain agents' opinions about given subject. The definition of conflict profile is following [5]:

$$profile(e) = \left\{ r_{B \cup \{Agent\}} : (r \in P) \wedge (e \prec r_A) \right\} \tag{10}$$

Referring to examples of conflict situations presented in Table 1, 2 and 3, the resulting conflict profiles are presented in Table 4. It presents two conflict profiles, one per each service extracted from conflict situation c_1 which is connected with $ServiceInfo$ relation. Table clearly illustrates differences in the opinions of individual agents. Each profile represents knowledge of all agents on individual service with respect to a given conflict situation. With such a knowledge representations it is possible to determine consensus.

Table 4. Example of conflict profiles for conflict situation c_1, c_2, c_3.

Service	Agent	Api	Access	Version	NamePos	ResourcesPos
s_1	a_1	1	1	1	$path$	$path$
s_1	a_2	0	1	1	$query$	$path$
s_1	a_3	0	1	1	$path$	$\{path, query\}$
s_2	a_1	0	0	1	0	$\{path, query\}$
s_2	a_2	0	1	0	$query$	0
s_2	a_3	0	1	1	0	$\{path, query\}$

4.4 Consensus and Distance Function

Referring to [5] the consensus of $profile(e)$ on subject $e \in Service$ for situation $c = \langle p, A \to B \rangle$ is represented by tuple $C(c, e)$ of type $A \cup B$, which satisfies the logical formulas from set Z. Based on the above the consensus definition of situation c is following:

$$C(c) = \{C(c, e) : e \in Service\} \tag{11}$$

According to Equation 11 the consensus of situation c is a set of consensuses of its conflict profiles. In order to establish consensus $C(c)$ distance functions are used. Those functions allow to calculate the distance between the values of attributes. To do so, for every conflict situation we need to create resulting from it conflict profile. Distance function is used for measuring the distance between value sets V_a of attributes V for each pair of sets of individual profile. Each profile $profile(e)$ is formed on the basis of conflict subjects e of conflict situation c. Referring to [4], for predefined situations c_1, c_2, c_3 we can use the following distance function:

$$\rho(X, Y) = \frac{1}{2card(V_a) - 1} \sum_{z \in V_a} Part(X, Y, z) \tag{12}$$

where
$Part(X, Y, z) = 1$ for every $z \in X \div Y$, $Part(X, Y, z) = 0$ for every $z \in X \cap Y$, $Part(X, Y, z) = 0$ for every $z \in V_a \setminus (X \cup Y)$

For above formula function $card(X)$ returns the cardinality of set X. The function described in Equation 12 reflects element shares in the distance. This kind of function is based on determining the value of shares of each element of the set V_a in the distance between two subsets of this set [4]. This function seems to be natural for finding consensus which best reflects agents opinions. However, its has some drawbacks. It assumes that the cardinality of V_a is known, which may be difficult to obtain in continuous systems where this variable depends on system's state. Another drawback is its high complexity for large attribute sets. Therefore, as an alternative we propose using function minimizing transformation cost [4]:

$$\delta_a(X, Y) = \frac{T(X, Y)}{\sum_{x \in V_a} E_{AR}(x)} \tag{13}$$

where
$T(X, Y)$ represents the minimal cost needed for transforming set X into set Y.
E_{AR} represents the cost of adding (or removing) an elementary value to (or from) a set.

Equation 13 relies on determining the minimal cost of transformation of one set to another. By transformation we mean adding, removing or altering an element from set X into set Y. The cost of transformation T is equal to the minimal normalized number of elements which need to be moved from one set to another set. The adding/removing cost depends on the system and it is usually based on a cost matrix.

5 Experimental Results

Based on the approach presented in this paper we implemented consensus system by which we conducted preliminary experiments. The system allowed to identify RESTful Web services based on different agent's knowledge. The aim of the experiment was to measure the effectiveness and performance of proposed approach using three different distance functions. Additionally, we contrasted our approach to *k-means* and *naive bayes* classifiers.

In order to identify services we collected 16 939 URIs using our implementation of Web Service Crawler. The URIs were extracted from four well known RESTful Web service providers: *delicious.com*, *Google*, *last.fm*, *Yahoo*. Among the URIs collection we inserted 755 of them as RESTful Web services. In experiment we used four different agents—one per each service provider. Agents' knowledge depended on training set of ANN, where there was one provider's data set per one agent. In result, every agent was an expert in identifying services of its corresponding provider. In order to evaluate the effectiveness we used classical measures in information retrieval which are: precision, recall and F-measure. The distance functions we used are presented in equations 12 and 13. For the second function we assumed that the cost of adding/removing one element is equal to 1. Additionally, we used third distance function—Jaccard index—as a simple alternative and comparison reference point. In order to measure performance we calculated average execution time of each method, where total average time of the experiment (of all five identification methods) was 7.22s. Preliminary experimental results concerning effectiveness are presented on Fig. 1.

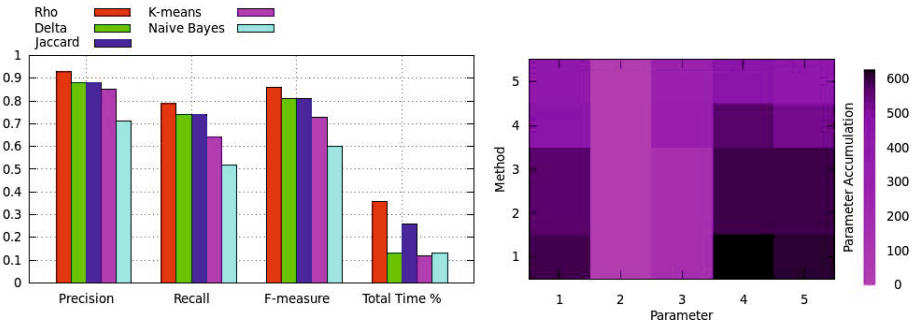


Fig. 1. Effectiveness and performance (on the left), and services parameter accumulation (on the right) of RESTful Web service identification using five methods.

The average execution time of service identification process using presented five methods was: 2.58s, 0.17s, 0.35s, 0.88s and 0.91s respectively. Total effectivenes (F-measure) was equal to: 0.86, 0.81, 0.81, 0.73 and 0.60 respectively. The results of methods utilising δ and *Jaccard* functions were identical and similar to the first method ρ. On the right side of Figure 1 we presented heat graph of accumulation (+1 if parameter exists in identified service) of service's URI structural

parameters (Definition 1) of all five methods. As one can see, in consensus approach almost every identified service contained parameters 1, 4, 5 which is true for given collection of RESTful URIs. In case of k-means and bayes classifiers the parameter clusterisation was smoother which resulted in lower efectiveness, as presented on the left side of the Figure 1.

6 Conclusions and Future Work

Experimental results clearly show that despite the fact that identification using function reflecting element shares in distance is more effective, its time complexity is very high. Moreover, by taking into account the ratio between performance and effectiveness, the transformation cost function seem to be the best. Its performance was slightly lower that k-menas and naive bayes but effectiveness was much better. On the other hand, it must be noted that true effectiveness of presented approach depends mainly on the complexity of ANN training vectors because it reflects agent's knowledge. Despite this fact, presented consensus system, including defined knowledge structure, conflict situations, profiles and distance functions, allows to effectively identify RESTful Web services and gives the possibility to easily extend the system by adding more attributes, relations and conditions, and allows to decide which distance function to use in order to receive best results. Presented approach also provides a promising alternative to traditional classifiers such as k-means and naive bayes.

For future work the binary values of knowledge attributes could be replaced by numerical ranges $[0, 1]$ obtained by agents using different fuzzy logic membership functions. This would highly improve the effectiveness of presented approach.

References

1. Czyszczoń, A.: Analiza rynku usług internetowych. In: Interdyscyplinarność Badań Naukowych, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, pp. 199–204 (2012) (in Polish)
2. Alonso, S., Herrera-Viedma, E., Chiclana, F., Herrera, F.: A web based consensus support system for group decision making problems and incomplete preferences. Inf. Sci. 180(23), 4477–4495 (2010)
3. Czyszczoń, A., Zgrzywa, A.: An artificial neural network approach to RESTful Web services identification. In: Information Systems Architecture and Technology. Service Oriented Networked Systems, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, pp. 175–184 (2011)
4. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management. Advanced Information and Knowledge Processing. Springer-Verlag New York, Inc., Secaucus (2008)
5. Nguyen, N.T.: Consensus system for solving conflicts in distributed systems. Information Sciences 147(1-4), 91–122 (2002)
6. Śliwko, L., Nguyen, N.T.: Using multi-agent systems and consensus methods for information retrieval in internet. Int. J. Intell. Inf. Database Syst. 1(2), 181–198 (2007)

Detection of Tennis Court Lines for Sport Video Categorization

Kazimierz Choroś

Institute of Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
kazimierz.choros@pwr.wroc.pl

Abstract. Digital video data are stored and offered by many public Web services such as Internet video collections, TV shows archives, Internet video-on-demand systems, personal video archives, and others. New methods and technologies of video indexing and retrieval in the Web are developed. Content-based indexing of TV sports news is based on the automatic segmentation, then recognition and classification of scenes reporting the sport events in a given discipline. Automatic classification of sports in TV sports news is one of the basic process in video indexing. There are many different strategies how to recognize a sport discipline. It may be achieved by player scenes analyses leading to the detection of playing fields, of superimposed text like player or team names, recognition of player faces or sport objects, detection of player and audience emotions, and also to the detection of lines typical for a given playing fields and for a given sport discipline. The paper proposes a framework of the automatic line detection of a tennis court for the selection of tennis shots from TV sports news. Categorization of sport videos is based on the minimum set of lines for the detection of a tennis court. The framework has been verified and tested in the Automatic Video Indexer AVI.

Keywords: content-based video indexing, video Web services, TV sport news analyses, sport video categorization, player scenes analysis, line detection, tennis court lines, AVI Indexer.

1 Introduction

The huge number of videos is currently available in the Web as well as in the local networks and systems. The most popular video websites observe tens of millions of unique visitors every month. For effective retrieval of video data not only standard text indexing and retrieval procedures should be used but also more and more sophisticated content-based video indexing and retrieval methods. An automatic processing of television broadcast is one of the most frequent application of content-based video indexing. TV news and especially TV sports news is one of the most viewed video content on the Web. The main goal of research and experiments with sport videos is to propose and develop automatic methods such as automatic detection or generation of highlights, video summarization and content annotation, player detection and tracking, action recognition, ball detection, kick detection such as

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 304–314, 2012.

penalty, free, and corner kick, replay detection, player number localization and recognition, text detection and recognition for player and game identification, detection of advertisement billboards and banners, authentic emotion detection of audience, as well as automatic sports classification in TV sports news. Because of a huge commercial appeal sports videos became a dominant application area for video automatic indexing and retrieval.

The main purpose of sport video processing is to categorize the sport shots for example in TV sports news. Due to the automatic categorization of sport events videos can be automatically indexed for content-based retrieval. So, the retrieval of news presenting the best or actual games, tournaments, matches, contests, races, cups, etc. or special player behaviours or actions like penalties, jumps, or race finishes, etc. in a desirable sports disciplines becomes more effective.

The analysis process of the recognition of sports disciplines is usually performed for all frames of TV sports news what is very time-consuming. For shot categorization the best highlights of sport events seem to be the most adequate for automatic categorisation of sport events. If in a single frame we detect a tennis game based on the detection of a tennis racket, tennis ball, tennis player, or tennis court lines the whole video shot or even whole scene can be classified as tennis. In almost every tennis news broadcast a wide plan of a tennis court is included. The analyses of newscasts concerning given sport disciplines have shown that many sport videos such as archery, diving, soccer, and tennis have repetitive structure patterns. A strong majority of tennis highlights in TV sport news are of a standard structure of six or seven shots: first player, second player, tennis court, serve, return ball or balls, and zoom presenting two players shaking hands over a tennis net.

The effective method of the detection of tennis games in TV sports news must not be based on the recognition of all frames belonging to a tennis shot or the more to a tennis scene – a group of tennis shots presenting a given game. If one of the frames is recognized with a great probability as the frame with tennis court the whole shot/scene can be classified as tennis shot/scene. So, for the categorization of sport shots or scenes the best frames are those belonging to a shot with a wide view of a court. From the point of view of the sport video classification the requirements in detection algorithms may therefore be relatively high because not all lines are needed to recognize what playing field is. Therefore, the minimum set of lines should be defined for every sport playing field.

The paper is organized as follows. The next section describes the main related works in the area of automatic indexing of sport videos, of line detection in playfields, and of tennis video scene categorization. Strategies in sport categorization will be discussed in third section: colour histogram comparison, text detection, sport object, face, player and audience emotions detection and analysis. The tennis court lines and their digital representations in digital videos will be outlined in the forth section. The fifth section presents the framework of tennis video shot detection in TV sports news. In the sixth section the experimental results of the categorization of sport video shot in headlines of TV sports news based on line detection in a playfield obtained in the AVI Indexer are reported. The final conclusions and the future research work areas are discussed in the last 7th section.

2 Related Works

There are many investigations carried out in the area of automatic recognition of video content and of visual information indexing and retrieval [1-4]. To retrieve efficiently videos stored in more and more huge multimedia databases in the Web we need new methods oriented to visual data, methods much more effective than traditional textual techniques applied for videos.

An automatic semantic categorization of sport video shots mainly of shots from TV sports news has become one of the most popular content-based video analysis because of a very high popularity of sport games in TV broadcasts, a huge amount of broadcast sport videos generated every day, and the large share of sport video materials in multimedia databases. Then, a great commercial appeal for sport video automatic indexing and retrieval systems is observed.

A unified framework for semantic shot classification in sports videos has been proposed in [5]. The proposed method has been tested over 3 types of sports videos: tennis, basketball, and soccer. Another approach and another kind of analyses of sports news were implemented in a system [6] that performs automatic annotation of soccer videos. This approach has resulted in detecting principal highlights, and recognizing identity of players based on face detection, and on the analysis of contextual information such as jersey's numbers and superimposed text captions. Some tests have also been performed in the AVI Indexer leading to the detection of soccer shots in TV sports news [7, 8].

Tennis is one of the sport disciplines most frequently used on content-based indexing experiments. The goals of the proposed approaches are an automatic detection of highlights in tennis games [9], action recognition [10], player detection and tracking [11], detection of faces in tennis video scenes of TV sports news [12], and event detection in tennis videos based on trajectory analysis [13].

Line detection is one of the techniques proposed for sport shot categorization. Characteristic specific lines for a given sport category can be used to detect playing field and then to detect the sport discipline. The lines on the field can also be used to determine the parameters for fast camera calibration [14]. Different solutions have been proposed for line detections, for example a gridding Hough transform [15, 16] for straight line detection or isotropic nonlinear filtering [17] for wide line detection. A method of the detection of field lines in sports videos has been patented in 2010 [18]. The problem of the detection of colour lines has been discussed in [19]. Line detection has been tested for soccer video [20, 21]. Playing fields with lines for tennis, badminton, volleyball, and soccer have been modelled in [22].

3 Strategies in Sport Categories Recognition

Content is very subjective and in consequence not easy to be recognized. Many algorithms, methods, frameworks, and strategies have be proposed for content analyses of digital videos and for automatic sport shots categorization [23]. They may be based on the traditional comparison of single frames with image pattern set or their histograms as well as on the detection of different specific elements of digital videos

typical for a given category of sport. In the case of TV sports news such elements are: lines in playing fields, player faces, sport equipments, etc.

The similarity between two images is usually based on colour histogram analyses. Therefore, histogram matching has become a common technique and many works on content-based video indexing have been based on the histogram-based patterns representations. The procedure is image matching or histogram matching is time-consuming because of a great number of frames in every video clip. The second problem is that it is easy to expect that the histograms of frames from ski jumping shots are different of those from tennis court shots. However, the histograms of football frames are very similar to baseball frames, similarly basketball to volleyball frames, or hockey frames to figure skating on ice, etc.

Text is usually present in sport video. We find the names of players or teams on player sport wears. Game places names, stadium names, league tables, numeric results, time, etc. or names of sports commentators are usually superimposed on the images, or included as closed captions. Because these textual elements are very characteristic for a specific sport discipline, so, they can serve as an important indicator in content-based indexing process. Text is omnipresent because in any sport broadcast we observe different words not only on playing fields but also on sports stadium grandstands, in the audience, and of course as publicity billboards or banners, etc. Extraction of text information involves detection, localization, tracking, extraction, enhancement, and recognition of the text from a given video frame.

The faces of most of the sportsmen are well known and easily recognizable because of the great popularity of sport idols. Their pictures are very common in the Web. The recognition of players is the next strategy for sport shot categorization. The most important phase in automatic face recognition process is face detection because it happens that for example raised up hand is taken as a face due to similar shape and colour. The distinction and the rejection of objects resembling faces but which are not is a crucial processing. When a single face is locating, we should extract the specific points such as eyes, eyebrow line, corners of the mouth, the whole mouth, nose, and other related to the chosen method of identification. These points determine the values of parameters for face recognition: symmetry, distance between the eyes, the distance between the line of eyes, and lips.

In many sport disciplines different objects are used such as ball, disc, cricket bat, javelin, tennis racket, hockey stick, net, soccer post, springboard, diving board, and many others. Players are using different sport equipments, protective equipments, wear, footwear, etc. The recognition of these objects in sport videos leads to the identification of content and to the categorization of sport video shots.

The detection and analysis of players and audience emotions is a novel viewpoint and perhaps the most sophisticated approach. We try to recognize reactions such as “exciting”, as well as “happy” and “sad” emotion while playing a game, cheering in the stadium grandstands, or observing a sports video broadcast. Emotions are produced by visual, vocal, and other physiological means. One of the important way humans display emotions is through facial expressions and body gestures. The strategy consists in creating an authentic expression database based on spontaneous emotions and then in comparing these patterns with video frames. Further, the video shots with different kinds of emotions can be also used for highlight summarization and event detection to comply with user preference.

The next strategy leads to the detection of a playing field of the game in a video shot, or of interesting area in a field, i.e. boundary lines, i.e. lines marking each end of a court, the penalty area, goal line – the end line between the goal posts in soccer, back boundary lines in tennis or basketball, etc. Playing field lines are painted on the playing surface and all lines are in the same colour, usually white, which clearly contrasts with the rest of the playing field. They are characteristic segments in playing field scenes, because these lines also determine peaks in colour histograms. The objective is also to discard the audience area often present on the sides and/or on the top of the frame. The pixels near the borders of the frame are analyzed to look for those pixels whose hue value is not belonging to the court colour nor to the court line colour. In many sport disciplines line structures are well-defined and lines provide a good feature for content analyses and for automatic sport shot categorization.

4 Tennis Court

The tennis playfield and court lines are formally defined in tennis rules and it can be noticed that they have several specific features such as parallelism, dimension proportions, or defined width. All lines on the tennis court are in the same colour, which clearly contrasts with the rest of the playing field.

Tennis court is a rectangular and flat surface, nowadays it is usually grass like Wimbledon courts in London, clay like Roland Garros courts in Paris (French Open), or hardcourt such as decoturf courts in New York (US Open) or plexicushion courts in Melbourne in Australia (Australian Open). The colour of grass courts is green, of clay courts is usually red because they are made of crushed brick, and hard courts are of any colour. The court is 23.77 m (78 feet) long, and 8.23 m (27 feet) wide for singles matches and 10.97 m (36 feet) for doubles matches (Fig. 1). Additional clear space around the court can be of the same colour as court but also its colour can be different. A net that is stretched across the full width of the court, divides a court into two equal parts. The lines and the net are most frequently white, so, the borders of the net can often be detected as a line.

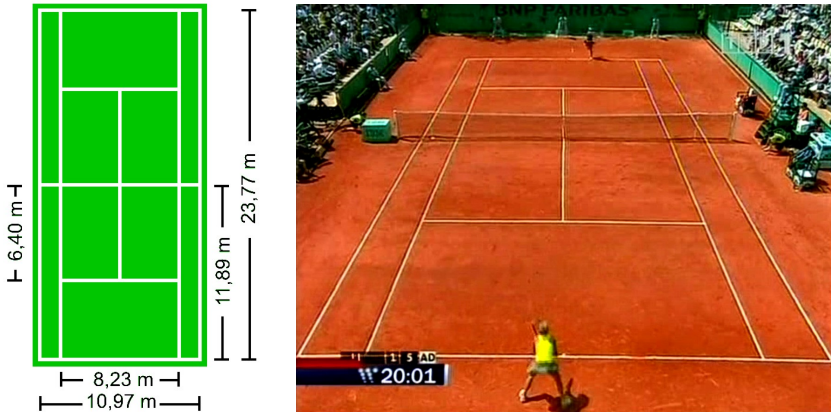


Fig. 1. Scheme of lines on a tennis court and the standard, most frequent view of a tennis game

The baselines on the court backs delineate the width of the court similarly as the lines that are called service line on the middle of the court. The short marks called centre marks are placed in the centre of each baseline. The doubles sidelines are the outermost lines that make up the length of the court and are used for doubles play. The singles sidelines used as boundaries for singles play are the lines inside of the doubles sidelines. These longest boundaries parallel lines will be crucial in the presented line detection method. The court area between the doubles sideline and the singles sideline is called the doubles alley. It is playable in doubles play. There is also another line parallel to the baselines. This line runs across the centre of each player's side of the court and is called the service. During the game the serve ball must touch this area between the service line and the net on the other side of the court. The line dividing the service line in two is called the centre line or centre service line and it is parallel to the boundaries lines. All the court lines must be between 25,4 and 50,8 mm (1 and 2 inches) in width. Whereas, the baselines can be up to almost 100 mm (4 inches) wide.

In digital videos of TV broadcast with standard resolution equal to 720 x 576 pixels court lines are relatively extremely fine lines. In the most frequent wide plan of tennis game lines are even only one pixel wide. But on the other hand lines are relatively long lines, they run almost through the entire height of the screen (Fig. 1). This observation will be used in the algorithm of tennis shot detection.

The analyses of the frames with the wide view of a tennis court, the most adequate for the shot categorization, showed that for the most characteristic lines of tennis court we can determine their minimum length. The baseline on the bottom of the screen is a minimum of 500 pixels, the service line is a minimum of 350 pixels, and the sideline is a minimum of 300 pixels (Fig. 2). These values could be used as parameters for the process of elimination of lines detected in the analyzed image.

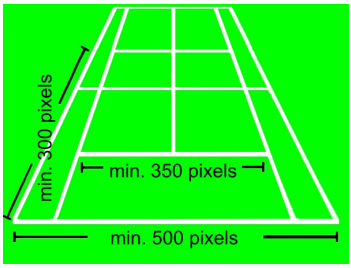


Fig. 2. The length of the main tennis lines observed in the frames with wide view in TV shots

The second important observation is that the very long sidelines, the most characteristic lines for tennis courts are parallel but in the TV image due to the perspective view lines converge quite well to the centre.

5 Line Detection of Tennis Court

The framework for line detection in sport videos leading to the classification of a analysed video (frame, shot, scene) to tennis player scene category is based on the following sequence of processes:

- binarization of video frames,
  - standard process assuming that all lines on the tennis court are white (it is the most frequent case, the test will be performed only for this case of white lines),
  - detection of the dominant playing field colour and then detection of colour of lines on the surface of a court;
- white pixel bolding in the binarized image – all neighbour pixel of every white pixel becomes also white,
- line detection – main process based on the analysis of pixels of line colour (Fig. 3),

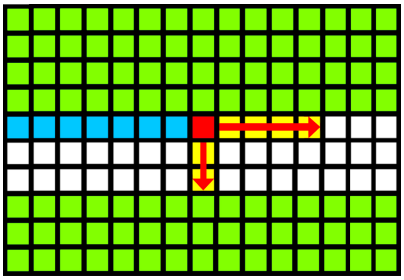


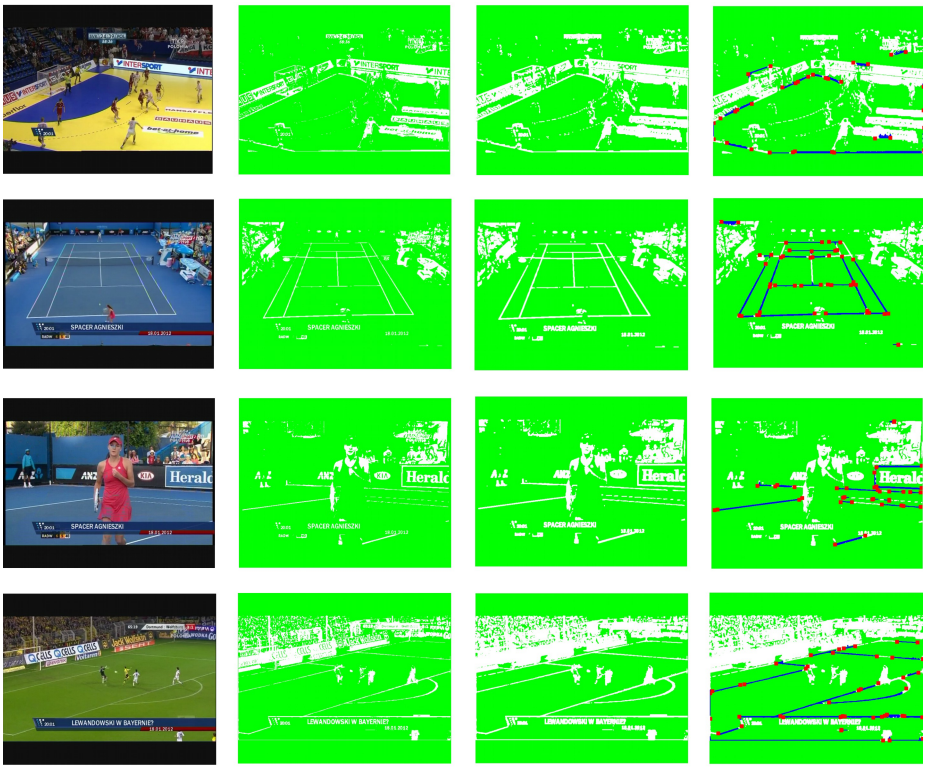
Fig. 3. Example of pixel analysis for line detection: line is defined as a sequence of pixels in the line colour (in white) towards one of the borders, central red pixel is a pixel actually being analyzed, blue pixels are pixels already assigned to a line, white pixels are pixels potentially belonging to the line, yellow pixels are pixels in the analysed direction.

- short line aggregation – because some parts of a line are often hidden by a player, TV logo, match score imposed on the video, etc. lines lying on the same direction are aggregated,
- joining of two line ends close to each other of perpendicular lines to make a corner.
- recognition of tennis shots in a TV sports videos is performed on the basis of the detection of two pairs of long lines more or less vertical and one long horizontal line at the bottom of the image – such a condition is assumed to be sufficient to select tennis shots among all shots of TV sports news.

6 Tests in AVI Indexer

The procedure of tennis event detection in sport video have been applied in the AVI indexer [8]. The tests of the efficacy of the procedure of line detection have been performed for the headlines of the Polish TV sports news of 18 January, 2012, during one of the most important competitions of world tennis. During these days in January every broadcast included tennis news and usually in headlines of sports news one of the main events presented was tennis.

Three events have been included in these headlines: two shots of handball, two shots of tennis, and one shot of soccer. To reduce the number of frames examined in the tests only two frames of each shot have been tested: the first one and the frame from the middle of the shot. Such a solution is proposed not only to reduce the time processing mainly of line detection but also because of the assumption that if one frame is detected as a tennis frame the whole shot can be categorized as a tennis shot.



a) b) c) d)

Fig. 4. Shots in headlines of TV sports news: a) frame selected from video, b) binarized frame, c) bolded white parts of the frame, d) lines (segments with red ends) detected in the frame.

In the image with the lines (image d) detected in the frame of the soccer headline (last frame in the Figure 4) we can easily identify the shapes of the lines of a soccer playfield. The analysis of soccer playfield lines should enable us to define the minimum set of lines sufficient to detect soccer shots in the sport videos and probably also for other sport disciplines.

The first shot of tennis headline (Table 1: frame 5 and 6, Figure 4) presents a wide view of tennis court, so good view for tennis detection based on line detection algorithm. The second shot (Table 1: frame 7 and 8, Figure 4) is zoom on a player. The court lines are slightly visible and even for a human is not so easy to identify the sport discipline, unless the player is popular and well known. In such a case the method based on face detection of players is more efficient for sport shots categorization [12]. For such shots we should not expect to identify a tennis court using line detection method.

The results obtained in the tests (Table 1) performed in the AVI Indexer confirm that line detection procedures are useful for the categorization of sport videos. It has been also shown that there is no need to examine the whole video, i.e. all frames in the video, we can reduce the processing to selected frames. Furthermore, not all lines

must be detected. The minimum set of lines, two pairs of long lines more or less vertical and one long horizontal line at the bottom of the image in the case of a tennis game, is sufficient for the categorization of sport video shots.

Table 1. Results of the detection of tennis shots in headlines of TV sports news

	Sport discipline	Time code [sec:frames]	Is a tennis court clearly visible in the frame?	Has tennis court been identified?	Is the result of tennis court detection correct?
1.	Handball	00:05	No	No	Yes
2.	Handball	01:22	No	No	Yes
3.	Handball	03:19	No	No	Yes
4.	Handball	04:21	No	No	Yes
5.	Tennis	06:05	Yes	Yes	Yes
6.	Tennis	08:08	Yes	Yes	Yes
7.	Tennis	10:11	only slightly	No	Yes
8.	Tennis	11:06	only slightly	No	Yes
9.	Soccer	12:05	No	No	Yes
10.	Soccer	15:00	No	No	Yes

7 Final Conclusion and Further Studies

Line detection methods are useful for the categorization of sport video shots. Not all lines must be detected. The minimum set of tennis lines includes two pairs of long vertical lines but due to the perspective view converging to the top of the image in a TV broadcast and then one long horizontal line at the bottom of the image. Such minimum set of lines is sufficient for the categorization of sport shots. The results of tests performed in the AVI Indexer have shown the usefulness of this approach.

In further research the tests on more reach video material will be performed. New solutions will be proposed for zoom frames with a small part of a tennis court may to be also sufficient to detect tennis video shots. Then, for detection of shots other sport disciplines minimum line sets will be defined on the basis of analyses of sport videos.

Finally, new computing techniques will be still developed leading to new functions implemented in the Automatic Video Indexer.

References

1. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Event detection and recognition for semantic annotation of video. Multimedia Tools and Applications 51, 279–302 (2011)

2. Geetha, P., Narayanan, V.: A survey of content-based video retrieval. Journal of Computer Science 4(6), 474–486 (2008)
3. Hu, W., Xie, N., Li, L., Zeng, X., Maybank, S.: A survey on visual content-based video indexing and retrieval. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 41(6), 797–819 (2011)
4. Money, A.G., Agius, H.: Video summarisation: a conceptual framework and survey of the state of the art. Journal of Visual Communication and Image Representation 19, 121–143 (2008)
5. Ling-Yu, D., Min, X., Qi, T., Chang-Sheng, X., Jin, J.S.: A unified framework for semantic shot classification in sports video. IEEE Transactions on Multimedia 7(6), 1066–1083 (2005)
6. Bertini, M., Del Bimbo, A., Nunziati, W.: Automatic Annotation of Sport Video Content. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 1066–1078. Springer, Heidelberg (2005)
7. Choroś, K., Pawlaczyk, P.: Content-Based Scene Detection and Analysis Method for Automatic Classification of TV Sports News. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS (LNAI), vol. 6086, pp. 120–129. Springer, Heidelberg (2010)
8. Choroś, K.: Video Structure Analysis and Content-Based Indexing in the Automatic Video Indexer AVI. In: Nguyen, N.T., Zgrzywa, A., Czyżewski, A. (eds.) Advances in Multimedia and Network Information System Technologies. AISC, vol. 80, pp. 79–90. Springer, Heidelberg (2010)
9. Huang, Y., Choiu, C., Sandnes, F.E.: An intelligent strategy for the automatic detection of highlights in tennis video recordings. Expert Systems with Applications 36(6), 9907–9918 (2009)
10. Zhu, G., Xu, C., Huang, Q., Gao, W.: Action recognition in broadcast tennis video. In: Proc. of 18th Int. Conf. on Pattern Recognition (ICPR), vol. 1, pp. 251–254 (2006)
11. Jiang, Y., Lai, K., Hsieh, C., Lai, M.: Player Detection and Tracking in Broadcast Tennis Video. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 759–770. Springer, Heidelberg (2009)
12. Choroś, K., Fijałkowski, D.: Detection of faces in tennis video scenes of TV sports news. In: Świątek, J., et al. (eds.) Information Systems Architecture and Technology: System Analysis Approach to the Design, Control and Decision Support, pp. 127–137. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2010)
13. Chi-Kao, C., Min-Yuan, F., Chung-Ming, K., Nai-Chung, Y.: Event detection for broadcast tennis videos based on trajectory analysis. In: Proc. of 2nd Int. Conf. on Communications and Networks (CECNet), pp. 1800–1803 (2012)
14. Farin, D., Han, J., de With, P.H.N.: Fast camera calibration for the analysis of sport sequences. In: Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME), pp. 482–485 (2005)
15. Yu, X., Lai, H.C., Liu, S.X.F., Leong, H.W.: A gridding Hough transform for detecting the straight lines in sports video. In: Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME), pp. 518–521 (2005)
16. Li, Y., Liu, G., Qian, X.: Ball and field line detection for placed kick refinement. In: Proc. of Global Congress on Intelligent Systems (GCIS), vol. 4, pp. 404–407 (2009)
17. Liu, L., Zhang, D., You, J.: Detecting wide lines using isotropic nonlinear filtering. IEEE Transactions on Image Processing 16(6), 1584–1595 (2007)
18. Jacob, M.G., Bhagavathy, S., Barcon-Palau, J., Llach, J.: Detection of field lines in sports videos. Patent WO2010083021 (2010)

19. Lacroix, V.: Color Line Detection. In: Maino, G., Foresti, G.L. (eds.) ICIAP 2011, Part I. LNCS, vol. 6978, pp. 318–326. Springer, Heidelberg (2011)
20. Cai, Z.Q., Tai, J.: Line detection in soccer video. In: Proc. of the Fifth Int. Conf. on Information, Communications and Signal Processing, pp. 538–541 (2005)
21. Sun, L., Liu, G.: Field lines and players detection and recognition in soccer video. In: Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP), pp. 1237–1240 (2009)
22. Kopf, S., Guthier, B., Farin, D., Han, J.: Analysis and retargeting of ball sports video. In: Proc. of IEEE Workshop on Applications of Computer Vision (WACV), pp. 9–14 (2011)
23. Choroś, K.: Video structure analysis for content-based indexing and categorisation of TV sports news. Int. J. Intelligent Information and Database Systems 6 (in press, 2012)

The Application of Orthogonal Subspace Projection in Multi-spectral Images Processing for Cancer Recognition in Human Skin Tissue

Andrzej Zacher[1], Aldona Drabik[2], Jerzy Paweł Nowacki[2], and Konrad Wojciechowski[2]

[1] The Silesian University of Technology, Institute of Informatics
Andrzej.Zacher@polsl.pl
[2] Polish-Japanese Institute of Information Technology

Abstract. This paper analyses multi-spectral images and their application in the process of cancer recognition in human skin. Cancerous part of a tissue can be characterized by higher accumulation of photosensitive substances then healthy. In order to detect the spectrum of Protoporphyrin IX in the human skin images Orthogonal Subspace Projection classifier was presented. For every pixel it calculates the content of Protoporphyrin IX spectrum in the global pixel spectrum. After pixel classification it was necessary to separate regions with cancer from healthy parts of a tissue by applying non-linear mapping with low frequency removal or mean shift segmentation enhanced with edge detection for better region recognition. Both proposals gave successful results.

1 Introduction

Multi-spectral images are obtained by the use of bandpass filters designed to collects data only for the required wavelength interval together with camera devices which are able to register and present the desired image component as gray-scale image. Multi-spectral images provide information that are almost invisible for human eye. Additionally, obtaining the discrete spectrum of each pixel is possible by simply combining the contribution of each spectral image.

Multi-spectral images were used in [1] to analyze whether gene amplification in cells is morphologically or genetically related to prior tumor invasion. Very useful for that purpose were Beltrami flow-based reaction-diffusion and directional diffusion filters. In [2] they extended the Hidden Markov Chain (HMC) model to perform a segmentation of multi-sepctral images. In order to keep mutual dependence between the layers, the Independent Component Analysis (ICA) was adopted. The outcome of unsupervised classification on a four bands SPOT-IV signal was presented. Also in [3] a method was proposed to design an automatic classifiers for discrimination between cancerous and healthy tissue. It was suggested that spectra is not sufficient to recognize fully between those two tissue classes, however some high degree of discrimination is possible. In order to do that spectral features should be selected carefully using either some kind of heuristic or proposed Haar wavelet packet method.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 315–324, 2012.

Multi-spectral images, however, have a couple of disadvantages. Each of them needs to be handled in a reasonable manner.

1. Every spectral image contains some noise. It is a high frequency noise, which sums up and highly influences the final data.
2. Multi-spectral images enable to compute directly the whole spectrum for every needed pixel. However, if sampling interval is too big, a significant number of information can be lost.
3. Every spectral image depends on the light illuminating the analyzed area.
4. Spectra obtained for pixels from images acquired for different tissue samples are difficult to compare. There is no reference between them, since every situation was illuminated in a bit different way. Also skin of every human being varies.
5. Creating a combined picture from many multi-spectral images by applying max-to-white operator can be problematic. Unimportant noise peaks in constant color image will be magnified to white color, making them the most visible.

All of those problems will be addressed in this paper by applying Orthogonal Subspace Projection (OSP) algorithm together with different image segmentation approaches. OSP method is very promising and was used as a basis for further image processing and cancer detection. The choice of this classification technique was dictated by no information where the cancer is located and if it at all exists somewhere.

OSP gives the possibility to deliver a feature-rich picture for a desired property. In other words, it is needed to specify what we are looking for and then display pixel by pixel the quantity describing how strong this parameter influences this part of image. As the result grayscale representation is obtained which can be further processed by variety of post processing methods indicating unambiguously cancerous part of a tissue.

2 Orthogonal Subspace Projection classification

A classification method widely used for multi- and hyper-spectral images is Orthogonal Subspace Projection (OSP). It bases on idea of linear unmixing of mixed pixel vectors, containing a linear combination of endmembers. Individual components can be quantified giving the number describing the amount of a given property in pixel vector. In order for the OSP algorithm to be applicable, the number of samples should be equal or greater than that for classified endmember. In our case this constraint is totally fulfilled by multi-spectral images, since only two signatures was analyzed i.e. isolated cancer spectrum and healthy skin [4].

The main idea of this classifier is to remove all undesired and unwanted signatures in a pixel. Generally those components can be treated as a background. In the end a matched filter is utilized to derive the expected spectral endmember existing in that pixel.

The problem defined for only one signature of interest, can be expressed as a mixed pixel vector r_i described by the following linear model:

$$r_i = (d\alpha_p + U\gamma) + n_i, \tag{1}$$

where:
r - column vector $l \times 1$,
l - number of samples,
i - pixel number in multi-spectral image,
p - number of distinct signatures,
α_i - column vector $p \times 1$ representing the fraction of the given endmember in r,
n - column vector $l \times 1$ describing additive, white gaussian noise,
d - column vector $l \times 1$ containing signature of interest,
α_p - faction of desired component,
U - matrix $l \times (p-1)$ composed of the remaining endmembers or just background,
γ - column vector $(p-1) \times 1$ containing the remaining fractions of α.

In out case d and U are represented by the isolated spectrum of Protoporphyrin IX and the spectrum of healthy skin tissue respectively. It means that, since the number of endmembers is equal to two, $p = 2$.

The goal of OSP classification is to find such P that eliminates the influence of matrix of unwanted components U. This operator projects r onto a subspace that is orthogonal to the columns of U by using a least squares optimal interference rejection operator:

$$P = (I - UU^{\dagger}), \tag{2}$$

where:
I - identity matrix,
$U^{\dagger}$ - pseudo inverse of U denoted by $U^{\dagger} = (U^T U)^{-1} U^T$.

Finally it is necessary to find matched filter for a desired endmember, such that maximizes the signal-to-noise ratio (SNR). Then an overall classification operator for a signature of interest in the presence of background signature and white noise can be defined as:

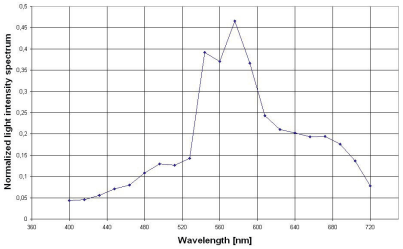
$$q^T = d^T P, \tag{3}$$

By applying this operator to all of the pixels in a multi-spectral image, each spectral vector is transformed to a scalar representing a measure of the presence of the endmember of interest. Pixels with the highest intensity denote the existence of the desired signature in the image [5].

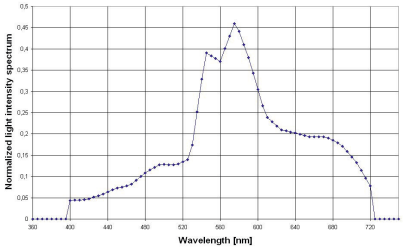
3 Data Preparation

As the input the application takes 21 multi-spectral images and as the result displays all processing steps, which finally show the location of the places, where

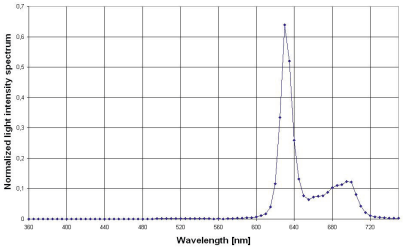
Table 1. White light spectrum and the response of Protoporphyrin IX



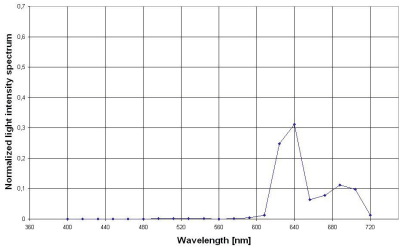
Spectrum from white light (21 samples)



Spectrum from white light (79 samples)



Spectral response on white light of Protoporphyrin IX (79 samples)



Spectral response on white light of Protoporphyrin IX (21 samples)

the cancer occurs. It is important to note that when capturing a sequence of images, they should contain not only the suspected, cancerous regions, but mainly healthy parts of tissue.

In the beginning of algorithm two fluorescence spectra need to be defined. First, the fluorescence response of Protoporphyrin IX when illuminating with white or blue light. Second, the spectrum of a healthy skin tissue.

The light spectrum was obtained by taking multi-spectral images of white fabric illuminated with appropriate light. Since every pixel in the image is a vector, in order to find an average image color, all vectors were summed, divided by the number of pixels and normalized. Having defined a spectrum of white light in form of 21 samples from 400nm to 720nm, it was necessary to convert this spectrum to 79 samples from 360nm to 750nm. This kind of spectrum can be read by the another application [6] which, utilizing Monte Carlo simulation and EEM i.e excitation-emission matrix, was able to produce a response spectra of Protoporphyrin IX illuminated with desired light. In order to resample original spectrum, a simple linear interpolation was performed.

In [6] a model of human skin tissue was proposed, where the only fluorophore available in the system was Protoporphyrin IX. For simplicity the color of the skin was neglected. The application started tracing photons in the scene, perform subsurface scattering, the fluorescence phenomenon occurred and finally only photons that escaped the tissue was stored in photon map and rendered.

The spectrum of the Protoporphyrin IX was registered as a vector of 79 samples. The shape of the spectrum and position of its maxima is exactly the same as in [7]. It means the simulation gave correct results.

Since the image processing procedure, that is going to be described, works directly on 21 multi-spectral images. It means that a fluorophore response spectrum cannot be sampled 79 times in range 360nm to 750nm, but it needs to be downsampled. Again, a linear interpolation was applied, but this time the result has some artifacts. For the wave components of higher frequency than the new sample rate the signal is missing. It is very good visible for the main peak value around 630nm, which after transformation was cut off drastically. However, the obtained 21 samples spectra is still good enough to use it in the further processing.

When the multi-spectral image of healthy skin tissue was not available or was difficult to obtain, another approach was specified. Since more than 50% of the investigated area doesn't contain a cancer, the spectrum of a healthy skin tissue was calculated as the average spectrum of all pixels in the image. In most cases healthy parts of the skin covered about 80% of the analyzed area and this simplification gave quite good results.

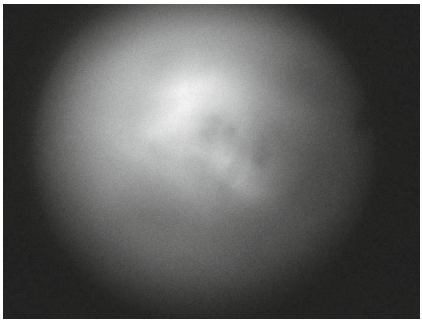
4 Algorithm

In the beginning a script reads a sequence of multi-spectral images into a 3-dimensional matrix,image cube. As the next step the Orthogonal Subspace Projection (OSP) transformation was utilized for every pixel in the image. In order to do that every spectral vector needs to be multiplied by the classification operator defined in equation (3), which requires the spectrum of the healthy tissue as a background - see formula (2). Since in the beginning the assumption was made that most of the analyzed region should be covered by healthy skin, the spectrum of undesired spectral components was calculated as the average of all pixel vectors in the image. As the result OSP transformation, 2-dimensional matrix was generated, where the value of every entry represents a measure of the presence of the Protoporphyrin IX in the spectrum.

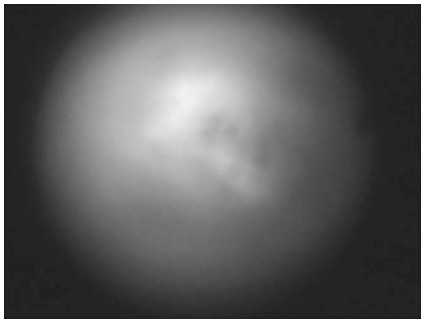
All values obtained after OSP transformation had to be transformed to intensities, which can be displayed on the computer screen. For this reason max-to-white operator was applied. It maps the lowest values to black color (0), the highest to white color (1) and the rest are uniformly interpolated between those two boundary values. Matrix after this operation could be finally stored in form of an image - Fig. 1a.

Obtained image has a very noticeable, high frequency noise, which was eliminated by applying a median filter of size 9x9 pixels - Fig. 1b. Now everything looks smoother and can be further analyzed.

Orthogonal Subspace Projection applied per pixel converts 23-dimensional multi-spectral image space (2-dimensions represents position, 21-dimensions describes spectrum of each pixel) into 3-dimensional space. It can be imagined as terrain landscape, where the higher the point is, the greatest is the level of

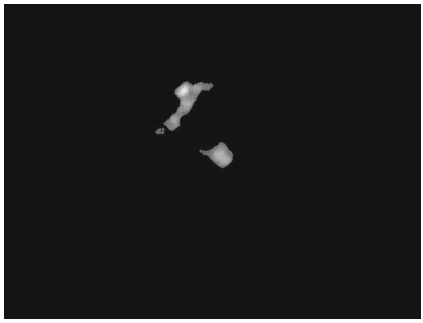


(a) Image after OSP.

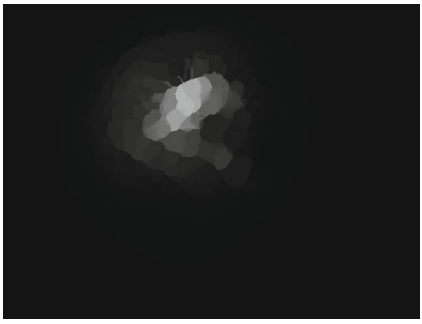


(b) Image after median filtering.

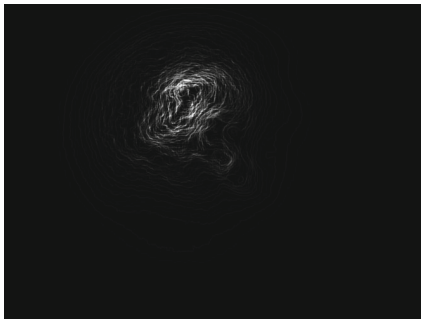
(c) Image after nonlinear mapping.



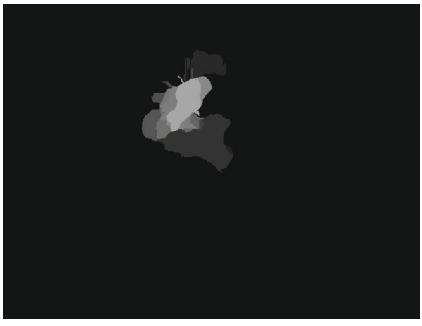
(d) High frequency image.



(e) Image after mean shift filtering.



(f) Gradient image.

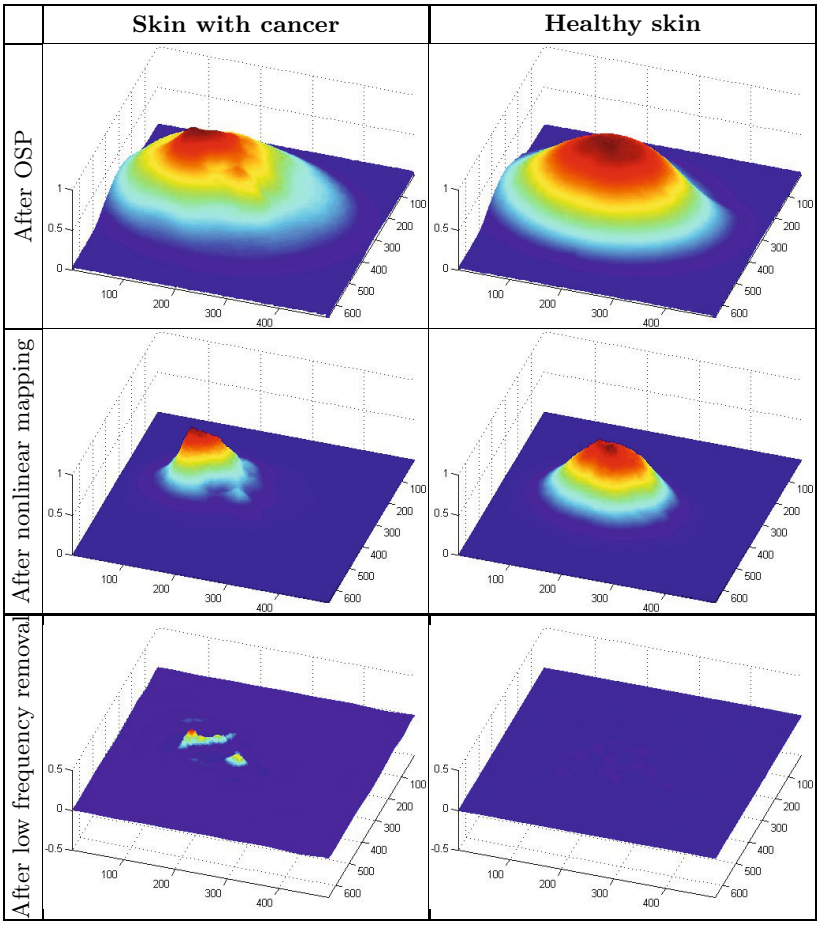


(g) Segmented image.

(h) Image with merged regions.

Fig. 1. Cancer recognition for tissue sample signature - 15674/1 (cancer)

Table 2. Comparison the process of filtering for healthy and cancerous skin tissue



fluorophore concentration (see table 2). Because of this fact, only distinct pixels should be isolated, which have a greater intensity than some threshold. A nonlinear transformation was applied for each pixel that amplifies biggest value and suppress the lowest ones. Every intensity defined in range from 0 to 1 was raised to the power of four. This exponent gave the best value and was found by experiments - Fig. 1c.

Looking at Table 2 it is important to notice, that no matter whether investigated area contains fluorescent substance or not, the OSP image exhibits similar curvature to 2-dimensional Gaussian function. This bias comes from the center of the skin illuminated by the light source. In the middle of the image the light intensity is the highest, but decreases slowly with the distance to the midpoint. This influence interfere the OSP calculation making low values much higher than they really are. Despite of the fact that all vectors used by Orthogonal Subspace

Projection were normalized, the impact of higher energy spectrum is still significant. In order to avoid the effect of central region illumination, a frequency filter was designed. The cut off value *K0* was chosen carefully, so that only signal of low frequencies remained, but other signal components were removed. Finally, the 2-dimensional Fourier transform was calculated, then multiplied by filter and eventually 2-dimensional inverse Fourier transform computed. Obtained in this way image was subtracted from the original one leaving only signal of high frequency. As it can be seen in the last row of Table 2, the bias was removed and the remaining part is of big interest. In the end all pixel values below a given threshold (t=0.05) were removed. Also a closing operator was applied to the image to eliminate not smooth cancer edges. The result is presented at Fig. 1d.

Instead of applying high pass filter to the OSP image also another approach was investigated. Mean shift filtering with synergistic segmentation based on edge detection mechanism was applied using EDISON application [9]. It was decided to use Mean Shift Synergistic Segmentation, because it relies on gradient vectors between separate pixels. If the gradient is too small, then most probably adjacent points belong to the same cluster. This is the desired behavior, since only rapid changes are of high importance and indicate the appearance of fluorophore in investigated area. Mean shift segmentation algorithm has also another advantage. It enables to configure parameters of image filtering and segmentation in a handy manner. It is not known in advance how many clusters there is going to be, so popular k-means clustering technique couldn't be used. Additionally basing on the images generated by EDISON, even if the results are not satisfactory, one can draw a proper conclusion about fluorophore concentration. Finally, mean shift segmentation is fast and the outcome of image processing was successful [8].

It was decided to create filtered 1e, gradient 1f and segmented image 1g as the output files of the application. Filtered image shows clustered image before segmentation. It can be analyzed if segmentation phase was too greedy or generous. Gradient image can be also very helpful, since bright and thick lines indicate big changes in pixel intensities and can be analyzed as the potential region of cancer. The only drawback of this approach is the direction of the pixel value changes. From the point of view of feature recognition, only pixels with higher values than background are interesting. However, gradient and also segmented image will indicate not only "hills", but also "valleys" in OSP image. Fortunately it didn't have big influence on final results.

As the last step after image segmentation, additional merging of obtained regions was performed. The idea behind that was to show only black and white image, where white color corresponds to cancer and black to healthy parts of the skin. The resulting image is presented in figure 1h.

5 Results

In order to obtain ground truth data 10 patients were diagnosed based on histopathological examiantion of skin stretch. Found 3 patients with Basal Cell

Carcinoma, 3 patients with Bowen's disease - carcinoma in situ, 2 patients with pigmented nevi, 1 with actinic keratosis and 1 patient with Barrett's esophagus. All result were additionally compared with autofluorescence diagnosis based on Xillix Onco-LIFE.

All of the investigated tissue samples were classified correctly by either image filtering or mean shift segmentation. It was proven that the algorithm is able to recognize healthy and cancerous skin either for different or for the same patient. It was also able to localize diseased tissue on the image very precisely. As the advantage, the process of recognition has several steps. Each of them can help to provide more accurate and reliable diagnosis. Sometimes even single OSP image is enough to get good understanding of tissue condition. In all 18 cases filtering always correctly recognized cancerous changes. Synergistic segmentation was however successful in 16 cases, which is still very good.

For tissue samples illuminated from blue light source the situation is different. It turned out that it is extremely difficult to take a multi-spectral photograph of a skin tissue in such environment. Among 21 available samples only 7 could be used by image recognition algorithms. It is only 33% of all available samples. The rest of them was very noisy or there was no data at all. It means that it is easier to take the photo of tissue illuminated with white light. Probably for blue light it is more difficult to find proper camera configuration and obtain sharp images. Also important seems to be the distance to the skin.

Images illuminated with blue light and containing useful information were also examined and the outcome was very satisfactory. Removal of low frequency disturbances always correctly classified skin samples and gave exact position of diseased parts of a tissue. However, mean shift segmentation wasn't that good as for the white light and only for 4 among 7 cases gave the expected results. It seems that for blue light illumination synergistic segmentation cannot correctly recognize intensity changes on the image. Most probably another parameters settings would be needed.

6 Conclusions

The application of Orthogonal Subspace Projection classification together with various postprocessing methods gave vary successful results in area of cancer recognition and photodynamic diagnosis. Low frequency removal and mean shift filtering enhanced with synergistic segmentation were good candidates to unambiguously localize suspicious places in human skin. The algorithm was able to correctly detect tumor and healthy human skin tissue in almost 100% cases for white and blue light sources. The obtained results are very promising. They gave the idea about tumor existence without performing tissue examination by histopathology. The application of multi-spectral images in photodynamic diagnosis uncovers properties of a tissue that normally are not visible by human observer without additional tools. However, this algorithm is not only limited to human skin tissue. As the future work it would be challenging to adjust presented method to internal parts of the human body like oesophagus or stomach.

References

1. Adiga, U., Malladi, R., Fernandez-Gonzalez, R., de Solorzano, C.O.: High-Throughput Analysis of Multispectral Images of Breast Cancer Tissue. IEEE Transactions on Image Processing 15(8), 2259–2268 (2006)
2. Derrode, S., Mercier, G., Pieczynski, W.: Unsupervised multicomponent image segmentation combining a vectorial HMC model and ICA. In: IEEE International Conference on Image Processing, vol. II, pp. 407–410 (2003)
3. Woolfe, F., Maggioni, M., Davis, G., Warner, F., Coifman, R., Zucker, S.: Hyperspectral microscopic discrimination between normal and cancerous colon biopsies. IEEE Transactions on Medical Imaging and Remote Sensing 99(99) (1999)
4. Ren, H., Chang, C.-I.: A Generalized Orthogonal Subspace Projection Approach to Unsupervised Multispectral Image Classification. IEEE Transactions on Geoscience and Remote Sensing 38(6), 2515–2528 (2000)
5. Ientilucci, E.J.: Hyperspectral Image Classification Using Orthogonal Subspace Projections: Image Simulation and Noise Analysis. Rochester Institute of Technology, College of Science, Center for Imaging Science, Digital Imaging and Remote Sensing Laboratory (2001)
6. Zacher, A.: Utilization of Multi-spectral Images in Photodynamic Diagnosis. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010, Part II. LNCS, vol. 6375, pp. 367–375. Springer, Heidelberg (2010)
7. Moesta, K.T., Ebert, B., Handke, T., Nolte, D., Nowak, C., Haensch, W.E., Pandey, R.K., Dougherty, T.J., Rinneberg, H., Schlag, P.M.: Protoporphyrin IX Occurs Naturally in Colorectal Cancers and Their Metastases. Cancer Research 61(3), 991–999 (2001)
8. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
9. Christoudias, C.M., Georgescu, B., Meer, P.: Synergism in Low Level Vision. In: 16th International Conference on Pattern Recognition, Quebec City, Canada, vol. IV, pp. 150–155 (2002)

Length and Coverage of Inhibitory Decision Rules

Fawaz Alsolami[1], Igor Chikalov[1], Mikhail Moshkov[1], and Beata Marta Zielosko[1,2]

[1] Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{fawaz.alsolami,igor.chikalov,mikhail.moshkov,beata.zielosko}@kaust.edu.sa
[2] Institute of Computer Science, University of Silesia
39, Będzińska St., 41-200 Sosnowiec, Poland

Abstract. Authors present algorithms for optimization of inhibitory rules relative to the length and coverage. Inhibitory rules have a relation "attribute $\neq$ value" on the right-hand side. The considered algorithms are based on extensions of dynamic programming. Paper contains also comparison of length and coverage of inhibitory rules constructed by a greedy algorithm and by the dynamic programming algorithm.

Keywords: inhibitory decision rules, length, coverage, dynamic programming.

1 Introduction

In the paper, we study algorithms for optimization of inhibitory rules. Such rules on the right hand side have a relation "attribute $\neq$ value" in contrast to usual rules which right-hand side is presented as "attribute = value".

It was shown in [10, 11] that, for some information systems, usual rules cannot describe the whole information contained in the system. However, inhibitory rules describe the whole information for every information system [7]. Classifiers based on inhibitory rules have often better accuracy than classifiers based on usual rules [4–6]. Greedy algorithms for inhibitory rule construction were studied in [7].

In this paper, we consider algorithms for optimization of inhibitory rules relative to the length and coverage. Such algorithms are based on extensions of dynamic programming. For a given decision table T, we construct a directed acyclic graph $\Lambda(T)$. Nodes of this graph are subtables of the table T given by systems of equations of the kind "attribute = value". We finish the partitioning of a subtable when it has less different decisions than T. This graph allows us to describe the set of so-called nonredundant inhibitory rules. After the optimization relative to the length, we obtain a changed graph $\Lambda(T)$ which describes all nonredundant inhibitory rules with minimum length. In the case of optimization relative to the coverage we obtain a changed graph $\Lambda(T)$ which describes all nonredundant inhibitory rules with maximum coverage. The choice of length is

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 325–334, 2012.

connected with the Minimum Description Length principle [9]. The rule coverage is important to discover major patterns in the data.

In [1], we presented procedure of optimization of inhibitory rules relative to the length. In [2], we describe, inter alia procedure of optimization relative to the coverage. In this paper, we present comparison of length and coverage of inhibitory rules constructed by dynamic programming algorithm with the length and coverage of inhibitory rules constructed by a greedy algorithm. We describe also sequential optimization of inhibitory rules relative to length and coverage, and compute number of rows in decision tables for which exist rules with minimum length and maximum coverage. Dynamic programming approach allows us to find optimal (from different points of view) inhibitory decision rules. In the case of sequential optimization relative to the coverage and length, we find nonredundant inhibitory decision rules with maximum coverage and among them, rules with minimum length. We can find also totally optimal nonredundant inhibitory decision rules relative to the length and coverage, i.e., rules with maximum coverage and minimum length. Similar approach for usual exact decision rule optimization was considered in [3].

We consider also results of experiments with some decision tables from UCI ML Repository [8].

The paper consists of eight sections. In Sect. 2, we present main notions. In Sect. 3, a directed acyclic graph is considered. Based on this graph we can describe the whole set of nonredundant inhibitory rules for each row of a decision table. Sections 4 and 5 contain descriptions of procedures of optimization relative to the length and coverage. In Sect. 6, we discuss a possibility of sequential rule optimization relative to different criteria. Section 7 contains results of experiments and Sect. 8 – conclusions.

2 Main Notions

In this section, we present definitions of notions corresponding to decision tables and inhibitory rules.

A *decision table* T is a rectangular table with n columns labeled with conditional attributes $f_1, \ldots, f_n$. Rows of this table are filled with nonnegative integers which are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer (decision) which is interpreted as a value of the decision attribute d. We denote by $D(T)$ the set of decisions attached to rows of the table T. We denote by $N(T)$ the number of rows in the table T.

A table obtained from T by the removal of some rows is called a *subtable* of the table T. A subtable T' of the table T is called *reduced* if $|D(T')| < |D(T)|$, and *unreduced* otherwise when $|D(T')| = |D(T)|$.

Let T be nonempty, $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $a_1, \ldots, a_m$ be nonnegative integers. We denote by $T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m)$ the subtable of the table T which contains only rows that have numbers $a_1, \ldots, a_m$ at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$. Such nonempty subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ the set of attributes from $\{f_1, \ldots, f_n\}$ which are not constant on T. For any $f_i \in E(T)$, we denote by $E(T, f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i_1} = a_1 \wedge \ldots \wedge f_{i_m} = a_m \rightarrow d \neq k \tag{1}$$

is called an *inhibitory rule over* T if $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $a_1, \ldots a_m$ are nonnegative integers, and $k \in D(T)$. It's possible that $m = 0$. In this case (1) is equal to the rule

$$\rightarrow d \neq k. \tag{2}$$

Let Θ be a subtable of T and $r = (b_1, \ldots, b_n)$ be a row of Θ. We will say that the rule (1) is *realizable for* r, if $a_1 = b_{i_1}, \ldots, a_m = b_{i_m}$. The rule (2) is realizable for any row from Θ.

We will say that the rule (1) is *true for* Θ if each row of Θ for which the rule (1) is realizable has the decision attached to it that is different from k. The rule (2) is true for Θ if and only if each row of Θ is labeled with the decision different from k. If the rule (1) is an inhibitory rule over T which is true for Θ and realizable for r, we will say that (1) is an *inhibitory rule for* Θ *and* r *over* T.

We will say that the rule (1) with $m > 0$ is a *nonredundant* inhibitory rule for Θ and r over T if (1) is an inhibitory rule for Θ and r over T and the following conditions hold:

(i) $f_{i_1} \in E(\Theta)$, and if $m > 1$ then $f_{i_j} \in E(T(f_{i_1}, a_1) \ldots (f_{i_{j-1}}, a_{j-1}))$ for $j = 2, \ldots, m$;

(ii) if $m = 1$ then Θ is unreduced, and if $m > 1$ then the subtable $\Theta' = \Theta(f_{i_1}, a_1) \ldots (f_{i_{m-1}}, a_{m-1})$ is unreduced.

The rule (2) is a *nonredundant* inhibitory rule for Θ and r over T if (2) is an inhibitory rule for Θ and r over T, i.e., if each row of Θ is labeled with the decision different from k and $k \in D(T)$.

Let Θ be a subtable of T, τ be a nonredundant rule over T, and τ be equal to (1).

The number m of conditions on the left-hand side of τ is called the *length* of this rule and is denoted by $l(\tau)$. The length of inhibitory rule (2) is equal to 0.

The *coverage* of τ relative to Θ is the number of rows in Θ for which τ is realizable and which are labeled with the decisions other than k. We denote it by $c(\tau)$. The coverage of inhibitory rule (2) relative to Θ is equal to the number of rows in Θ which are labeled with the decisions other than k. If τ is true for Θ then $c(\tau) = N(\Theta(f_{i_1}, a_1) \ldots (f_{i_m}, a_m))$.

3 Directed Acyclic Graph $\Lambda(T)$

Now, we consider an algorithm that constructs a directed acyclic graph $\Lambda(T)$ which will be used to describe the set of nonredundant inhibitory rules for T and for each row r of T over T. Nodes of the graph are some separable subtables of the table T. During each step, the algorithm processes one node and marks it

with the symbol *. At the first step, the algorithm constructs a graph containing a single node T which is not marked with *.

Let us assume that the algorithm has already performed p steps. We describe now the step $(p+1)$. If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $\Lambda(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet. If Θ is reduced, then mark Θ with the symbol * and go to the step $(p+2)$. Otherwise, for each $f_i \in E(\Theta)$, draw a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{b_1, \ldots, b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, b_1), \ldots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. We label each row r of Θ with the set of attributes $E_{\Lambda(T)}(\Theta, r) = E(\Theta)$ (this set can be changed during a procedure of optimization). Mark the node Θ with the symbol * and proceed to the step $(p+2)$.

The graph $\Lambda(T)$ is a directed acyclic graph. A node of this graph will be called *terminal* if there are no edges leaving this node. Note that a node Θ of $\Lambda(T)$ is terminal if and only if Θ is reduced.

Later, we will describe procedures of optimization of the graph $\Lambda(T)$ relative to the length and coverage. As a result we will obtain a graph Γ with the same sets of nodes and edges as in $\Lambda(T)$. The only difference is that any row r of each unreduced table Θ from Γ is labeled with a nonempty set of attributes $E_\Gamma(\Theta, r) \subseteq E(\Theta)$.

Let G be the graph $\Lambda(T)$ or a graph Γ obtained from $\Lambda(T)$ by procedures of optimization.

Now for each node Θ of G and for each row r of Θ we describe a set of inhibitory rules $Rul_G(\Theta, r)$ over T. Let Θ be a terminal node of G, i.e., Θ is a reduced table. Then

$$Rul_G(\Theta, r) = \{\rightarrow d \neq k : k \in D(T) \setminus D(\Theta)\}.$$

Let now Θ be a nonterminal node of G such that for each child Θ' of Θ and for each row r' of Θ' the set of rules $Rul_G(\Theta', r')$ is already defined. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. For any $f_i \in E_G(\Theta, r)$, we define the set of rules $Rul_G(\Theta, r, f_i)$ as follows:

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \rightarrow d \neq k : \alpha \rightarrow d \neq k \in Rul_G(\Theta(f_i, b_i), r)\}.$$

Then

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i).$$

Theorem 1. *For any node Θ of $\Lambda(T)$ and for any row r of Θ, the set $Rul_{\Lambda(T)}(\Theta, r)$ is equal to the set of all nonredundant inhibitory rules for Θ and r over T.*

Let us consider a decision table T_0 presented at the top of Fig. 1. We denote by G_0 the graph $\Lambda(T_0)$ which is depicted in Fig. 1. For each node (subtable) Θ of G_0 which contains the last row r_4 of the table T_0 we add to Θ the set of all nonredundant inhibitory rules for Θ and r_4 over T_0.

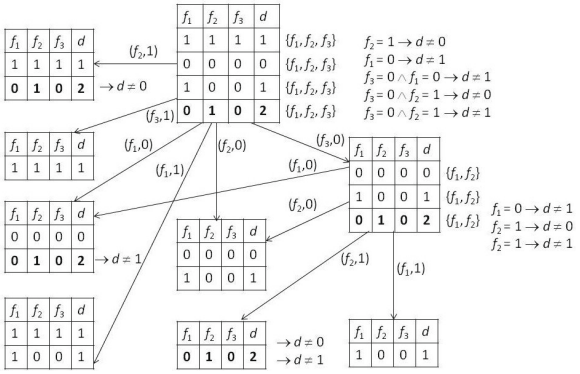


Fig. 1. Graph $G_0 = \Lambda(T_0)$

4 Procedure of Optimization Relative to Length

In this section, we describe the procedure of optimization of the graph G relative to the length l.

We will move from the terminal nodes of the graph G which are reduced subtables to the node T. We will assign to each row r of each table Θ the number $Opt^l_G(\Theta, r)$ – the minimum length of an inhibitory rule from $Rul_G(\Theta, r)$, and we will change the set $E_G(\Theta, r)$ attached to the row r in the nonterminal table Θ. We denote the obtained graph by $G(l)$.

Let Θ be a terminal node of G. Then we assign to each row r of Θ the number $Opt^l_G(\Theta, r) = 0$.

Let Θ be a nonterminal node and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We assign the number

$$Opt^l_G(\Theta, r) = \min\{Opt^l_G(\Theta(f_i, b_i), r) + 1 : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

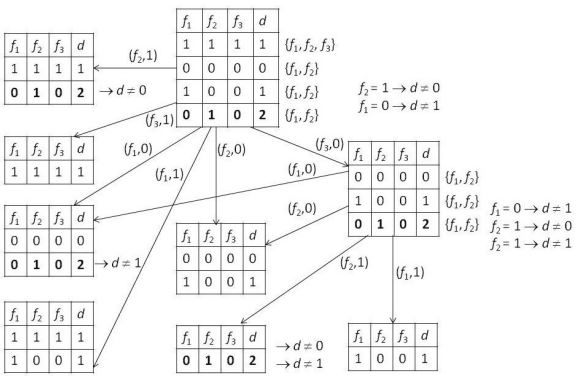


Fig. 2. Graph $G_0(l)$

$$E_{G(l)}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^l_G(\Theta(f_i, b_i), r) + 1 = Opt^l_G(\Theta, r)\}.$$

One can show that, for each node Θ of the graph $G(l)$ and for each row r of Θ, the set of rules $Rul_{G(l)}(\Theta, r)$ coincides with the set of all rules with the minimum length from $Rul_G(\Theta, r)$.

Figure 2 presents the directed acyclic graph $G_0(l)$ obtained from the graph G_0 (see Fig. 1) by the procedure of optimization relative to the length. For each node (subtable) Θ of $G_0(l)$ which contains the last row r_4 of the table T_0 we add to Θ the set of all nonredundant inhibitory rules for Θ and r_4 over T_0 with minimum length.

5 Procedure of Optimization Relative to Coverage

In this section, we describe the procedure of optimization of the graph G relative to the coverage c.

We will move from the terminal nodes of the graph G which are reduced subtables to the node T. We will assign to each row r of each table Θ the number $Opt^c_G(\Theta, r)$ – the maximum coverage of an inhibitory rule from $Rul_G(\Theta, r)$, and we will change the set $E_G(\Theta, r)$ attached to the row r in the nonterminal table Θ. We denote the obtained graph by $G(c)$.

Let Θ be a terminal node of G. Then we assign the number

$$Opt^c_G(\Theta, r) = N(\Theta)$$

to each row r of Θ.

Let Θ be a nonterminal node and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We assign the number

$$Opt^c_G(\Theta, r) = \max\{Opt^c_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

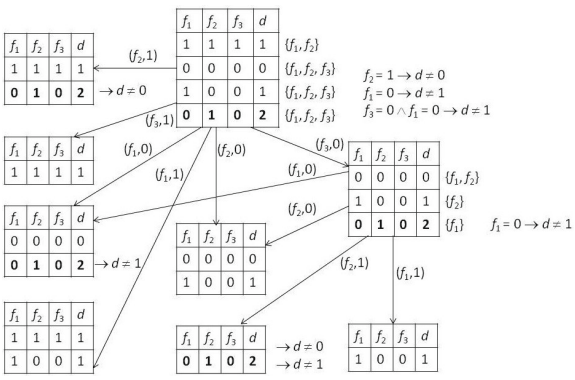


Fig. 3. Graph $G_0(c)$

$$E_{G(c)}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^c_G(\Theta(f_i, b_i), r) = Opt^c_G(\Theta, r)\}.$$

One can show that, for each node Θ of the graph $G(c)$ and for each row r of Θ, the set of rules $Rul_{G(c)}(\Theta, r)$ coincides with the set of all rules with the maximum coverage from $Rul_G(\Theta, r)$.

Figure 3 presents the directed acyclic graph $G_0(c)$ obtained from the graph G_0 (see Fig. 1) by the procedure of optimization relative to the coverage. For each node (subtable) Θ of $G_0(c)$ which contains the last row r_4 of the table T_0 we add to Θ the set of all nonredundant inhibitory rules for Θ and r_4 over T_0 with maximum coverage.

6 Sequential Optimization

We can make sequential optimization of nonredundant inhibitory rules relative to the length and coverage. We can find all nonredundant inhibitory rules with maximum coverage and after that among these rules find all rules with minimum length. We can also change the order of optimization, i.e., find all nonredundant inhibitory rules with minimum length and after that find among such rules all rules with maximum coverage. Figure 4 presents the directed acyclic graph $G_0(cl)$ obtained from the graph $G_0(c)$ (see Fig. 3) by the procedure of optimization relative to the length. For each node (subtable) Θ of $G_0(cl)$ which contains the last row r_4 of the table T_0 we add to Θ the set of all rules with minimum length among all nonredundant inhibitory rules for Θ and r_4 over T_0 with maximum coverage. For row r_4 of T_0, we found two totally optimal relative to length and coverage nonredundant inhibitory rules (see Fig 4). It is clear that these rules have maximum coverage. One can show (see Fig. 2) that the considered rules have minimum length.

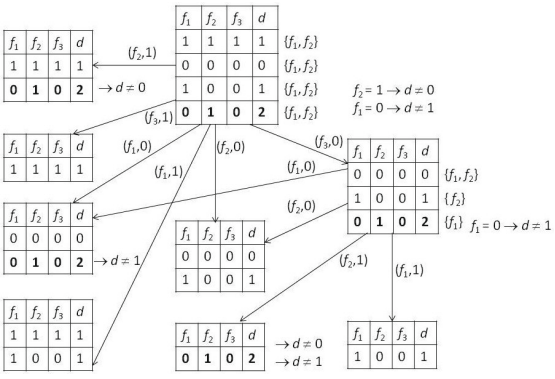


Fig. 4. Graph $G_0(cl)$

7 Experimental Results

We considered decision tables from UCI Machine Learning Repository [8]. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. In some tables there were equal rows with, possibly, different decisions. In this case each group of identical rows was replaced with a single row from the group with the most common decision for this group. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute.

For each such decision table T we constructed the directed acyclic graph $\Lambda(T)$. We applied to $\Lambda(T)$ the procedure of optimization relative to the length. Minimum, maximum and average length of obtained rules (among all rows of T) can be found in Table 1 (column "Dynamic programming"). Also, we applied to $\Lambda(T)$ the procedure of optimization relative to the coverage. Minimum, maximum and average coverage of obtained rules (among all rows of T) can be found in Table 2 (column "Dynamic programming").

We used also a greedy algorithm to construct inhibitory rules [7]. Average length and average coverage of obtained rules (among all rows of T) can be found in Tables 1 and 2 (column "Greedy") respectively. To see the improvements for rules constructed by the dynamic programming approach relative to the rules constructed by the greedy algorithm we compute:

- for average length the value $\frac{Greedy - Dynamic\_programming}{Greedy}$ (column "Improvement" in Table 1),
- for average coverage the value $\frac{Dynamic\_programming - Greedy}{Greedy}$ (column "Improvement" in Table 2).

Table 1. Length of inhibitory rules

Decision table	Rows	Attr	Dynamic programming			Greedy	Improvement
			min	max	average	average	%
adult-stretch	16	4	1	2	1.25	1.25	0.00
balance-scale	625	4	2	4	2.67	2.70	1.18
breast-cancer	266	9	1	6	2.67	2.73	2.24
cars	1728	6	1	3	1.05	1.46	28.24
hayes-roth-data	69	4	1	3	1.67	1.67	0.00
lymphography	148	18	1	1	1.00	1.14	11.89
nursery	12960	8	1	1	1.00	1.13	11.43
shuttle-landing	15	6	1	4	1.40	1.40	0.00
soybean-small	47	35	1	1	1.00	1.00	0.00
teeth	23	8	1	1	1.00	1.00	0.00
zoo	59	16	1	1	1.00	1.00	0.00

Results in Table 1 show that improvement according to the length of rules constructed by dynamic programming algorithm is not significant usually. Opposite situation we can see in Table 2 for improvement according to the coverage of rules constructed by dynamic programming algorithm. Only for data set adult-stretch we don't have any changes in average values of coverage. The biggest improvement we can observe for data sets lymphography, soybean-small, teeth, zoo and breast-cancer.

Table 2. Coverage of inhibitory rules

Decision table	Rows	Attr	Dynamic programming			Greedy	Improvement
			min	max	average	average	%
adult-stretch	16	4	4	8	7.00	7.00	0.00
balance-scale	625	4	1	25	11.94	11.66	2.42
breast-cancer	266	9	1	25	9.53	4.09	132.88
cars	1728	6	36	576	543.74	419.06	29.75
hayes-roth-data	69	4	3	12	7.61	7.23	5.21
lymphography	148	18	77	142	141.00	20.84	576.65
nursery	12960	8	4320	6480	5400.00	3084.01	75.10
shuttle-landing	15	6	1	3	2.13	1.87	14.25
soybean-small	47	35	37	37	37.00	8.89	316.01
teeth	23	8	14	17	16.22	3.74	333.73
zoo	59	16	47	51	50.46	13.37	277.31

Table 3. Sequential optimization of inhibitory rules

Decision table	coverage+length		length+coverage		Rows with	Time in [ms]	
	coverage	length	length	coverage	tot opt	dp	greedy
adult-stretch	7.00	1.25	1.25	7.00	16	99	5
balance-scale	11.94	2.67	2.67	11.94	625	810	17
breast-cancer	9.53	3.43	2.67	7.04	133	8563	17
cars	543.74	1.05	1.05	543.74	1728	2740	20
hayes-roth-data	7.61	1.70	1.67	7.58	67	115	7
lymphography	141.00	1.00	1.00	141.00	148	386	5
nursery	5400.00	1.00	1.00	5400.00	12960	14419	100
shuttle-landing	2.13	1.73	1.40	1.87	13	148	2
soybean-small	37.00	1.00	1.00	37.00	47	591	6
teeth	16.22	1.00	1.00	16.22	23	64	2
zoo	50.46	1.00	1.00	50.46	59	194	2

We applied also to the directed acyclic graph $\Lambda(T)$ sequentially the procedure of optimization relative to the coverage and after that the procedure of optimization relative to the length. Average length and coverage of obtained rules (among all rows of T) can be found in Table 3 (column "coverage+length"). After that, we compute the number of rows for which there exist totally optimal rules relative to length and coverage (column "Rows with tot opt"). We also present results of sequential optimization relative to the length and then relative to the coverage (column "length+coverage"). For data sets breast-cancer, hayes-roth-data and shuttle-landing the number of rows with totally optimal relative to length and coverage nonredundant inhibitory rules is less than the number of rows in these decision tables. In this case, values of average length and average coverage of rules depend on the order of optimization. The last two columns in Table 3 present time in miliseconds of performed experiemnts for dynamic programming algorithm (column "dp") and greedy algorithm (column "greedy").

8 Conclusions

In the paper, we considered algorithms for exact inhibitory rule optimization relative to the length and coverage which are based on extensions of dynamic programming. Presented results show that improvement according to the coverage of rules constructed by the dynamic programming algorithm in comparison

with rules constructed by greedy algorithm is significant. Sequential optimization allows often to construct rules both with minimum length and maximum coverage. Short rules which cover many objects can be useful from the point of view of knowledge representation. Further we will study approximate inhibitory rules using dynamic programming approach.

References

1. Alsolami, F., Chikalov, I., Moshkov, M., Zielosko, B.: Optimization of inhibitory decision rules relative to length. Studia Informatica 33, 395–406 (2012)
2. Alsolami, F., Chikalov, I., Moshkov, M., Zielosko, B.: Optimization of Inhibitory Decision Rules Relative to Length and Coverage. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 149–154. Springer, Heidelberg (2012)
3. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic Programming Approach for Exact Decision Rule Optimization. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems. ISRL, vol. 42, pp. 211–228. Springer, Heidelberg (2013)
4. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Two Families of Classification Algorithms. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 297–304. Springer, Heidelberg (2007)
5. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Comparison of Lazy Classification Algorithms Based on Deterministic and Inhibitory Decision Rules. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 55–62. Springer, Heidelberg (2008)
6. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Lazy classification algorithms based on deterministic and inhibitory rules. In: Magdalena, L., Ojeda-Aciego, M., Verdegay, J.L. (eds.) IPMU 2008, Torremolinos (Malaga), Spain, June 22-27, pp. 1773–1778 (2008)
7. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: Inhibitory Rules in Data Analysis. SCI, vol. 163. Springer, Heidelberg (2009)
8. Frank, A., Asuncion, A.: UCI ML Repository, http://archive.ics.uci.edu/ml
9. Rissanen, J.: Modeling by shortest data description. Automatica 14, 465–471 (1978)
10. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences 41(3), 237–254 (1993)
11. Suraj, Z.: Some Remarks on Extensions and Restrictions of Information Systems. In: Ziarko, W., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 204–211. Springer, Heidelberg (2001)

Refining the Judgment Threshold to Improve Recognizing Textual Entailment Using Similarity

Quang-Thuy Ha*, Thi-Oanh Ha, Thi-Dung Nguyen, and Thuy-Linh Nguyen Thi

Vietnam National University, Hanoi (VNU), College of Technology (UET),
144, Xuan Thuy, Cau Giay, Hanoi, Vietnam
{OanhHT,DungNT,LinhNTT,ThuyHQ}@vnu.edu.vn

Abstract. In recent years, Recognizing Textual Entailment (RTE) catches strongly the attention of the Natural Language Processing (NLP) community. Using Similarity is an useful method for RTE, in which the Judgment Threshold plays an important role as the learning model. This paper proposes an RTE model based on using similarity. We describe clearly the solutions to determine and to refine the Judgment Threshold for Improvement RTE. The measure of the synonym similarity also is considered. Experiments on a Vietnamese version of the RTE3 corpus are showed.

Keywords: Recognizing Textual Entailment, the Judgment Threshold, Refining the Judgment Threshold.

1 Introduction

In recent years, Recognizing Textual Entailment (RTE) becomes an important task in the field of Natural Language Processing (NLP). There are some methods for Recognizing Textual Entailment Methods [1,2,6,11]. The RTE methods based on similarity are very useful [3,7,10,11,12]. In the methods, there exists the problem to determine the judgment threshold [2,3,5,6,11]. Rui Wang [11] even considered the problem as a very difficult problem. In almost works based on the methods, the way to determine the judgment threshold had not been described clearly.

In this work, an RTE model based on using similarity is proposed. In this model, we describe clearly formula for determining the judgment threshold. We also propose a solution for refining the judgment threshold to Improve RTE performance.

This paper makes the following contributions:

- Propose an explicit formula to determine the judgment threshold for recognizing textual entailment using similarity.
- Using a secondary learning dataset to refine the judgment threshold for improvement the RTE performance.

The rest of this article is organized as following. In the next section, a Textual Entailment Recognizing model based on using similarity is showed. Our solutions to determine and to refine the judgment threshold is described clearly. Experiments and

* Corresponding author.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 335–344, 2012.

remarks are described in the third section. In fourth section, related works are introduced to show the importance of the judgment threshold in recognizing textual entailment systems. The thresholds were determined empirically in previous works. Conclusions are shown in the last section.

2 Our Approach

2.1 An RTE Model Based on Using Similarity

Figure 1 describes our model for Recognizing Textual Entailment based on using similarity. In this model, the sample dataset was random divided into three data subsets, which were called by the training dataset, the refining dataset, and the test dataset.

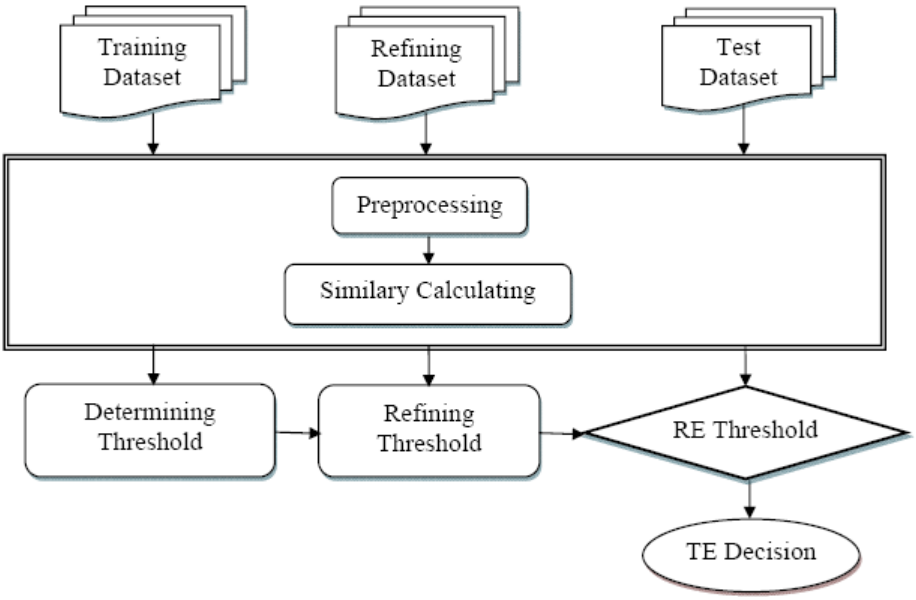


Fig. 1. Model for Textual Entailment Recognizing based on using similarity

The preprocessing is implemented on all of the samples for the presentation of the text T and the hypothesis H.

There are some methods to calculate the similarity of a pair of the text T and the hypothesis H. In this work, we use the lexical similarity described in [3,4] with using a synonym dictionary. In this case, a word appears in the text (the hypothesis) is matched not only with the same word in the hypothesis (the text) but also its synonym words. A semantic similarity of a pair of two synonym words is defined in the subsection 2.3. We also use a combination of above lexical similarity and some similar measures described in [2-5,9,11].

Example 1. An “Entailment" pair (T1, H1) and a “NO Entailment" pair (T2, H2).

T1: <t>Dù sao, bão nhiệt đới lớn băng qua một khoảng cách lớn cũng có thể gây ra những thiệt hại nặng nề.</t> (<t>However, also minor tropical storms passing at relatively great distance can cause severe damage.</t>)

H1: <h>Bão nhiệt đới gây ra những thiệt hại nặng nề.</h> (<h>Tropical storms cause severe damage.</h>)

T2: <t>Năm 1929, ông đã thành lập Viện Toán học (Istituto di Matematica) tại Đại học Milan, cùng với Gian Antonio Maggi và Giulio Vivanti.</t> (<t>In 1929 he founded the Institute of Mathematics (Istituto di Matematica) at the University of Milan, along with Gian Antonio Maggi and Giulio Vivanti.</t>)

H2: <h>Trường Đại học của Milan được thành lập bởi Gian Antonio Maggi và Giulio Vivanti.</h> (<h> The University of Milan is founded by Gian Antonio Maggi and Giulio Vivanti </h>)

The semantic lexical similarity and the combinative similarity of pairs (T1, H1) and (T2, H2) are showed in the Table 1.

Table 1. The semantic lexical similarity (SLS) and the combinative similarity (CS) of pairs (T1, H1) and (T2, H2)

(T,H) pair	SLS	CS
“Entailment" pair (T1, H1)	1.000	3.183E-4
“NO Entailment" pair (T2, H2)	0.667	3.556E-8

The Judgment Threshold for RTE is identified through two steps. In the first step, the training dataset is used for determining the threshold. In the second step, the threshold is refining based on using the refining dataset then we have the RE threshold. The test dataset is used for evaluations the performance of the system.

2.2 Determining and Refining the Judgment Threshold

The judgment threshold plays an important role in RTE systems (see the section 4 below). However, identifying the best judgment threshold is a challenge. Generally, the absolute of words similarity can not greater than 1 then that evidence helps us determining and refining the Judgment threshold.

2.2.1 Determining the Judgment Threshold

The more greater the similarity between hypothesis H and text T is, the more reliable the “Entailment" judgment is. If the threshold is too high (near by 1), some "Entailment" sample pairs may be discarded; on the contrary, if the threshold is too low, some "NOT Entailment" sample pairs may be accepted. The judgment threshold is determined by solving the task for maximizing following function:

$$\sum_{x_i \in Training} y_i * sign(sim(x_i) - thr) \rightarrow \max \quad (1)$$

where

Training be the training dataset,

$\boldsymbol{x_i}$ be a sample pair (the text T_i, the hypothesis H_i) in the training dataset;

$\boldsymbol{y_i}$ be the entailment index respective to x_i. If Judgment is YES then $y_i = 1$, else $y_i = -1$;

sim $\boldsymbol{(x_i)}$ be a similarity score between T_i and H_i. The score is calculated by the method liked methods described in [3,4,11].

Sign be a function which produces 1 with a positive parameter with -1 to negative parameter.

Thr be the judgment threshold.

A procedure of three steps was implemented for solving (1): (i) put all of sim (x_i) in the horizontal axes; (ii) a reversing pointer and a counter is used for the summation in (1). At the initial time, the pointer points at 1 and the counter is set to 0; (iii) Once the pointer meets the sim (x*i*) of a "Entailment" sample then the counter increased by 1, else, in the case of the pointer meets the sim (x_i) of a "No Entailment" sample then the counter decreased by 1. At the end of the procedure, one (or more) interval (s) in which the counter reaches the maximum is (are) found. The judgment threshold may be any value in the interval (s). In the experiments, the middle point of the most left interval was chosen for both cases of determining and refining the judgment threshold.

2.2.2 Refining Judgment Threshold

The procedure to refine the judgment threshold is similar as the procedure to determine the threshold but one modification. Only intervals maximized the determining counter are considered. In the case of the determining counter is higher than the refining counter, the judgment threshold in the determining process is chosen. Figure 2 describes selected intervals of the counters by processes of determining and refining the judgment threshold.

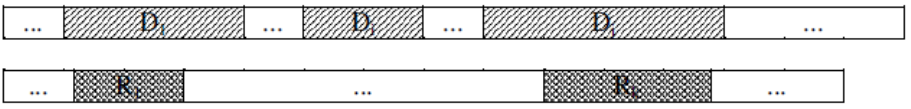


Fig. 2. Selected intervals of the determining counter (above) and the refining counter (below)

2.3 Semantic Similarity Measure of Words in the Synonym Dictionary

In general, the more meaning of two synonym words is, the less similarity score of them is. Assume that (v,w) be a pair of two synonym words in the synonym dictionary, following function is proposed to score the similarity between them:

$$sim(v,w) = \frac{\alpha}{n_v * n_w} \tag{2}$$

where,

n_v and n_w be the number of meanings of v and w, respectively

α be the penalty coefficient.

The value of α is determined by experiments. In this work, α had been set by 1.

For example, the synonym words pair “cha” and “bố” is considered. In the synonym dictionary, the word “bố” synonym with “ba”, “tía”. The word “cha” synonym with “ba”, “bọ”, “bố”, “phụ thân”, “thân phụ”, “thầy”, “tía". Following (2), the similarity score of ("cha", "bố”) is 1/14.

3 Experiments and Results

3.1 The Vietnamese Version of RTE3 Corpus

In this work, we reuse the Vietnamese version of RTE3 corpus, which is described by Minh Quang Nhat Pham et al. [8]. There are TE-labeled 1600 pairs of the text T and the hypothesis H in Vietnamese.

3.2 Experiments

In this work, 10-fold cross-validation experiments had been done. The sample dataset was random divided into 10 subsets. For each fold, two cases of implementations were considered. In the case of no-refining, the current subset was considered as the test dataset and the union of 9 other subsets was considered as the training dataset. In the case of refining, the current subset also was considered as the test dataset, the union of 8 other subsets was considered as the training dataset, and the last subset was considered as the refining dataset.

Table 2. Using semantic lexical similarity only: The Threshold and the measure F1 on 10-fold cross-validation experiments in the no-refining cases (NR) and in the corresponding refining cases (RF)

Exp.	NR		RF	
	Threshold	F1	Threshold	F1
1	-0.2851	0.630	-0.2964	0.653
2	-0.3023	0.623	-0.3098	0.660
3	-0.3127	0.658	-0.3156	0.658
4	-0.1259	0.625	-0.7143	0.639
5	-0.2443	0.657	-0.7432	0.668
6	-0.0881	0.626	-0.7463	0.669
7	-0.4173	0.660	-0.7605	0.675
8	-0.2075	0.648	-0.6935	0.638
9	-0.3152	0.651	-0.6762	0.668
10	-0.1432	0.634	-0.6192	0.663
Arv.		0.640		0.659

We used the F1 measure for evaluations the proposed method.

3.2.1 Using a Lexical Similarity

In this case, semantic lexical similarity had been used. The experiment results is described in the Table 2. The macro average F1 of the no-refining cases reached the value of 0.640 and the macro average F1 of the refining cases reached the value of 0.659. The experimental results are also showed in the Figure 3.

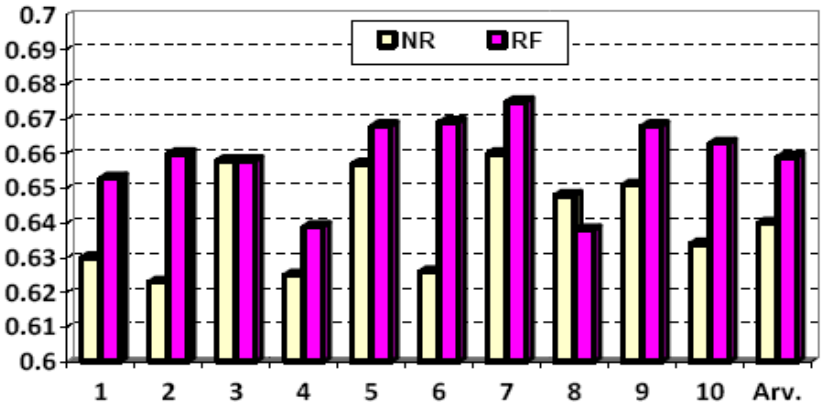


Fig. 3. Using semantic lexical similarity only: The effect of RTE on the F1 measure in the no-refining cases (NR) and in the corresponding refining cases (RF)

Table 2. Using a combinative similarity: The Threshold and the measure F1 on 10-fold cross-validation experiments in the no-refining cases (NR) and in the corresponding refining cases (RF)

Exp.	NR		RF	
	Threshold	F1	Threshold	F1
1	5.91E-8	0.675	5.31E-8	0.678
2	5.82E-8	0.673	5.22E-8	0.676
3	5.56E-8	0.676	3.59E-8	0.666
4	5.11E-8	0.675	3.48E-8	0.664
5	4.41E-8	0.671	5.82E-8	0.674
6	3.59E-8	0.664	5.62E-8	0.677
7	3.52E-8	0.663	5.32E-8	0.678
8	4.41E-8	0.673	5.16E-8	0.676
9	3.27E-8	0.661	3.64E-8	0.665
10	4.34E-8	0.660	3.37E-8	0.664
Arv.		0.669		0.672

3.2.2 Using a Combinative Similarity

A combinative of some similar measures (lexical similarity, cosine similarity, word overlap) and some distant measures (Manhattan, Standard Levenshtein) had been implemented. The experiment results is described in the Table 3. The macro average

F1 of the no-refining cases reached the value of 0.669 and the macro average F1 of the refining cases reached the value of 0.672. The experimental results are also showed in the Figure 4. The experiments show that using the combinative similarity is better than using the lexical similarity only.

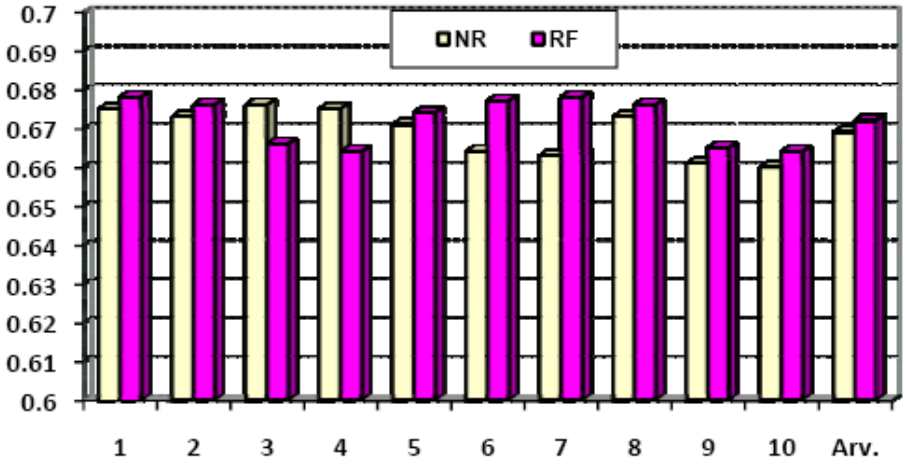


Fig. 4. Using a combinative similarity: The effect of RTE on the F1 measure in the no-refining cases (NR) and in the corresponding refining cases (RF)

4 Related Work

4.1 Recognizing Textual Entailment Using Similarity and the Judgment Threshold

There are some kinds of approaches for RTE such as logic-based approaches, approaches based on using similarity (surface string similarity, syntactic similarity, similarity measures operating on symbolic meaning representations), approaches based on machine learning, approaches based on using vector space models of semantics, and approaches based on decoding [1,2,6,11]. The RTE approaches based on using similarity are very useful. Some RTE methods on the approach have reached the highest performances in TAC - RTE tracks [10,12]. In the RTE approaches, the Judgment Threshold plays a very important role to decide the answer.

Valentin Jijkoun and Maarten de Rijke [3] showed one of the first successful systems for RTE based on using similarity. Firstly, the directed "semantic" word overlap between the text T and the hypothesis H was done on two feature scores, which were the score of the importance of the word for the similarity identification and the score the similarity between two words. The first score was determined by using the normalized inverse collection frequency (ICF) of words with normalizing into values between 0 and 1. The second one, the score of the similarity between two words, was calculated on two similarity measures. Then, the similarity score between

the text T and the hypothesis H had been combined and was compared with a experiment threshold. If the score was greater than the threshold then the answer would be "YES", and in the another case, the the answer would be "NO". In the system, the Judgment Threshold had been determined by experiment. To invest the effective change of the system, the authors set the thresholds changed from 0.1 to 0.9 and they found that there was remarkably different optimal threshold values for the development data (ranged in [0.6-0.7]) and test data (ranged in [0.2-0.4]).

Masaaki Tsuchida and Kai Ishikawa, 2011 [10] proposed an RTE model including two phases. The candidate pairs for "Entailment" was chosen in the first phase, then a SVM classifier was used to discard "Not-Entailment" candidate pairs in the second phase. A pair of the text T and the hypothesis H became into a candidate pairs if its Entailment Score was over the threshold T_{arg} (empirically determined to 0.70). In the second phase, the system detected a threshold of the model for predicting false-positive pairs with the pre-defined precision T_{prec} (empirically determined to 0.80) by using a development set, then discarded candidate pairs if the values predicted by the model for pairs exceeds the predicting threshold. The system's F1 score was the highest in the RTE-7 Challenge [11].

In some cases, high similarity scores do not necessarily guarantee an entailment, thus similarity scores alone must not be used for predicting textual entailment. Kenichi Yokote et al. [12] proposed a similarity based RTE model but the similarity measures had not been used directly. In this model, a set of non-linear transformation functions for similarity measures and the optimal non-linear combination of those transformation functions to predict textual entailment was learned together. Firstly, a word-similarity matrix using a set of similarity measures $S = \{s_1, s_2, \ldots, s_L\}$ for sentences T and H was constructed. After determining the salience vector, the feature vector of sentence pair was calculated by multiplication the salience vector with the word similarity matrix. Secondly, a Joint Learning of Similarity Transformations Functions and RTE predicting was completed. For learning of Similarity Transformation Functions, the parameter vector α was determined. The model also used the threshold θ_i, which was tuned such that the highest micro-averaged F-score had been obtained on the training dataset for each similarity measure s_i in S.

Adrian Iftene [2] proposed an Entailment_Contradiction_Unknown three-way RTE system, which used two thresholds identified on the training dataset. The pairs of the text T and the hypothesis H with the global fitness was under the *Threshold1* (identified to -0.9) would be "Contradictions", other pairs with the global fitness was under the *Threshold2* (ranged from 0.01 to 0.07) would be "Unknown", and the rest pairs would be "Entailment". Rui Wang [11] introduces a three-way RTE system, which was a upgraded version of Adrian Iftene's system. The system also relied on two empirically computed thresholds that separate the high correlation pairs from the low correlation pairs.

4.2 Recognizing Textual Entailment in Vietnamese Language

The Recognizing Textual Entailment in Vietnamese Language is a very fresh topic. Minh Quang Nhat Pham el al. [8] presented two Vietnamese RTE systems based on

Machine Translation components. One system was an English RTE system succeeded the Vietnamese-English Machine Translation component (The front-end system). In the other system, the Vietnamese-English Machine Translation component was integrated in the training phase. In experiments, the authors used a Vietnamese version of RTE3 corpus as the samples. The corpus included 1600 pairs of the text T and the hypothesis H in Vietnamese. They used two baseline systems of simple local lexical matching (LLM) and machine-learning-based system with monolingual features (ML_mono). A machine-learning-based system with bilingual Vietnamese-English features (ML_bi) was chosen as a competitive system. Two front-end systems (ML_mt and EDITS_mt) were also implemented.

Table 4 shows the performance of systems in [8] and in our work by the average F1 measure. In the case of RTE task, our systems were simpler but it showed a little higher performance.

Table 3. Experiment results in [8] and in this work

System		Average F1
Minh et al. [8]	LLM	**0.590**
	ML_mono	0.610
	ML-bi	0.642
	ML_mt	0.639
	EDITS_mt	0.665
Our work	Using the lexical similarity (RF)	**0.659**
	Using the Combinative similarity (RF)	0.672

5 Conclusion

In this paper, we have showed a Vietnamese RTE based on similarity. We also have showed clearly the method to determine and to refine the judgment threshold for improvement RTE. The similarity weigh of synonym words also had been indentified. Experiments on a Vietnamese version of RTE3 corpus showed that the average F1 measure reached the value of 0.659 in the case of using the lexical similarity only and the average F1 measure reached the value of 0.672 in the case of using a combinative similarity. The experiments also showed that the using the combinative similarity is better than the using the lexical similarity only.

The solution integrated our model with combinative similary and the method Joint Learning of Similarity Transformations Functions and RTE predicting in [12] will be investigated.

Acknowledgments. This work was supported in part by Vietnamese MOET Grant B2012-01-24, Polish NCN Grants 2011/01/B/ST6/02759 and 2011/01/B/ST6/027569, and Polish NCBiR Grant SP/I/1/77065/10. Thanks Mr. Pham Quang Nhat Minh for supporting the Vietnamese version of RTE3 corpus.

References

1. Androutsopoulos, I., Malakasiotis, P.: A Survey of Paraphrasing and Textual Entailment Methods. Journal of Artificial Intelligence Research 38, 135–187 (2010)
2. Iftene, A.: Textual Entailment, PhD. Thesis, "Al. I. Cuza" University (Romania) (2009)
3. Jijkoun, V., de Rijke, M.: Recognizing Textual Entailment: Is Word Similarity Enough? In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 449–460. Springer, Heidelberg (2006)
4. Lin, D.: An Information-Theoretic Definition of Similarity. In: ICML 1998, pp. 296–304 (1998)
5. Muramatsu, Y., Uduka, K., Yamamoto, K.: Textual Entailment Recognition using Word Overlap, Mutual Information and Subpath Set. In: Proceedings of the 2nd Workshop on Cognitive Aspects of the Lexicon, pp. 18–27 (2010)
6. Moruz, M.-A.: Predication Driven Textual Entailment, PhD. Thesis, "Al. I. Cuza" University (Romania) (2011)
7. Pérez, D., Alfonseca, E.: Using Bleu-like Algorithms for the Automatic Recognition of Entailment. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 191–204. Springer, Heidelberg (2006)
8. Pham, M.Q.N., Le Nguyen, M., Shimazu, A.: Using Machine Translation for Recognizing Textual Entailment in Vietnamese Language. In: RIVF 2012, pp. 1–6 (2012)
9. Castillo, J.J.: An approach to Recognizing Textual Entailment and TE Search Task using SVM. Procesamiento del Lenguaje Natural (44), marzo de 2010, 139–145 (2010)
10. Tsuchida, M., Ishikawa, K.: IKOMA at TAC2011: A Method for Recognizing Textual Entailment using Lexical-level and Sentence Structure-level features. In: Proceeding of TAC 2011 (2011)
11. Wang, R.: Intrinsic and Extrinsic Approaches to Recognizing Textual Entailment, PhD Thesis, Saarland University (German) (2011)
12. Yokote, K.-I., Bollegala, D., Ishizuka, M.: Similarity is not Entailment-Jointly Learning Similarity Transformations for Textual Entailment. In: Proceedings of the 26th National Conference on Artificial Intelligence (AAAI), pp. 1720–1726 (2012)

Optimization of β-Decision Rules Relative to Number of Misclassifications

Beata Marta Zielosko

Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
beata.zielosko@kaust.edu.sa
Institute of Computer Science, University of Silesia
39, Będzińska St., 41-200 Sosnowiec, Poland

Abstract. In the paper, we present an algorithm for optimization of approximate decision rules relative to the number of misclassifications. The considered algorithm is based on extensions of dynamic programming and constructs a directed acyclic graph $\Delta_\beta(T)$. Based on this graph we can describe the whole set of so-called irredundant β-decision rules. We can optimize rules from this set according to the number of misclassifications. Results of experiments with decision tables from the UCI Machine Learning Repository are presented.

Keywords: approximate decision rules, number of misclassifications, dynamic programming.

1 Introduction

Decision rules are used in many areas connected with data mining and machine learning. Exact decision rules can be overfitted, i.e., dependent essentially on the noise or adjusted too much to the existing examples. If decision rules are considered as a way of knowledge representation then instead of exact decision rules with many attributes, it is more appropriate to work with approximate decision rules which contain smaller number of attributes and have relatively good accuracy [6, 9, 10, 12–14].

There are many approaches to the construction of decision rules, for example, Boolean reasoning [11], Apriori algorithm [1], ant colony optimization [7], genetic algorithms [14], different kind of greedy algorithms [9, 11], dynamic programming [3–5].

Approach based on dynamic programming allows to construct optimal (from different viewpoints) decision rules. In [3] we studied dynamic programming approach for exact decision rule optimization. In [4] we considered optimization of approximate decision rules relative to the number of misclassifications and we used uncertainty measure which is the difference between number of rows in T and number of rows with the most common decision for T. In [5] we studied dynamic programming approach for optimization of β-decision rules relative to

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 345–354, 2012.

length and coverage. In this paper, we present dynamic programming algorithm for optimization of β-decision rules relative to the number of misclassifications.

To work with approximate rules we use an uncertainty measure $R(T)$ that is the number of unordered pairs of rows with different decisions in the decision table T. For a nonnegative real β, we consider β-decision rules that localize rows in subtables of T with uncertainty at most β. The algorithm constructs a directed acyclic graph $\Delta_\beta(T)$. Based on this graph we can describe the whole set of so-called irredundant β-decision rules. We can optimize rules from this set according to the number of misclassifications. This parameter is important for accuracy of classification, so optimization of β-decision rules relative to the number of misclassifications can be considered as tool which supports design of classifiers. To predict value of a decision attribute for a new object we can use in a classifier only rules with the minimum number of misclassifications.

This paper consists of six sections. Section 2 contains definitions of main notions. In Section 3, we present an algorithm for construction of directed acyclic graph which allows to describe the whole set of irredundant β-decision rules. In Section 4, we consider a procedure of optimization of irredundant β-decision rules relative to the number of misclassifications. Section 5 contains results of experiments with decision tables from the UCI Machine Learning Repository [8]. Section 6 contains conclusions.

2 Main Notions

In this section, we consider definitions of notions corresponding to decision tables and decision rules.

A *decision table* T is a rectangular table with n columns labeled with conditional attributes $f_1, \ldots, f_n$. Rows of this table are filled by nonnegative integers which are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer (decision) which is interpreted as value of the decision attribute.

We denote by $N(T)$ the number of rows in the table T. By $R(T)$ we denote the number of unordered pairs of rows with different decisions. We will interpret this value as *uncertainty* of the table T.

A minimum decision value which is attached to the maximum number of rows in T will be called the *most common decision for* T.

Let $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $a_1, \ldots, a_m$ be nonnegative integers. We denote by $T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m)$ the subtable of the table T which contains only rows of T that have numbers $a_1, \ldots, a_m$ at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$. Such subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ the set of attributes from $\{f_1, \ldots, f_n\}$ which are not constant on T, i.e., they have at least two different values. For any $f_i \in E(T)$, we denote by $E(T, f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i_1} = a_1 \wedge \ldots \wedge f_{i_m} = a_m \rightarrow d \tag{1}$$

is called a *decision rule over* T if $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, and $a_1, \ldots a_m, d$ are nonnegative integers. It is possible that $m = 0$. In this case (1) is equal to

$$\rightarrow d \tag{2}$$

Let $r = (b_1, \ldots, b_n)$ be a row of T. We will say that the rule (1) is *realizable for* r, if $a_1 = b_{i_1}, \ldots, a_m = b_{i_m}$. If $m = 0$ then (2) is realizable for any row from T.

Let β be a nonnegative real number. We will say that the rule (1) is *β-true for T* if d is the most common decision for $T' = T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m)$ and $R(T') \leq \beta$. If $m = 0$ then the rule (2) is β-true for T if d is the most common decision for T and $R(T) \leq \beta$.

If the rule (1) is β-true for T and realizable for r, we will say that (1) is a *β-decision rule for T and r*.

We will say that the rule (1) with $m > 0$ is an *irredundant* β-decision rule for T and r if (1) is a β-decision rule for T and r and the following conditions hold:

(i) $f_{i_1} \in E(T)$, and if $m > 1$ then $f_{i_j} \in E(T(f_{i_1}, a_1) \ldots (f_{i_{j-1}}, a_{j-1}))$ for $j = 2, \ldots, m$;
(ii) if $m = 1$ then $R(T) > \beta$, and if $m > 1$ then $R(T(f_{i_1}, a_1) \ldots (f_{i_{m-1}}, a_{m-1})) > \beta$.

If $m = 0$ then the rule (2) is an *irredundant* β-decision rule for T and r if (2) is a β-decision rule for T and r, i.e., if d is the most common decision for T and $R(T) \leq \beta$.

Let τ be a decision rule over T and τ be equal to (1). The *number of misclassifications* of τ is the number of rows in T for which τ is realizable and which are labeled with decisions different from d. We denote it by $\mu(\tau)$. The number of misclassifications of the decision rule (2) is equal to the number of rows in T which are labeled with decisions different from d.

3 Directed Acyclic Graph $\Delta_\beta(T)$

Now, we consider an algorithm that constructs a directed acyclic graph $\Delta_\beta(T)$. Based on this graph we describe the set of irredundant β-decision rules for T and for each row r of T. Nodes of the graph are some separable subtables of the table T. During each step, the algorithm processes one node and marks it with the symbol *. At the first step, the algorithm constructs a graph containing a single node T which is not marked with *.

Let the algorithm have already performed p steps. Let us describe the step $(p+1)$. If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $\Delta_\beta(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet. If $R(\Theta) \leq \beta$ mark the considered node with symbol * and proceed to the step $(p + 2)$. If $R(\Theta) > \beta$, for each $f_i \in E(\Theta)$, draw a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{b_1, \ldots, b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, b_1), \ldots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. We label each row r of Θ with the set of attributes $E_{\Delta_\beta(T)}(\Theta, r) = E(\Theta)$. Mark the node Θ with the symbol * and proceed to the step $(p + 2)$.

The graph $\Delta_\beta(T)$ is a directed acyclic graph. A node of this graph will be called *terminal* if there are no edges leaving this node. Note that a node Θ of $\Delta_\beta(T)$ is terminal if and only if $R(\Theta) \leq \beta$.

Later, we will describe the procedure of optimization of the graph $\Delta_\beta(T)$ relative to the number of misclassifications. As a result we will obtain a graph G with the same sets of nodes and edges as in $\Delta_\beta(T)$. The only difference is that any row r of each nonterminal node Θ of G is labeled with a nonempty set of attributes $E_G(\Theta, r) \subseteq E(\Theta)$. It is possible also that $G = \Delta_\beta(T)$.

Now, for each node Θ of G and for each row r of Θ we describe a set of β-decision rules $Rul_G(\Theta, r)$. We will move from terminal nodes of G to the node T.

Let Θ be a terminal node of G and d be the most common decision for Θ. Then

$$Rul_G(\Theta, r) = \{\rightarrow d\}.$$

Let now Θ be a nonterminal node of G such that for each child Θ' of Θ and for each row r' of Θ', the set of rules $Rul_G(\Theta', r')$ is already defined. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. For any $f_i \in E_G(\Theta, r)$, we define the set of rules $Rul_G(\Theta, r, f_i)$ as follows:

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \gamma \rightarrow s : \gamma \rightarrow s \in Rul_G(\Theta(f_i, b_i), r)\}.$$

Then

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i).$$

Theorem 1. *[3] For each node Θ of the graph $\Delta_\beta(T)$ and for each row r of Θ, the set $Rul_{\Delta_\beta(T)}(\Theta, r)$ is equal to the set of all irredundant β-decision rules for Θ and r.*

To illustrate the algorithm presented above, we consider an example based on decision table T_0 (see Fig. 1). In the example we set $\beta = 2$, so during the construction of the graph $\Delta_2(T_0)$ we stop the partitioning of a subtable Θ of T_0 when $R(\Theta) \leq 2$. We denote $G = \Delta_2(T_0)$.

For each node Θ of the graph G and for each row r of Θ we describe the set $Rul_G(\Theta, r)$. We will move from terminal nodes of G to the node T_0. Terminal nodes of the graph G are Θ_1, Θ_2, Θ_3, Θ_4, Θ_5, Θ_7, Θ_8. For these nodes,

$Rul_G(\Theta_1, r_1) = Rul_G(\Theta_1, r_4) = Rul_G(\Theta_1, r_5) = \{\rightarrow 3\}$;
$Rul_G(\Theta_2, r_2) = Rul_G(\Theta_2, r_3) = \{\rightarrow 1\}$;
$Rul_G(\Theta_3, r_1) = Rul_G(\Theta_3, r_2) = Rul_G(\Theta_3, r_5) = \{\rightarrow 1\}$;
$Rul_G(\Theta_4, r_3) = Rul_G(\Theta_4, r_4) = \{\rightarrow 2\}$;
$Rul_G(\Theta_5, r_5) = \{\rightarrow 3\}$;
$Rul_G(\Theta_7, r_1) = Rul_G(\Theta_7, r_4) = \{\rightarrow 1\}$;
$Rul_G(\Theta_8, r_1) = Rul_G(\Theta_8, r_2) = \{\rightarrow 1\}$.

Now we can describe the sets of rules attached to rows of Θ_6. This is a nonterminal node of G for which all children Θ_2, Θ_4, Θ_7, and Θ_8 are already treated.

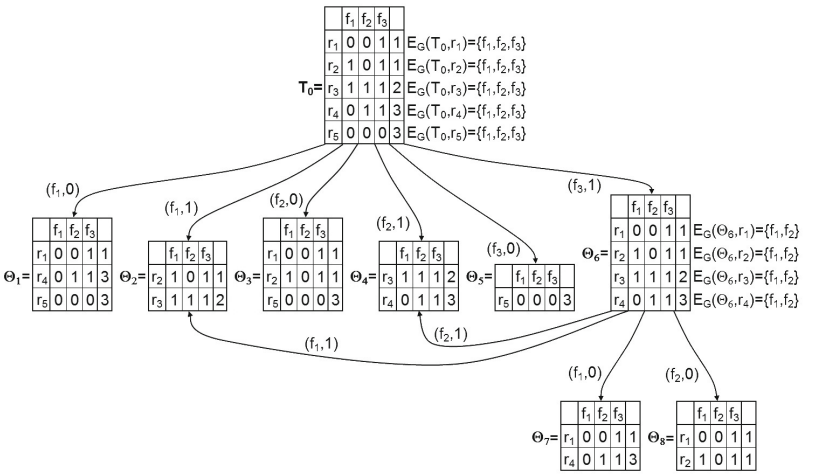


Fig. 1. Directed acyclic graph $G = \Delta_2(T_0)$

$Rul_G(\Theta_6, r_1) = \{f_1 = 0 \to 1, f_2 = 0 \to 1\}$;
$Rul_G(\Theta_6, r_2) = \{f_1 = 1 \to 1, f_2 = 0 \to 1\}$;
$Rul_G(\Theta_6, r_3) = \{f_1 = 1 \to 1, f_2 = 1 \to 2\}$;
$Rul_G(\Theta_6, r_4) = \{f_1 = 0 \to 1, f_2 = 1 \to 2\}$.

Finally, we can describe the sets of rules attached to rows of T_0:

$Rul_G(T_0, r_1) = \{f_1 = 0 \to 3, f_2 = 0 \to 1, f_3 = 1 \wedge f_1 = 0 \to 1,$
$f_3 = 1 \wedge f_2 = 0 \to 1\}$;
$Rul_G(T_0, r_2) = \{f_1 = 1 \to 1, f_2 = 0 \to 1, f_3 = 1 \wedge f_1 = 1 \to 1,$
$f_3 = 1 \wedge f_2 = 0 \to 1\}$;
$Rul_G(T_0, r_3) = \{f_1 = 1 \to 1, f_2 = 1 \to 2, f_3 = 1 \wedge f_1 = 1 \to 1,$
$f_3 = 1 \wedge f_2 = 1 \to 2\}$;
$Rul_G(T_0, r_4) = \{f_1 = 0 \to 3, f_2 = 1 \to 2, f_3 = 1 \wedge f_1 = 0 \to 1,$
$f_3 = 1 \wedge f_2 = 1 \to 2\}$;
$Rul_G(T_0, r_5) = \{f_1 = 0 \to 3, f_2 = 0 \to 1, f_3 = 0 \to 3\}$.

4 Procedure of Optimization Relative to Number of Misclassifications

Let $G = \Delta_\beta(T)$. We consider a procedure of optimization of the graph G relative to the number of misclassifications μ. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^\mu_G(\Theta, r)$ of β-decision rules with the minimum number of misclassifications from $Rul_G(\Theta, r)$ and value $Opt^\mu_G(\Theta, r)$ – the minimum number of misclassifications of a β-decision rule from $Rul_G(\Theta, r)$.

The idea of the procedure is simple. It is clear that for each terminal node Θ of G and for each row r of Θ, the following equalities hold:

$$Rul^{\mu}_{G}(\Theta, r) = Rul_{G}(\Theta, r) = \{\rightarrow d\},$$

where d is the most common decision for Θ, and $Opt^{\mu}_{G}(\Theta, r)$ is equal to the number of rows in Θ labeled with decisions different from d.

Let Θ be a nonterminal node of G, and $r = (b_1, \ldots, b_n)$ be a row of Θ. We know that

$$Rul_{G}(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_{G}(\Theta, r, f_i)$$

and, for $f_i \in E_G(\Theta, r)$,

$$Rul_{G}(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow s : \sigma \rightarrow s \in Rul_{G}(\Theta(f_i, b_i), r)\}.$$

For $f_i \in E_G(\Theta, r)$, we denote by $Rul^{\mu}_{G}(\Theta, r, f_i)$ the set of all β-decision rules with the minimum number of misclassifications from $Rul_{G}(\Theta, r, f_i)$ and by $Opt^{\mu}_{G}(\Theta, r, f_i)$ we denote the minimum number of misclassifications of a β-decision rule from $Rul_{G}(\Theta, r, f_i)$.

One can show that

$$Rul^{\mu}_{G}(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow s : \sigma \rightarrow s \in Rul^{\mu}_{G}(\Theta(f_i, b_i), r)\},$$

$$Opt^{\mu}_{G}(\Theta, r, f_i) = Opt^{\mu}_{G}(\Theta(f_i, b_i), r),$$

and $Opt^{\mu}_{G}(\Theta, r) = \min\{Opt^{\mu}_{G}(\Theta, r, f_i) : f_i \in E_G(\Theta, r)\} = \min\{Opt^{\mu}_{G}(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$. It's easy to see also that

$$Rul^{\mu}_{G}(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r), Opt^{\mu}_{G}(\Theta(f_i, b_i), r) = Opt^{\mu}_{G}(\Theta, r)} Rul^{\mu}_{G}(\Theta, r, f_i).$$

We now describe the procedure of optimization of the graph G relative to the number of misclassifications μ.

We will move from the terminal nodes of the graph G to the node T. We will correspond to each row r of each table Θ the number $Opt^{\mu}_{G}(\Theta, r)$ which is the minimum number of misclassifications of a β-decision rule from $Rul_{G}(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by G^{μ}.

Let Θ be a terminal node of G and d be the most common decision for Θ. Then we correspond to each row r of Θ the number $Opt^{\mu}_{G}(\Theta, r)$ which is equal to the number of rows in Θ which are labeled with decisions different from d.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number $Opt^{\mu}_{G}(\Theta, r) = \min\{Opt^{\mu}_{G}(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$ to the row r in the table Θ, and we set $E_{G^{\mu}}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^{\mu}_{G}(\Theta(f_i, b_i), r) = Opt^{\mu}_{G}(\Theta, r)\}$.

From the reasoning before the description of the procedure of optimization relative to the number of misclassifications (first part of Section 4) the next statement follows.

Theorem 2. *For each node Θ of the graph G^μ and for each row r of Θ, the set $Rul_{G^\mu}(\Theta, r)$ is equal to the set $Rul^\mu_G(\Theta, r)$ of all β-decision rules with the minimum number of misclassifications from the set $Rul_G(\Theta, r)$.*

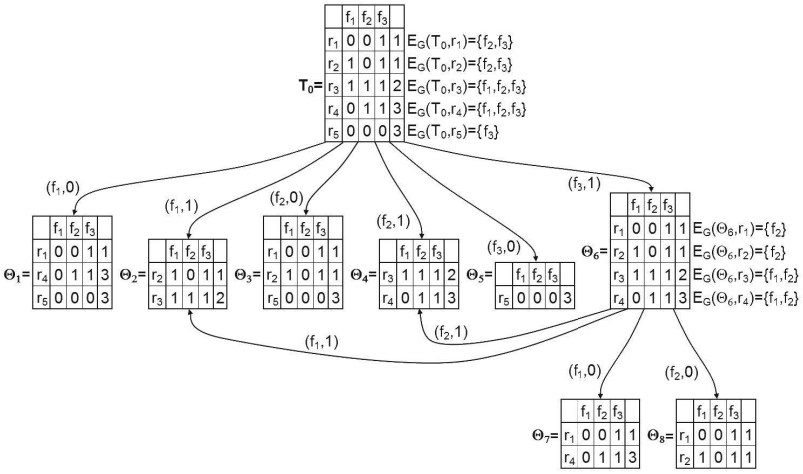


Fig. 2. Graph $G^\mu = \Delta_2(T_0)^\mu$

Figure 2 presents the directed acyclic graph G^μ obtained from the graph G (see Fig. 1) by the procedure of optimization relative to the number of misclassifications.

Using the graph G^μ we can describe for each row r_i, $i = 1, \ldots, 5$, of the table T_0 the set $Rul^\mu_G(T_0, r_i)$ of all irredundant 2-decision rules for T_0 and r_i with minimum number of misclassifications. We will give also the value $Opt^\mu_G(T_0, r_i)$ which is equal to the minimum number of misclassifications of an irredundant 2-decision rule for T_0 and r_i. This value was obtained during the procedure of optimization of the graph G relative to the number of misclassifications. We have:

$Rul^\mu_G(T_0, r_1) = \{f_3 = 1 \wedge f_2 = 0 \to 1\}$, $Opt^\mu_G(T_0, r_1) = 0$;
$Rul^\mu_G(T_0, r_2) = \{f_3 = 1 \wedge f_2 = 0 \to 1\}$, $Opt^\mu_G(T_0, r_2) = 0$;
$Rul^\mu_G(T_0, r_3) = \{f_1 = 1 \to 1, f_2 = 1 \to 2, f_3 = 1 \wedge f_1 = 1 \to 1,$
$f_3 = 1 \wedge f_2 = 1 \to 2\}$, $Opt^\mu_G(T_0, r_3) = 1$;
$Rul^\mu_G(T_0, r_4) = \{f_1 = 0 \to 3, f_2 = 1 \to 2, f_3 = 1 \wedge f_1 = 0 \to 1,$
$f_3 = 1 \wedge f_2 = 1 \to 2\}$, $Opt^\mu_G(T_0, r_4) = 1$;
$Rul^\mu_G(T_0, r_5) = \{f_3 = 0 \to 3\}$, $Opt^\mu_G(T_0, r_5) = 0$.

5 Experimental Results

Experiments were done using software system Dagger [2]. It is implemented in C++ and uses Pthreads and MPI libraries for managing threads and processes

respectively. It runs on a single-processor computer or multiprocessor system with shared memory.

We studied a number of decision tables from the UCI Machine Learning Repository [8]. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. In some tables there were equal rows with, possibly, different decisions. In this case each group of identical rows was replaced with a single row from the group with the most common decision for this group. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute.

Let T be one of these decision tables and values of β are from the set $B(T) = \{R(T) \times 0.01, R(T) \times 0.1, R(T) \times 0.2, R(T) \times 0.3, R(T) \times 0.5\}$.

We studied the minimum number of misclassifications of irredundant β-decision rules. Results can be found in Table 1. For each row r of T, we find the minimum number of misclassifications of an irredundant β-decision rule for T and r. After that, we find for rows of T the minimum number of misclassifications of a decision rule with minimum number of misclassifications (column "min"), the maximum number of misclassifications of such a rule (column "max"), and the average number of misclassifications of rules with minimum number of misclassifications – one for each row (column "avg"). Results presented in Table 1

Table 1. Minimum number of misclassifications of β-decision rules

Name of	$\beta = R(T) \times 0.01$			$\beta = R(T) \times 0.1$			$\beta = R(T) \times 0.2$			$\beta = R(T) \times 0.3$			$\beta = R(T) \times 0.5$		
decision table	min	avg	max	min	avg	max	min	avg	max	min	avg	max	min	avg	max
Adult-stretch	0	0.0	0	0	0.00	0	0	0.0	0	0	0.00	0	0	1.00	4
Balance-scale	0	0.4	4	1	4.2	22	5	15.0	115	15	37.8	116	27	39.2	116
Breast-cancer	0	0.0	0	0	0.2	3	0	0.9	6	0	1.8	10	0	4.2	22
Cars	0	0.8	6	0	17.2	87	0	24.57	95	0	36.5	122	0	110.2	406
Hayes-roth	0	0.0	0	0	0.3	1	0	0.7	3	0	1.6	4	0	5.7	12
Lymphography	0	0.0	0	0	0.0	0	0	0.0	1	0	0.2	3	0	1.3	8
Monks1-test	0	0.0	0	0	2.0	4	0	4.5	9	0	6.0	12	0	36.0	72
Monks1-train	0	0.0	0	0	0.2	1	0	0.6	3	0	1.3	4	0	5.9	20
Monks3-test	0	0.0	0	0	0.0	0	0	3.7	9	0	5.6	12	0	19.3	36
Monks3-train	0	0.0	0	0	0.7	1	0	0.5	3	0	1.3	5	0	4.2	12
Nursery	0	1.6	28	0	72.1	300	0	162.2	646	0	290.0	862	0	764.4	2590
Zoo	0	0.0	0	0	0.0	0	0	0.0	0	0	0.0	0	0	0.2	4

show that the minimum number of misclassifications of β-decision rules is non-decreasing when the value of β is increasing.

We can consider number of rows in decision table T as an upper bound on the number of misclassifications of irredundant β-decision rules. Table 2 presents number of attributes (column "Attr"), number of rows (column "Rows") and real values of misclassifications (column "max" from Table 1) as percentage of number of rows in decision table T, for β from the set $B(T)$. For $\beta = R(T) \times 0.5$, we can find the maximum real values of misclassifications relative to the number of rows for data sets: "Adult-stretch" – 25%, "Cars" – 23.5% and "Nursery" – 20%, and the minimum real values of misclassifications relative to number of rows for data sets: "Lymphography" – 5.4% and "Zoo" – 6.8%. However, only for "Adult-stretch" and "Zoo" real values of misclassifications relative to the number

Table 2. Real values of misclassifications relative to the number of rows in T

Decision table	Attr	Rows	$R(T) \times 0.01$	$R(T) \times 0.1$	$R(T) \times 0.2$	$R(T) \times 0.3$	$R(T) \times 0.5$
Adult-stretch	4	16	0.0	0.0	0.0	0.0	25.0
Balance-scale	4	625	0.6	3.5	18.4	18.6	18.6
Breast-cancer	9	266	0.0	1.1	2.3	3.8	8.3
Cars	6	1728	0.3	5.0	5.5	7.1	23.5
Hayes-roth	4	69	0.0	1.4	4.3	5.8	17.4
Lymphography	18	148	0.0	0.0	0.7	2.0	5.4
Monks1-test	6	432	0.0	0.9	2.1	2.8	16.7
Monks1-train	6	124	0.0	0.8	2.4	3.2	16.1
Monks3-test	6	432	0.0	0.0	2.1	2.8	8.3
Monks3-train	6	122	0.0	0.8	2.5	4.1	9.8
Nursery	8	12960	0.2	2.3	5.0	6.7	20.0
Zoo	16	59	0.0	0.0	0.0	0.0	6.8

of rows are equal to 0 for $\beta \in \{R(T) \times 0.01, R(T) \times 0.1, R(T) \times 0.2, R(T) \times 0.3\}$. It means that the problem of optimization of β-decision rules relative to the number of misclassifications is reasonable.

6 Conclusions

We considered a dynamic programming approach for the optimization of β-decision rules relative to the number of misclassifications. Results of experiments presented in Table 2 show that real value of the minimum number of misclassifications is notably lesser than the number of rows in decision table. Presented procedure of optimization of approximate rules relative to the number of misclassifications can be considered as tool which supports design of classifiers.

In the future we will study accuracy of classifiers based on optimized rules.

Acknowledgment. The author would like to thank you Prof. Mikhail Moshkov and Dr. Igor Chikalov for possibility to use software system Dagger.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, September 12-15, pp. 487–499. Morgan Kaufmann (1994)
2. Alkhalid, A., Amin, T., Chikalov, I., Hussain, S., Moshkov, M., Zielosko, B.: Dagger: A Tool for Analysis and Optimization of Decision Trees and Rules. In: Computational Informatics, Social Factors and New Information Technologies: Hypermedia Perspectives and Avant-Garde Experiences in the Era of Communicability Expansion, pp. 29–39. Blue Herons Editions, Bergamo (2011)
3. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic Programming Approach for Exact Decision Rule Optimization. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems. ISRL, vol. 42, pp. 211–228. Springer, Heidelberg (2013)

4. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Optimization of approximate decision rules relative to number of misclassifications. In: Proc. of the 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems. Springer (to appear, 2012)
5. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach to optimization of approximate decision rules. Information Sciences (submitted)
6. Bazan, J.G., Nguyen, S.H., Nguyen, T.T., Skowron, A., Stepaniuk, J.: Decision rules synthesis for object classification. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 23–57. Physica-Verlag, Heidelberg (1998)
7. Boryczka, U., Kozak, J.: New Algorithms for Generation Decision Trees – Ant-Miner and Its Modifications. In: Abraham, A., Hassanien, A.E., de Leon Ferreira de Carvalho, A.C.P., Snášel, V. (eds.) Foundations of Comput. Intel. (6). SCI, vol. 206, pp. 229–262. Springer, Heidelberg (2009)
8. Frank, A., Asuncion, A.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml
9. Moshkov, M., Piliszczuk, M., Zielosko, B.: Partial Covers, Red. & Dec. Rules in Rough Sets. SCI, vol. 145. Springer, Heidelberg (2008)
10. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning. SCI, vol. 360. Springer, Heidelberg (2011)
11. Nguyen, H.S.: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)
12. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
13. Skowron, A.: Rough sets in KDD. In: Shi, Z., Faltings, B., Musem, M. (eds.) 16th World Computer Congress. Proc. Conf. Intelligent Information Processing, pp. 1–17. Publishing House of Electronic Industry, Beijing (2000)
14. Wróblewski, J.: Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47, 351–360 (2001)

Advance Missing Data Processing for Collaborative Filtering

Nguyen Cong Hoan and Vu Thanh Nguyen

University of Information Technology, Vietnam National University - HoChiMinh City
{hoannc,nguyenvt}@uit.edu.vn

Abstract. Memory-based collaborative filtering (CF) is widely used in the recommendation system based on the similar users or items. But all of these approaches suffer from data sparsity. In many cases, the user-item matrix is quite sparse, which directly leads to inaccurate recommend results. This paper focuses the memory-based collaborative filtering problem on the factor: missing data processing. We propose an advance missing data processing includes two steps: (1) using enhanced CHARM algorithm for mining closed subsets – group of users that share interest in some items, (2) using adjusted Slope One algorithm base on subsets for utilizing not only information of both users and items but also information that fall neither in the user array nor in the item array. After that, we use Pearson Correlation Coefficient algorithm for predicting rating for active user. Finally, the empirical evaluation results reveal that the proposed approach outperforms other state-of-the-art CF algorithms and it is more robust against data sparsity.

Keywords: Recommender, Collaborative Filtering, Sparsity, Missing Data, Slope One, Pearson Correlation Coefficient.

1 Introduction

In competitive environment, website providers need to help clients find interesting products quickly and accurately. In this manner, the modern web should discover through large amounts of dynamic data and show valuable information to the users. In a smaller context, website providers should build a high automation recommendation system by investigating large amounts of data about their clients. The recommender systems are based on a database of user ratings, called CF. And CF complies the assumption: the active user will prefer those items which the similar users prefer.

The algorithms for collaborative recommendations can be grouped into two general classes: memory-based and model-based. And on memory-based group, two type of method are studied: user-based [13, 14, 15, 24] and item-based [2, 3, 12, 19]. They share the same idea that user (item)-based methods find the similar users (items) for an active user. For example, item-based methods first look for some similar items which have similar rating styles with the active user, and then employ the ratings from those similar items to predict the ratings for the active user. On both methods, the step

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 355–364, 2012.

which finds the similar users (items) is very important. To compute the similarity between users or items, two most popular approaches are correlation and cosine-based. The correlation approach includes: Pearson Correlation Coefficient [20] and Vector Space Similarity algorithm [10]. The cosine-based approach includes: [3, 27]. Opposite to memory-based, model-based uses a part of data to generate predefined model. Then the model will be used to predict unknown rating for active user.

Within the user-item matrix, the number of already obtained ratings is usually very small. When the matrix is sparse, it is not enough amounts of needed ratings to be predicted. And this is the fundamental problem of all of approaches. Recently, many algorithms have been proposed to overcome this problem. In [25], the sparsity problem is alleviated by applying associative retrieval framework and related spreading activation algorithms. The dimensionality reduction technique, such as Singular Value Decomposition [4, 7] was used to reduce the dimensionality of sparse ratings matrixes. And by combining the memory-based approaches with model-based approaches, Xue et al. [13] introduced the cluster-based smoothing to solve the data sparsity problem. Although the simulation showed that this approach can achieve better performance than others, the above algorithm encountered difficulties in determining the optimize number of clusters and number of users in cluster. On the other hand, the full-filling all the missing data in the user-item matrix could lead to the negative effect for the recommendation phase.

In this paper, we first use enhanced CHARM algorithm for mining closed subsets. The users which are on the subset will rate for some same items, although the rating value should be different. In other words, the subset includes the users that share interest in some items. Second, we propose an advance missing data processing – Slope One algorithm based on subsets. So that it should exploit the information both from users and items. Additionally, information that falls neither in the user array nor in the item array is taken into account. Combining two sub-steps above is our missing data processing. Naturally, this processing does not full-fill data to user-item matrix. After processing phase, we use Pearson Correlation Coefficient (PCC) algorithm for predicting phase.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 presents some background concepts relating to CHARM algorithm, Slope one algorithm, and four reference schemes. Section 4 outlines the proposed algorithms. Section 5 describes experimental results, followed by a conclusion in Section 6.

2 Related Work

2.1 Memory-Based Schemes

Memory-based schemes are the most popular prediction methods and are widely adopted in commercial collaborative filtering systems [12, 22]. They use a similarity measure to build a prediction. The chosen similarity measure determines the accuracy

of the prediction. The most analyzed examples of memory-based CF include user-based approaches [13, 14, 15, 24] and item-based approaches [2, 12, 19]. User-based approaches predict the ratings of active users based on the ratings of similar users found, and item-based approaches predict the ratings of active users based on the information of similar items computed. In general, the scheme which considers the differences of user rating styles should achieve better performance.

2.2 Model-Based and Hybrid Schemes

There are many model-based schemes. They are inherited from linear algebra (SVD, PCA, or Eigenvectors) [11, 17, 18, 20, 21, 23] or borrowed from artificial intelligence such as Bayesian methods, Latent Classes, and Neural Networks [5, 9, 10] or on clustering [1, 6]. Its building model step is not less important than the computation of similarity measure on memory-based scheme. Model-based schemes try to balance robustness and accuracy of predictions, especially when little data are available. In comparison to memory-based schemes, model-based schemes are typically faster at real time. But they have expensive training and re-training phases. They also cannot cover a diverse user range as the memory-based approaches do [13].

Base on memory-based and model-based schemes, hybrid CF methods have been studied recently [8, 13, 16]. They take the advantages of both memory-based and model-based approaches.

3 Background

3.1 Notation

Let $\boldsymbol{I}$ is set of items in user-item matrix. With a given user, the ratings which are rated are represented by an incomplete array u, called evaluation, and u_i is the rating for an item $i \in I$. The subset of $\boldsymbol{I}$ consisting of all items which are rated in u is $S(u)$. The set of all evaluations in the training set is χ. The number of elements in a set S is $card(S)$. The average of ratings in an evaluation u is denoted $\bar{u}$. For an item $i \in I$ let $S_i(\chi)$ is the set of all evaluations $u \in \chi$ such that they contain item i($i \in S(u)$). Given two evaluations u, v, we define the scalar product $\langle u, v \rangle$ as $\sum_{i \in S(u) \cap S(v)} u_i v_i$. And $P(u)_i$ represent the predicted value of item i for a given user.

3.2 Reference Schemes

To test the new scheme, we use the following reference schemes:

The first is PER USER AVERAGE [27], the most basic prediction scheme. It is given by the equation:

$$P(u) = \bar{u}$$

That is, we predict that a user will rate everything according to that user's average rating.

The second is BIAS FROM MEAN [27], based on the user's average and the average deviation from the user mean for the item in question across all users in the training set. It is given by the equation:

$$P(u)_i = \bar{u} + \frac{1}{card(S_i(\chi))} \sum_{v \in S_i(\chi)} v_i - \bar{v}$$

The third is ADJUSTED COSINE, an item-based approach that is reported to work best [2, 27]. Between two items *i* and *j*, the adjusted cosine similarity measure is given by the equation:

$$sim(i, j) = \frac{\sum_{u \in S_{i,j}(\chi)} (u_i - \bar{u})(u_j - \bar{u})}{\sqrt{\sum_{u \in S_{i,j}(\chi)} (u_i - \bar{u})^2 \sum_{u \in S_{i,j}(\chi)} (u_j - \bar{u})^2}}$$

Based on above measure, the prediction is obtained as a weighted sum of these

$$P(u)_i = \frac{\sum_{j \in S(u)} | sim(i, j) | (\alpha_{i,j} u_j \beta_{i,j})}{\sum_{j \in S(u)} | sim(i, j) |}$$

Where the regression coefficients $\alpha_{i,j}\beta_{i,j}$ are chosen to minimize $\sum_{u \in S_{i,j}(u)} (\alpha_{i,j} u_j \beta_{i,j} - u_i)^2$ with *i* and *j* fixed.

The last is PEARSON, one of the most popular and accurate memory-based schemes [22, 27]. It uses Pearson Correlation Coefficient for computing the similarity as the equation below:

$$Corr(u, w) = \frac{\left\langle u - \bar{u}, w - \bar{w} \right\rangle}{\sqrt{\sum_{i \in S(u) \cap S(w)} (u_i - \bar{u})^2 \sum_{i \in S(u) \cap S(w)} (w_i - \bar{w})^2}}$$

Then, it takes the form of a weighted sum over all users in χ

$$P(u)_i = \bar{u} + \frac{\sum_{v \in S_i(\chi)} \gamma(u, v)(v_i - \bar{v})}{\sum_{v \in S_i(\chi)} | \gamma(u, v) |}$$

With

$$\gamma(u, v) = Corr(u, v) | Corr(u, v) |^{\rho - 1}$$

Where $\rho = 2.5$ and ρ is the Case Amplification power. Pearson's correlation together with Case Amplification is shown to be a reasonably accurate memory-based scheme for CF in [14, 15] though more accurate schemes exist.

3.3 Enhanced CHARM Algorithm

CHARM is an efficient algorithm for mining all frequent closed itemsets. It enumerates closed sets using a dual item set-tidset search tree, using an efficient hybrid search that skips many levels [26].

To apply CHARM algorithm for user-item matrix, we transform it as: CHARM input transactions set is set of items of user-matrix; CHARM input items is set of users which rate items. The transformation is detailed as below:

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	1		3	5	1	
u_2	4	3	5	1	1	3
u_3		4		1	1	1
u_4	5		2		1	5
u_5	2	1	1	1	1	

User-item matrix

Transaction	Items
i_1	u_1, u_2, u_4, u_5
i_2	u_2, u_3, u_5
i_3	u_1, u_2, u_4, u_5
i_4	u_1, u_2, u_3, u_5
i_5	u_1, u_2, u_3, u_4, u_5
i_6	u_2,u_3,u_4

CHARM input format

Let C **= CHARM** (*min_sup*)**,** is a set of all closed itemsets resulted by CHARM algorithm. And *min_sup* is minimum support of the closed itemsets. With c is a closed itemset belong C**,** so f_c is a frequency (or support) of c. And CHARM algorithm indicates that $f_c \geq \min\_\sup$. Recall to the transformation above, so itemset c is a set of users which share interest in some items (following, called sub-userset or cluster)

Of course, the clusters with only one member don't take a role in processing missing data. And to improve the coherence of the clusters, we propose a heuristic h to prune the cluster which the similarity of members is not high. This similarity is threshold by parameterμ.

The heuristic h is given by equation:

$$h(c) = \begin{cases} c, if\ (lengh(c) > 1 \\ and\ Corr(u,v) < \mu) \\ \varnothing, otherwise \end{cases}$$

By applying the heuristic h on the set C, we obtain a set C', with:

$$C' = \{h(c), c \in C\}$$

Re-combining cluster c with user-item matrix, we obtain a sub user-item matrix by deleting the users which does not belong to c. That is, the columns of sub matrix are the same with original matrix and its rows are member of cluster. The ratings of the new matrix are moved corresponding from original matrix. For example, with cluster c = *{u2, u3, u5}*, the sub user-item matrix is detailed as:

Sub user-item matrix

	i_1	i_2	i_3	i_4	i_5	i_6
u_2	4	3	5	1	1	3
u_3		4		1	1	1
u_5	2	1	1	1	1	

We call the sub user-item matrix of sub-userset c is m_c, and the set of m_c is called M, with c belong C'.

$$m_c = \{< u, v >, u \in c, v \in I\}$$

$$M = \{m_c, c \in C'\}$$

3.4 Adjusted Slope-One Algorithm

Slope-one algorithm is introduced and used for directly prediction [27]. In this paper, we proposed an adjusted Slope-one version for processing missing data. Slope-one algorithm is based on the idea: only ratings by users who have rated some common items with the predictive user and only ratings of items that the predictive user has also rated are entered into the prediction of ratings [27]. By which, it discovers information not only from other user (item) which rated the same item (user), but also from data points that fall neither in the user array nor in the item array.

The simplest image of the Slope-one algorithms is in Figure-1 below:

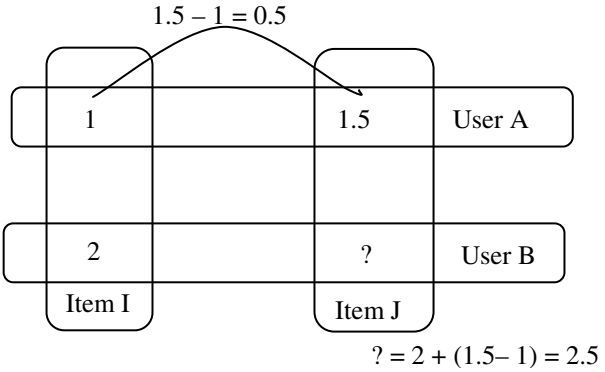


Fig. 1. Unknown rating of User B should be obtained by linking with User A on the sharing rating item [27].

The formal Slope-one is given as equation:

$$P^{S1}(u)_j = \bar{u} + \frac{1}{card(R_j)} \sum_{i \in R_j} dev_{j,i}$$

Where $R_j = \{i \mid i \in S(u), i \neq j, card(S_{j,i}(\chi)) > 0\}$ and

$$dev_{j,i} = \sum_{u \in S_{j,i}(\chi)} \frac{u_j - u_i}{card(S_{j,i}(\chi))}$$

Where $u \in S_{j,i}(\chi)$ mean two items j and i with ratings u_j and u_i respectively in some user evaluation u.

When the number of observed ratings is considered, the Slope-one becomes Weighted Slope-one, is defined by equation:

$$P^{wS1}(u)_j = \frac{\sum_{i \in S(u)-\{j\}} (dev_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u)-\{j\}} c_{j,i}}$$

Where $c_{j,i} = card(S_{j,i}(\chi))$

Instead of using Slope-one with all training set χ, we propose adjusted Slope-one on the *M* space, which is resulted from Enhanced CHARM algorithm above to fill missing data. The value of missing data of items for a given user is defined over filling operator as:

$$P^{filling}(u)_j = \frac{\sum_{c \in C''} P^{wS1}(u)_{jc} (\frac{1 - f_c}{card(c)})}{\sum_{c \in C''} f_c}$$

With C'' contains clusters which the given user is member, $P^{wS1}(u)_{jc}$ is computed on cluster *c* with sub matrix $m_c \in M$.

4 Proposed CF

We summary our proposed CF, call CeSCF, including the following:

Input: user-item matrix, active user
Step 1: using enhanced CHARM to get sub user-item matrix
Step 2: using adjusted Slope-one to fill missing data
Step 3: using Pearson Correlation Coefficient to predict rating for active user
Output: the ratings for the active user

5 Experimental Results

To test our schemes, we have used the Movielens dataset. It is from the Grouplens Research Group at the University of Minnesota. Rating data is collected from movie websites where ratings range from 1 to 5 in increments by 1. This dataset includes 100.000 ratings for 943 users and 1682 items. And its sparsity is about 6% (=*100.000/(943*1682)*). The *min_sup* of CHARM algorithm is double of sparsity, *min_sup = 12%*. Test method is k-fold, with k=10.

The All But One Mean Average Error (MAE) metric have been used to measure the prediction quality of our proposed approach with four reference schemes. To compute the MAE value, we successively hide the current rating u_i and compute the prediction $P(u)_i$ by using recommender schemes.

The average error (MAE) over a test set χ' is given by:

$$MAE = \frac{1}{card(\chi')} \sum_{u \in \chi'} \frac{1}{card(S(u))} \sum_{i \in S(u)} | P(u)_i - u_i |$$

The final result is summarized in the Figure-2.

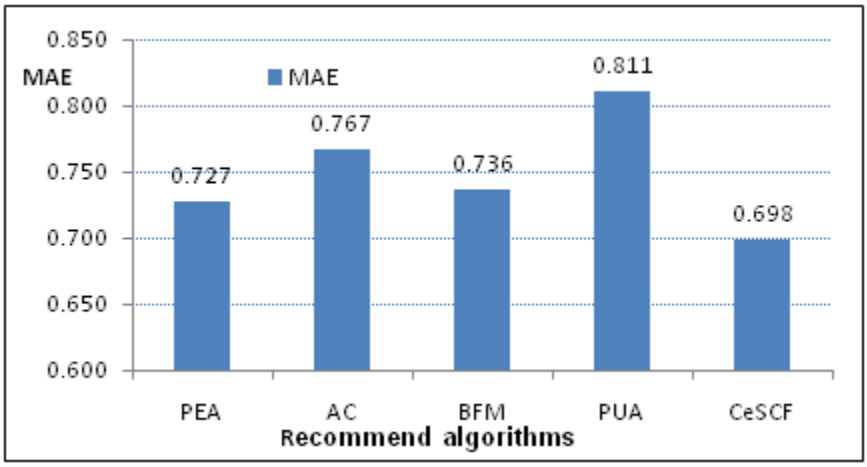


Fig. 2. All schemes compared - MAE lower is better. PUA-Per User Average, BFM-BiasiFromMean, AC-AdjustedCosine, PEA-Pearson)

Compare four references CFs and our CeSCF, our approach achieve accuracy better than the best CF (PEARSON). So that we are reasonable to conclude that: the appropriate missing data processing makes better prediction result and our approach reaches this goal.

The next, we consider the affect of the parameter μ in the heuristic *h* at enhanced CHARM algorithm (3.3). The empirical result is shown on the Figure-3.

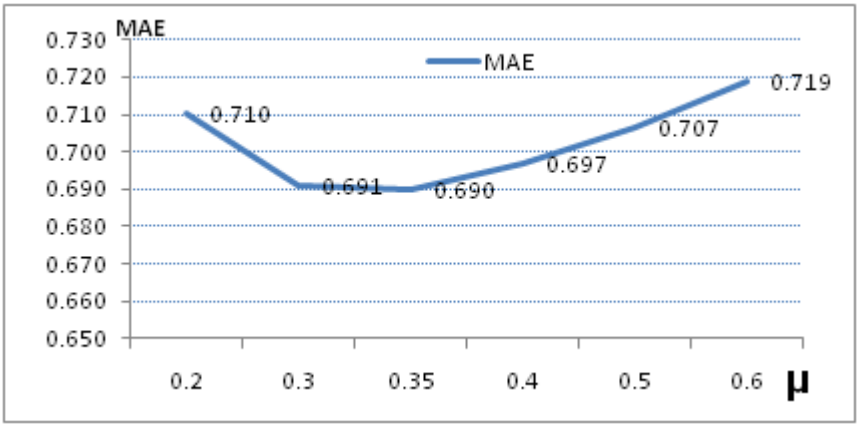


Fig. 3. The impact of μ to the MAE

Figure-3 shows that the quality of the prediction phase is impact byμ: MAE is relatively stable when μ is from 0.3 to 0.45 and increment when μ is on remaining region. It shows that determination which sub userset is retained, impacts to quality of missing data process and over all prediction scheme.

6 Conclusion

In this paper, we propose a missing data processing method for CF. By investigating whether a user (an item) has other similar users (items) and common sharing rating item, our approach determines how to obtain and fill the missing data. The traditional CF works as final step after missing data problem has been processed. The empirical evaluation results show that our proposed method for CF outperforms other state-of-the-art CF approaches.

For future work, we plan to conduct more research on discovering user set that shares interest in some items comparable to enhanced CHARM algorithm. Beside, splitting user set to like and dislike can be examined to improve quality of missing data processing. Lastly, the interested research is scalability analysis for missing data processing phase as well as all algorithms.

References

[1] Ansari, A., Essegaier, S., Kohli, R.: Internet Recommendations Systems. J. Marketing Research, 363–375 (2000)
[2] Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommender algorithms. In: WWW10 (2001)
[3] Lemire, D.: Scale and translation invariant collaborative filtering systems. Information Retrieval 8(1), 129–150 (2005)
[4] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of Dimensionality Reduction in Recommender Systems—A Case Study. In: Proc. ACM WebKDD Workshop (2000)
[5] Basu, C., Hirsh, H., Cohen, W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation, Recommender Systems. Papers from 1998 Workshop, Technical Report WS 98-08. AAAI Press (1998)
[6] Aggarwal, C.C., Wolf, J.L., Wu, K.-L., Yu, P.S.: Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In: Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (August 1999)
[7] Billsus, D., Pazzani, M.: Learning Collaborative Information Filters. In: Proc. Int'l Conf. Machine Learning (1998)
[8] Pennock, D.M., Horvitz, E., Lawrence, S., Giles, C.L.: Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In: Proc. of UAI (2000)
[9] Adomavicius, G., Tuzhilin, A.: Expert-Driven Validation of Rule-Based User Models in Personalization Applications. Data Mining and Knowledge Discovery 5(1 and 2), 33–58 (2001)
[10] Adomavicius, G., Tuzhilin, A.: Multidimensional Recommender Systems: A Data Warehousing Approach. In: Fiege, L., Mühl, G., Wilhelm, U.G. (eds.) WELCOM 2001. LNCS, vol. 2232, pp. 180–192. Springer, Heidelberg (2001)
[11] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. ACM Trans. Information Systems 23(1) (January 2005)
[12] Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 76–80 (January/February 2003)
[13] Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: Proc. of SIGIR 2005 (2005)

[14] Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proc of SIGIR (1999)
[15] Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. of UAI (1998)
[16] Ghazanfar, M., Prugel-Bennett, A.: An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative filtering. IAENG International Journal of Computer Science, 37 (2010)
[17] Armstrong, J.S.: Principles of Forecasting—A Handbook for Researchers and Practitioners. Kluwer Academic (2001)
[18] Canny, J.: Collaborative Filtering With Privacy Via Factor Analysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 238–245 (2002)
[19] Deshpande, M., Karypis, G.: Item-based top-n recommendation. ACM Trans. Inf. Syst. 22(1), 143–177 (2004)
[20] Buhmann, M.D.: Approximation and Interpolation with Radial Functions. In: Dyn, N., Leviatan, D., Levin, D., Pinkus, A. (eds.) Multivariate Approximation and Applications. Cambridge Univ. Press (2001)
[21] Belkin, N., Croft, B.: Information Filtering and Information Retrieval. Comm. ACM 35(12), 29–37 (1992)
[22] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of net news. In: Proc. of ACM Conference on Computer Supported Cooperative Work (1994)
[23] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley (1999)
[24] Jin, R., Chai, J.Y., Si, L.: An automatic weighting scheme for collaborative filtering. In: Proc. of SIGIR (2004)
[25] Huang, Z., Chen, H., Zeng, D.: Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. ACM Trans. Information Systems 22(1), 116–142 (2004)
[26] Zaki, M.J., Hsiao, C.: CHARM: An Efficient Algorithm for Closed. Itemset Mining. In: SDM 2002 (2002)
[27] Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: SIAM Data Mining (2005)

Improving Nearest Neighbor Classification Using Particle Swarm Optimization with Novel Fitness Function

Ali Adeli[1,2], Ahmad Ghorbani-Rad[3], M. Javad Zomorodian[1,4], Mehdi Neshat[5], and Saeed Mozaffari[6]

[1] Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran
[2] Institute of Computer Science, Bojnurd Darolfonoun Technical College, Bojnurd, Iran
[3] Department of Computer Engineering and Information Technology, Qazvin Islamic Azad University, Qazvin, Iran
[4] Institute of Computer Science, Shiraz Bahonar Technical College, Shiraz, Iran
[5] Department of Computer Science, Shirvan Branch, Islamic Azad University, Shirvan, Iran
[6] Department of Electrical and Computer Engineering, Semnan University, Semnan, Iran
{aliadeli,jzomorodian}@shirazu.ac.ir, ahmad.ghorbani@qiau.ac.ir, neshat_mehdi@ieee.org, mozaffari@semnan.ac.ir

Abstract. A new method of feature selection is presented in this paper. The proposed idea uses Particle Swarm Optimization (PSO) with fitness function in order to assign higher weights to informative features while noisy irrelevant features are given low weights. The measure of Area Under the receiver operating characteristics Curve (AUC) is used as the fitness function of the particles. Experimental results claim that the PSO-based feature weighting can improve the classification performance of the k-NN algorithm in comparison with the other important method in realm of feature weighting such as Mutual Information, Genetic Algorithm, Tabu Search and chi-squared (χ^2). Additionally, on synthetic data sets, this method is able to allocate very low weight to the noisy irrelevant features which may be considered as the eliminated features from the data set.

Keywords: AUC, Particle Swarm Intelligence, Feature weighting, Noisy feature elimination, k-NN.

1 Introduction

K-Nearest Neighbor (k-NN) is a well-known classifier which is based on the distance measure. K-NN classifies a new coming instance based on the majority class of its closest training instances. Although the k-NN is considered as a lazy classifier, it is a simple classifier which is applied in various real world applications. In some cases, k-NN shows poor performance because of few instances, noisy data and too many features.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 365–372, 2012.

To improve the performance of this classifier, many solutions are introduced. One of the effective solutions is searching in feature space in order to find optimal subset of features that can improve the classification accuracy of k-NN. It means that the importance of each feature can be defined by assigning a weight to each feature in order to eliminate irrelevant ones from noisy data sets. In this paper, we attack this problem and introduce a new algorithm to deal with. At first, k-NN is employed to assign different weights to all features.

In the realm of feature ranking, some methods are proposed that are described as follows: Weight Adjusted k Nearest Neighbor (WAKNN) which is introduced by Han [4] to overcome the problem of curse of dimensionality. He implemented his idea on the text classification using k-NN. In his work, each attribute takes a weight using the Mutual Information (MI) between each word and the class variable. In the domain of feature weighting, another work refers to Weighted Artificial Immune Recognition System [7]. In this paper, MI is the main algorithm for feature weighting. Note that the weighted attributes were added to the AIRS. Classification is the final step of AIRS algorithm that is performed by k-NN.

Jankowski recommended weighted k Nearest Neighbor (WkNN) idea [5]. In each fold of their algorithm, the initial weights for all features are set to 1. During each fold, the values of the weights are summed (subtracted) with Δ value. If the updated value can improve the accuracy of the k-NN, the new value is replaced with old one for corresponding feature. After each fold, weighting procedure returns a vector of weights. After 10 folds, the algorithm computes a normalized vector which is a summation of 10 vectors.

GAW is a common solution for weighting attributes that is suggested by Tang and Tseng [10]. GAW is based on the Genetic Algorithm (GA) with real representation. In this paper, weighing approach is used to improve the accuracy of Weighted Fuzzy k-NN (WFKNN) classifier. Guvenir and Akkus studied on Weighted Nearest Neighbor Feature Projection (WkNNFP) [3]. In WkNNFP, Single Feature Accuracy (SFA) procedure is utilized for feature weighting. In SFA, weight of each feature is determined according to accuracy which is obtained by considering only this feature.

Tabu Search (TS) is proposed as a weighting method in [9]. In this paper, a Hybrid Tabu Search/K-NN algorithm is proposed to perform both feature selection and feature weighting simultaneously. In other words, k-NN is used each weight set generated by TS. It searches heuristically in a local neighborhood area and moves from a solution to its best admissible neighbor.

The proposed chi-squared (χ^2) Feature Weighting (χ^2 FW) method can be classified as a mutual information approach for assigning features weights [11]. In this sense, the mutual information (the Chi-Squared statistical score) between the values of a feature and the class of the training instances are used to assign feature weights [11]. The algorithm uses Sequential Weighting as the weighting criteria in order to give weights to the features. The weighting criteria ranks features according to their χ^2 scores. In other words, the features having the lowest score have their weights set to 1, those with the second lowest-scored

features have their weight set to 2 and so on. The process goes on until weights are assigned to the highest χ^2 scored features [11]. In the wide range of weighting approach, algorithm processes the usefulness of each feature independently. So, the non-linear interaction between features has been ignored. While in the proposed method, each particle takes into account this interaction This paper is organized as follows: The proposed method is presented in section 2. Sections 3 includes data set, experimental results and discussion, and conclusion is mentioned in the last section of paper.

2 Proposed Method

Improving the classification accuracy of the k-NN algorithm is the aim of this study. To improve the k-NN classification the best solution is that the informative features are given large weights while noisy irrelevant features are given low weights. So a great approach is needed to select the best features easily. For this purpose, first contribution of the paper has focused on the PSO-based feature weighting. Note that the high weights shows that the corresponding feature is informative and relevant, and can help the classifier to achieve high accuracy. Also, low weight means that the feature is irrelevant. PSO is one of the best and popular search tools. PSO as a weighting procedure is a new approach which can classify input instances with informative and relevant feature. The AUC measure as a fitness function is the second contribution of this paper.

First of all, the data set has been split to the unseen data and training sets. Next, the 10-fold cross validation function has been used to validate the k-NN. Then in each fold, the PSO procedure is called. In PSO algorithm, a population of N particles has been produced randomly and the fitness function of population (particles) is computed. Note that each particle introduces a real values in range [0,1] for each dimension. After that, fitness value of each particle is calculated using AUC function. Then evaluated particles are used in evolutionary progress. The cycle of PSO approach will be described later. The evolutionary process has continued until the conditions are satisfied, i.e. variance of fitness value for the best particles is lower than a predefined threshold. After each fold, the training error should be computed with the validation set. For this propose, particles returns a vector (in size of features) with the best real values in range [0,1] (each value refers to corresponding feature). After that, the weighted features are stored to the k-NN algorithm for classification. Note that all weights will be employed in edited version of Minkowski metric [3] to compute the distance between training and testing instances. After 10 folds, best features are given higher weights while the irrelevant ones are given low weights and then they have been used in the k-NN. Finally, the testing error has been computed.

2.1 Particle Swarm Optimization (PSO)

PSO was proposed by Kennedy and Eberhart in 1995 [6]. It is one of evolutionary optimization techniques that are based on swarm intelligence. The idea of PSO

was born from social behavior of animals such as bird folk and fish swarm. In this method, there is a swarm of particles that each of particles is a feasible solution for optimization problem. Every particle searches on problem space in order to move toward final best solution by adjusting its path and moving toward the best personal experience and also the best swarm experience. Suppose that the population size is N. For particle i $(1 \leq i \leq N)$ in D-dimension space, current position is $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and velocity is $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. During optimization process, velocity and position of each particle at each step is updated by (1) and (2):

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 R^1_{i,j}(Pbest_{i,j}(t) - x_{i,j}(t)) + c_2 R^2_{i,j}(Gbest_j(t) - x_{i,j}(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where, $x_{i,j}$ is the component j of particle i, c_1 and c_2 are acceleration coefficients and w is inertia weight that can be a constant number or a positive function [8]. The parameter R is a random number with uniform distribution in interval [0, 1]. $Pbest_i(t)$ is the best position that is found by particle i until time t (the best individual experience of particle i) and $Gbest(t)$ is the best position that until time t is found by whole swarm's members (the best swarm experience). At each iteration, the best individual experience of particle i is given by (3):

$$Pbest_i(t+1) = \begin{Bmatrix} Pbest_i(t) & \text{if } f(x_i(t+1)) \geq f(\text{Pbest}_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(\text{Pbest}_i(t)) \end{Bmatrix} \quad (3)$$

where, $f(x)$ is the fitness value of vector x. The best swarm experience is given by (4):

$$Gbest(t+1) = \arg\min_{p_i} f(Pbest_i(t+1)), \qquad 1 \leq i \leq N \quad (4)$$

2.2 Fitness Function

The fitness function of the population is the AUC measure which is a scalar value, in interval [0,1], to indicate the discriminative power of binary classifiers. So, high value of AUC shows a remarkable performance for binary classification [6]. To calculate the AUC measure, the Receiver Operating Characteristic (ROC) is required. The ROC is a two dimensional curve to evaluate the classification accuracy in the binary class problem [2]. The AUC value is calculated by taking the integral on the area under the ROC curve [12].

After evaluation of features, best weight for each feature is computed. We employ the k-NN with weighted Minkowski distance function which is defined as (5) that consider wi as the weight of ith feature.

$$L_k(a, b) = (\sum_{i=1}^{d} |a_i - b_i|^k)^{1/k} \quad (5)$$

where a_i and b_i are training and testing instances in d dimension, respectively. Parameter k determines the type of distance, where $k = 1$ and $k = 2$ refer to the Manhattan and Euclidean distances, respectively.

Table 1. Data sets is used in this experiment. number of features and the number of samples of each data set is mentioned

Data set	# features	# samples	# class
Glass	10	214	6
Ionosphere	34	351	2
Iris	4	150	3
Hepatitis	19	155	2
Pima	8	760	2
Sonar	60	208	2
Soybean	35	307	19
Vote	16	435	2
WBC	9	699	2

3 Experimental Results

In the testing phase, in order to validate empirical results, 10-fold cross validation is used. After each fold, to calculate the training error of the k-NN classification, the validation set has been used. After 10 fold, the testing error of the k-NN classification has been calculated using the testing set (unseen data). Tables 2 and 3 reported the results of the test. The data sets used for the analysis of the model have been indexed in Table 1. Experimental results were achieved in two types. In the first type, our presented method deals with 9 binary class (multi-class) distribution of data which are mentioned in Table 3. All data sets were chosen from UCI repository [1]. Next type of testing is applied in order to analyze the behavior of the proposed method on generated irrelevant features (Table 2). For the second type, our method tested on the seven synthetic data sets which are randomly generated with some relevant and irrelevant features [11],[12]. All the synthetic data sets contain 500 samples in the binary class distribution. The values of all features (relevant/irrelevant) are randomly picked from distribution in interval [0,1]. In all synthetic data sets, a data point belongs to positive class if the average value of relevant features for this instance is smaller than the threshold; otherwise it belongs to negative class. The threshold is set as the average values of all features in whole data. So, for each data set, the threshold is deterministic.

3.1 Discussion

In this section, the results of the proposed method are compared with some important feature weighting methods such as Mutual Information (MI), Genetic Algorithm (GA), Tabu Search (TS) and chi-squared Feature Weighting (χ^2 FW). Note that the basic classifier used in the all mentioned weighting methods is the k-NN. Experimental results indicate that the proposed method can improve the classification performance of the k-NN. Furthermore, in a number of cases, k-NN classifier with the PSO-based weighting method can perform better than

the simple k-NN (without feature ranking). In Table 2, effecting of weighting method on k-NN classification is presented. For this purpose, some data sets with different number of relevant and irrelevant features are generated. Experimental results indicate that in all cases of generated data sets, the proposed method is more efficient than the simple k-NN without weighting mechanism. In Table 3, the proposed method outperforms the rest on the data sets such as WBC, Glass, Iris, Pima, Sonar and Vote. Note that the both of the proposed idea and GA-based weighting method show equal and the best results on Hepatitis data set. In comparison with the mentioned weighting methods for k-NN classifier, experimental results prove that the feature weighting scheme based on PSO is an impressive solution to improve the accuracy of k-NN classifier.

WAKNN computes the weight of each feature according to value of MI between this feature and class label [4]. The reason of WAKNN's weak performance is that they process the usefulness of each feature independently. So, the nonlinear interaction between features has been ignored. In other words, WAKNN considers the correlation between each feature and the class label independently from other features. In some cases, there are two features which the correlation between each feature and the class label is low, but the correlation between the combination of them as a features subset and the class label is high. However in the proposed method, each particle considers the contribution of all features on the classification problem.

The Table compares the proposed method with GAW which gives weight to features according to GA [10]. In our approach, one of the important advantages is AUC fitness function. In GAW, classification accuracy rate of the test set (known instances which were tested) is employed for fitness function. Comparison between fitness functions of GA and PSO illustrates that the AUC is a dominant function and assign fitness to particle with high confidence because of its statistical property. So, k-NN classifier with the PSO-based weighting algorithm and the AUC fitness function outperforms the k-NN with Genetic-based feature weighting.

The problem of TS is that this kind of search causes the objective function to deteriorate [9]. In other words, it may be fall into local optimum without any reaching to the best set of weights (in task of feature weighting).

Chi-squared Feature Weighting (χ^2 FW) is a weighting method that is calculated according to (6). In (6), i and j are discrete variables which can assume l and c possible values, respectively. n_{ij} and e_{ij} are the observed frequency and the expected frequency, respectively. Similar to WAKNN, the chi-squared method processes the usefulness of each feature independently. So, the nonlinear interaction between features has been ignored. In some cases, we need to analyze the effect of a group of features on the classification problem instead of considering just one feature.

$$\chi_2 = (\sum_{i=1}^{l}\sum_{j=1}^{c}\frac{(n_{ij}-e_{ij})^2}{e_{ij}} \tag{6}$$

Table 2. Empirical results of classic k-NN and the proposed approach when irrelevant features incorporated into the synthetic data sets.

Synthetic data set		k-NN Error rate	
# relevant features	# irrelevant features	Proposed Method	Without Selection
4	6	**20.21±0.64**	25.01±0.67
5	5	**22.86±0.27**	24.54±1.13
5	8	**18.84±0.45**	22.87±1.32
6	4	**23.41±0.61**	26.65±0.63
8	5	**23.25±0.33**	26.98±1.54
10	10	**23.25±0.33**	26.98±1.54

Table 3. Comparison of classification errors between the proposed method and the other weighting methods. The best results on each data set is highlighted in bold face. Note that the last column refers to results of the proposed method. Result of the proposed method on Soybean data set is shown with dash which means the program is not finished in a week or crashed.

Data set	K-NN Classification					
	Without weighting	Weighting based				
		MI	GA	TS	$\chi 2$ FW	Proposed method
Glass	88.43±0.24	92.78±0.25	89.56±0.45	90.40±0.36	90.56±0.48	**93.05±0.12**
Hepatitis	84.51±0.22	81.29±0.36	**87.72±0.16**	84.25±0.54	80.74±0.04	**87.72±0.16**
Ionosphere	89.74±0.38	89.46±0.73	85.48±0.41	**93.8±0.2**	90.35±0.52	91.86±0.36
Iris	93.33±0.43	92.67±0.18	96.84±0.37	96.7±0.42	95.02±0.11	**97.67±0.38**
Pima	69.02±0.10	75.01±0.90	67.36±0.28	74.59±0.20	75.97±0.25	**76.13±0.10**
Sonar	85.03±0.67	95.95±0.43	89.85±0.35	94.20±0.54	92.51±0.30	**98.72±0.09**
Soybean	89.01±0.08	91.55±0.46	**92.08±0.21**	90.78±0.78	89.65±0.11	—
Vote	92.93±0.61	95.64±0.32	92.65±0.18	94.03±0.14	94.52±0.72	**96.61±0.43**
WBC	93.99±0.34	95.56±0.56	94.52±0.21	95.02±0.56	96.63± 0.23	**97.36±0.28**

4 Conclusion

In this paper, a novel method has been introduced for feature weighting. The proposed method of feature weighting is based on PSO. The best particle returns real values in range [0,1] for all features. The weighting of features is based on the fitness of the particle which is calculated using the AUC, a statistical measure to compare classifiers. The experimental results show that the proposed method improves the k-NN classification. In some cases, the proposed method helps the k-NN to result in more accurate classification than some other method in the realm of feature weighting such as MI, TS and GA. Furthermore, on the synthetic data sets, this method is able to allocate very low weight to the noisy irrelevant features which may be considered as the eliminated features from the data set.

References

1. Blake, L., Merz, C.J.: Uci repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html

2. Fawcett, T.: Roc graphs: Notes and practical considerations for researchers. ReCALL 31(HPL-2003-4), 1–38 (2004), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.9777&rep=rep1&type=pdf
3. Altay Güvenir, H., Akkus, A.: Weighted K Neighbor Classification on Feature Projection 1. In: Proc. of the Twelfth International Symposium on Computer and Information Sciences, ISCIS XII, pp. 44–51 (1997)
4. Han, E.-H(S.), Karypis, G., Kumar, V.: Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 53–65. Springer, Heidelberg (2001), http://dl.acm.org/citation.cfm?id=646419.693652
5. Jankowski, N.: Discrete feature weighting selection algorithm. In: Proceedings of the International Joint Conference on Neural Networks, vol. 1, pp. 636–641 (July 2003)
6. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (November/December 1995)
7. Seeker, A., Freitas, A.: Wairs: improving classification accuracy by weighting attributes in the airs classifier. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 3759–3765 (September 2007)
8. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, pp. 69–73 (May 1998)
9. Tahir, M.A., Bouridane, A., Kurugollu, F.: Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier. Pattern Recogn. Lett. 28(4), 438–446 (2007), http://dx.doi.org/10.1016/j.patrec.2006.08.016
10. Tang, P.H., Tseng, M.H.: Medical data mining using bga and rga for weighting of features in fuzzy k-nn classification. In: 2009 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 3070–3075 (July 2009)
11. Vivencio, D., Hruschka, E., Nicoletti, M., dos Santos, E., Galvao, S.: Feature-weighted k-nearest neighbor classifier. In: FOCI 2007, pp. 481–486 (2007)
12. Zomorodian, M.J., Adeli, A., Sinaee, M., Hashemi, S.: Improving Nearest Neighbor Classification by Elimination of Noisy Irrelevant Features. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012, Part II. LNCS (LNAI), vol. 7197, pp. 11–21. Springer, Heidelberg (2012)

Sentiment Classification: A Combination of PMI, SentiWordNet and Fuzzy Function

Anh-Dung Vo and Cheol-Young Ock

Natural Language Processing Lab, School of Computer Engineering
and Information Technology, University of Ulsan, Ulsan, 680-749, Korea
{voanhdung,okcy}@mail.ulsan.ac.kr

Abstract. Discerning a consensus opinion about a product or service is difficult due to the many opinions on the web. To overcome this problem, sentiment classification has been applied as an important approach for evaluation in sentiment mining. Recently, researchers have proposed various approaches for evaluation in sentiment mining by applying several techniques such as unsupervised and machine learning methods. This paper proposes an unsupervised method for classifying the polarity of reviews using a combination of methods including PMI, SentiWordNet and adjusting the phrase score in the case of modification. The experiment results show that the proposed system achieves accuracy ranging from 69.36% for movie reviews to 80.16% for automotive reviews.

Keywords: Opinion Mining, Sentiment Analysis, Sentiment Classification, PMI, SentiWordNet, Modifier, Fuzzy Function.

1 Introduction

Opinion mining or sentiment analysis can be classified into three subtasks: sentiment classification, subjective/objective identification and feature/aspect-based sentiment analysis. Sentiment classification is perhaps the most widely studied topic [1-5]. It classifies an opinion document as expressing a positive or negative opinion. This task is also commonly known as document-level sentiment classification because the whole document is considered the basic information unit. There are two main approaches to sentiment detection: those that are based on machine learning techniques and those that involve semantic analysis techniques.

It is demonstrated that adjectives are good indicators of subjective and evaluative sentences [6-8]. Turney's group applied an unsupervised learning technique based on point-wise mutual information (PMI) [9, 10]. Meanwhile Pang et al. used supervised machine learning methods (SVM, Naïve Bayes) to classify movie reviews [11]. Whitelaw et al. otherwise applied WordNet to construct a lexicon [12]. To automatically determine whether a term is indeed a marker of opinion content, Esuli and Sebastiani introduced SentiWordNet[1] as an enhanced lexical resource for sentiment analysis [13]

[1] http://sentiwordnet.isti.cnr.it/

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 373–382, 2012.

whereas Ohana and Tierney applied SentiWordNet to document-level sentiment classification [14]. Recently, SentiWordNet 3.0 was released with better accuracy than previous versions [15]. SentiStrength, which has been known to extract sentiment strength from informal English text, used a new method to exploit the de facto grammars and spelling styles of cyberspace [16]. Several researchers have showed a lexicon-based approach to extracting sentiment from text. A semantic orientation calculator is applied to the polarity classification task by using a dictionary of words annotated with their semantic orientation, and incorporates intensification and negation [17].

In this paper, we present an unsupervised approach based on phrase sentiment scoring for review sentiment classification. It is an improvement over the point-wise mutual information–information retrieval (PMI-IR) method [9] because it combines information gained from SentiWordNet and manual assignments and adjusts the phrase sentiment score when modification is needed. There are four main steps in this approach: sentiment phrase extraction, PMI application, fuzzy adjustment and summary.

The first step is to extract sentiment phrases from a given review by applying phrase patterns. There are some exceptions for free format reviews.[2] The algorithm not only considers sentiment words (adjectives, adverbs), but also their modifiers. Consequently, both sentiment phrases and their modifiers are extracted. The second step is to use PMI to estimate the semantic orientation by applying PMI-IR. In the third step, we attempt to combine work in new ways based on the PMI-IR method [9]. The main focus of this paper is the use of a combination of methods including PMI, SentiWordNet, manual assignment and fuzzy function. Each sentiment phrase is evaluated using sentiment scores by applying PMI-IR, SentiWordNet and manual assignment scoring. A hybrid score is then calculated using a linear combination of component scores. When modification is needed, a fuzzy function is applied to adjust the sentiment score. The fourth step is to assign the given review to a class, positive or negative, based on its sentiment score.

Our system was tested with reviews from five domains including movies, apparel, automotive, beauty and camera photos. The algorithm achieved an average accuracy of 74.47%, ranging from 69.36% for movie reviews to 80.16% for automotive reviews.

2 A Combination of PMI, SentiWordNet and Fuzzy Function

The overall experimental procedure is illustrated in Fig. 1. For sentiment classification, the proposed system uses a review as input and produces a classification as output. This process involves four steps: sentiment phrase extraction, PMI-IR application, fuzzy adjustment and summary.

2.1 Sentiment Phrase Extraction

This step involves extracting sentiment phrases from a given review by applying POS Tagger and phrase patterns. A sentiment phrase includes sentiment words, context and modifiers.

[2] Free format of review: refers to reviews which are written in free format.

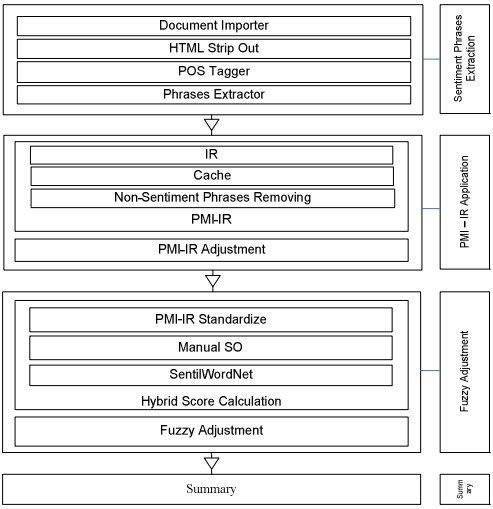


Fig. 1. The proposed system

Preprocessing. A document importer is used to import documents of various file types such as txt and xml. An xml file contains information about the review (product name, product type, star rating, date, reviewer, and content). In this paper we focus on sentiment classification, therefore the content of the review is considered. After the review is imported, HTML Strip Out is applied to remove HTML tags, which are meaningless for sentiment classification.

Sentiment Phrase Extraction. First, the POS Tagger is applied to the review. A phrase is considered a sentiment phrase if it matches one of the defined patterns [9, 18]. Although an isolated adjective may indicate subjectivity, there may be insufficient context to determine semantic orientation. Therefore, a sentiment phrase pattern is defined by two consecutive words where one member of the pair is an adjective or an adverb, and the second provides context. If an adjective does not conform to any of the patterns, it is considered a sentiment phrase without context and will be extracted.

Modifier and Negation Extraction. When reviewers express opinions about a product, they usually use some words (modifiers) to modify the meaning of the sentiment phrase. In every case, the modifier adds information to another element in the sentence. In this paper, we consider modifiers that increase/decrease/invert the meaning of the sentiment phrase. Table 1 describes manually collected modifiers, including "increase" (IN), "decrease" (DE), "invert" (INV), "invert-increase" (II), and "invert-decrease" (ID) modifiers. A modifier II includes an INV modifier followed by an IN modifier. For example, consider the sentence, "It is not a very good camera," shown in Fig. 2. The sentiment phrase "good camera" is modified by the IN modifier "very" and an invert/negation modifier "not". Therefore, this sentence has a II modifier.

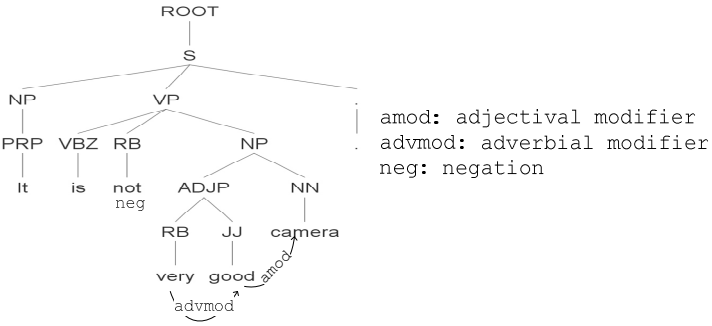


Fig. 2. Sentiment phrase extraction example

Table 1. Modifiers

	Type	Modifier Word
1	IN	very, really, extremely, highly, incredibly, almost, completely, absolutely, totally, too, so
2	DE	quite, bit, little
3	INV	not, never
4	II	INV followed by IN
5	ID	INV followed by DE

2.2 PMI-IR Application

This step estimates the semantic orientation score by applying PMI. Turney showed that a phrase has positive semantic orientation (SO) when it is more strongly associated with a positive reference word ("excellent") and negative when a phrase is more strongly associated with a negative reference word ("poor") (1) [9, 19]. PMI-IR is a method that estimates PMI by issuing queries and determining the number of matching documents (2).

$$\mathrm{SO}(\mathrm{phrase}) = \mathrm{PMI}(\mathrm{phrase}, \text{"exellent"}) - \mathrm{PMI}(\mathrm{phrase}, \text{"poor"}) \tag{1}$$

$$\text{s.t. } \mathrm{PMI}(\mathrm{word}_1, \mathrm{word}_2) = \log_2\left[\frac{\mathrm{p}(\mathrm{word}_1 \,\&\, word_2)}{\mathrm{p}(\mathrm{word}_1)\,\mathrm{p}(\mathrm{word}_2)}\right]$$

$$\mathrm{SO}(\mathrm{phrase}) = \log_2\left[\frac{\mathrm{hits}(\mathrm{phrase} \,\&\, exellent) * \mathrm{hits}(\mathrm{poor})}{\mathrm{hits}(\mathrm{phrase} \,\&\, poor) * \mathrm{hits}(\mathrm{exellent})}\right] \tag{2}$$

2.3 Fuzzy Adjustment

Each extracted phrase includes a sentiment phrase (sentiment word, context) and modifier (Fig. 3). The sentiment scores for each phrase are calculated by applying PMI (second step), SentiWordNet and manual assignment. A hybrid score is then calculated using a linear combination of component scores (3). Finally, a fuzzy function is applied to adjust the hybrid score based on the effect of the modifier in the case of modification (4).

$$\text{Hybrid Score} = \mathrm{w_p} * \mathrm{PMI_s} + \mathrm{w_s} * \mathrm{SWN_s} + \mathrm{w_m} * \mathrm{MA_s} \tag{3}$$

$$\text{Fuzzy Score} = f_{\text{fuzzy function}}(\text{Hybrid Score}) \quad (4)$$

s.t. w_p, w_s, and w_m are the weights of PMI scores (PMIs), SentiWordNet scores (SWNs) and manual assignment scores (MAs). Their sum is 1.0.

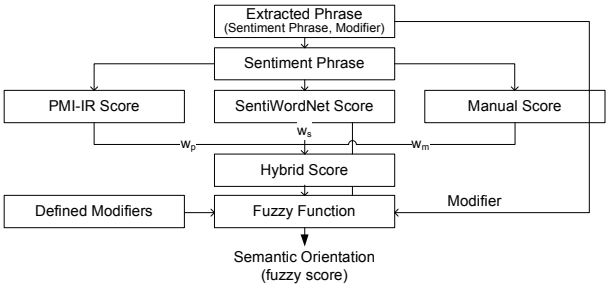


Fig. 3. Sentiment score calculation

SentiWordNet Score. Each synset of WordNet is associated with three numerical scores, Obj(s), Pos(s) and Neg(s), which describe how objective, positive and negative the terms contained in the synset are, respectively.

Manual Assignment Score. The technique for semi-manually collecting sentiment words includes context independence, domain dependence, high-frequency adjectives and English emotional vocabulary. Each item is assigned a semantic orientation in the interval [0.0, 1.0].

Hybrid Score Calculation. The hybrid score is a linear combination of PMI, SentiWordNet and the manual assignment score. The PMI score can be estimated in the second step whereas SentiWordNet and the manual assignment score cannot because the sentiment words cannot be contained in SentiWordNet. Therefore, the hybrid score is calculated using Eq. 5.

$$\text{Hybrid Score} = \begin{cases} w_p*PMI_s + w_s*SWN_S + w_m*MA_s & \text{(SWNs and MAs are both available)} \\ w_p*PMI_s + w_s*SWN_S & \text{(SWNs is available)} \\ w_p*PMI_s + w_m*MA_s & \text{(MAs is available)} \end{cases} \quad (5)$$

Fuzzy Adjustment. This section suggests an approach to address the "modifier," essentially an increase/decrease adjective and negation together with sentiment bearing words in a "fuzzy" way. Therefore, a fuzzy data model that utilizes fuzzy logic and fuzzy sets theory is needed [20, 21]. Fig. 4 shows examples of fuzzy adjustment.

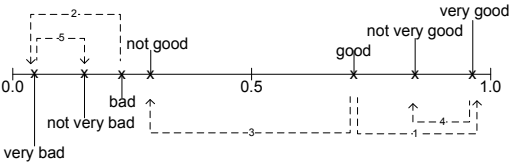


Fig. 4. Fuzzy adjustment example

Assume that "good" is moderately positive and "bad" is moderately negative.
Increase (IN)

(1) "good" $\overrightarrow{IN}$ "very good": very strongly positive

(2) "bad" $\overrightarrow{IN}$ "very bad": very strongly negative

Invert (INV)

(3) "good" $\overrightarrow{IV}$ "not good": moderately negative

Invert-Increase (II)

(1)→(4) "good" $\overrightarrow{II}$ "not very good": strongly positive

(2)→ (5) "bad" $\overrightarrow{II}$ "not very bad": strongly negative

Fuzzy Rules. Fuzzy adjustment can be described as IF THEN rules with the following forms:

IF [(phrase is class) AND (modifier is type)] THEN [adjustment]

class ∈ {positive, negative}: original class (before adjustment)
type ∈{IN, DE, INV, II, ID}
adjustment: increase/decrease sentiment score.

For example, IF [(phrase is positive) AND (modifier is IN)] THEN [increase sentiment score].

Nadali et al. proposed a method based on the combinations of opinion words around each product feature in a review sentence [22]. This methodology determines the strength of opinion orientation (very weak, weak, moderate, very strong and strong) on the product feature using a fuzzy logic technique. Our method is based on phrase sentiment scoring. Therefore, to adjust the sentiment score we use a power function (nonlinear) of the form $f(x) = x^{\alpha}$, where α is a constant real number (system parameter) and x is variable in the interval [0.0, 1.0]. Depending on the value of α, the sentiment score can be decreased with α>1 or increased with α<1. In our experiment, α is set to 1.2 for IN, 0.85 for DE, 0.9 for II and 0.9 for ID. Table 2 describes the fuzzy functions. For example, the first line in Table 2 shows that if a given phrase is negative (x<M) and its modifier is IN (increase), the fuzzy function is selected to be of the form $f(x) = x^{\alpha}$, otherwise $f(x) = x^{1/\alpha}$ (α = 1.2). In the case of a II (invert-increase) modifier, the adjustment includes two steps: increasing and then decreasing. Therefore, the fuzzy function is of the form $f(x) = x<M \ ? \ (x^{\alpha 1})^{\alpha 2} : (x^{1/\alpha 1})^{1/\alpha 2}$ (α1 = 1.2, α2 = 0.9) (Table 2, line 4).

- x: original sentiment score
- M: middle score; If (x>M) then phrase x is positive class, otherwise negative class.
- α1, α1: weight of the modifier indicating how the modifier increases or decreases.
- "?": ternary conditional (condition ? value_if_true : value_if_false).

Table 2. Fuzzy Functions

	Modifier	Fuzzy Function	α_1	α_2
1	II	$f(x) = x<M \ ? \ x^{\alpha 1} : x^{1/\alpha 1}$	1.2	
2	DE	$f(x) = x<M \ ? \ x^{1/\alpha 1} : x^{\alpha 1}$	0.85	
3	INV	$f(x) = 1 - x$		
4	II	$f(x) = x<M \ ? \ (x^{\alpha 1})^{\alpha 2} : (x^{1/\alpha 1})^{1/\alpha 2}$	1.2	0.9
5	ID	$f(x) = x<M \ ? \ (x^{1/\alpha 1})^{\alpha 2} : (x^{\alpha 1})^{1/\alpha 2}$	0.85	0.9

2.4 Summary

This step includes calculating the average sentiment scores of all phrases in the given review. If the average is larger than the middle score, the given review is classified as positive.

3 Experimental Results

Data sets. In order to evaluate the accuracy of our system, we tested it with Movie Review v1.1[3] and the Multi-Domain Sentiment Dataset v2.0[4]. The movie review data is a collection of movie review documents labeled with respect to their overall sentiment polarity. The Multi-Domain reviews contain star ratings that can be converted into binary labels. Table 3 summarizes the details of our experimental data.

Turney applied PMI-IR (Alta Vista[5]) to binary classification with an accuracy of 65.83% for movie reviews [9]. Our system uses Yahoo Search[6] and achieves an accuracy of 54.26% for PMI-IR and 69.36% overall by combining PMI-IR, SentiWordNet and the Fuzzy Function. Turney used reviews from Epinions[7] [9], while our dataset was Movie Review v1.1, thus the datasets are not directly comparable. However, the experimental results show that even when the accuracy of PMI-IR is low, the accuracy of our system with the combination method is better than that from Turney's method [9] (Table 4).

Table 3. Datasets

Domain	No. of Reviews	No. of Extracted Phrases	No. of Phrases with a Modifier
Movie	1400	71716	16064
Apparel	2000	8135	3721
Automotive	736	3365	1227
Beauty	1493	9024	3759
Camera & Photo	1999	15309	5886

Table 4. Comparison with (Turney, 2002)

	(Turney, 2002)	Our System
Domain	Movie	Movie
No. of Reviews	120	1400
PMI-IR Accuracy	65.83%	54.26%
Whole Accuracy	65.83%	69.36%

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[4] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
[5] Alta Vista is now owned by Yahoo.
[6] http://www.search.yahoo.com
[7] http://www.epinions.com

Table 5. Comparison with Related Work

	Applied Methods	Accuracy
(Turney, 2002)	PMI-IR	66% (movie) 84% (automobile)
(Pang et al., 2002)	Supervised Learning	77.1% (SVM, bigrams)
		82.9% (SVM, unigrams)
(Ohana et al., 2009)	SentiWordNet	67.40 (no refinement) 69.35% (refinement)
Our System	Combination of PMI-IR, SentiWordNet and Fuzzy Function	69.36% (movie) 80.16 (automotive)

Pang et al. used the same dataset (Movie Review) for sentiment classification using machine learning and the best performance, obtained using SVM in combination with unigrams, was 82.9% [11]. Meanwhile, Ohana and Tierney applied the SentiWordNet lexical resource and achieved accuracy ranging from 67.40% to 69.35% for film reviews [14]. Table 5 summarizes the comparisons of our method with related work.

We attempt to adjust the PMI-IR score (step 2) by adding a constant c to the original value before the hybrid score is calculated. Figure 5 shows the complex dependence of precision and recall on c in the movie domain. In our system, "excellent" and "poor" are assigned as pairs of reference words. Consequently, the PMI-IR score is estimated by comparing the given phrase to "excellent" vs. "poor." In fact, the number of matching documents returned by a given query differs depending on the number of matching documents returned. The experimental results show that the algorithm has the highest accuracy, 69.36%, with a positive bias, c = 1.6.

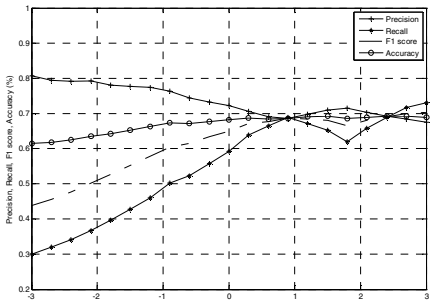


Fig. 5. The dependence of accuracy on c (movie reviews)

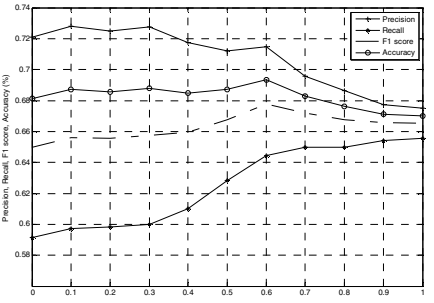


Fig. 6. The dependence of accuracy on β (movie reviews)

Table 6. The Accuracy of the Proposed System

Domain	PMI-IR accuracy	Overall accuracy	Precision	Recall	F1 Score
Movie	54.26%	69.36%	71.47%	64.43%	67.77%
Apparel	61.90%	71.95%	70.80%	74.70%	72.70
Automotive	58.56%	80.16%	81.65%	96.75%	88.56%
Beauty	55.57%	78.03%	79.79%	90.00%	84.59%
Camera, photo	57.38%	72.83%	69.85%	80.40%	74.76%

The hybrid score is calculated based on the combination of PMI, SentiWordNet and assignment score (5). In order to estimate how each measure adds to the knowledge derived from other measures, we assume that

$$Hybrid\ Score = \begin{Bmatrix} \beta * PMI_s + (1-\beta) * SWN_S \\ 0.1 * PMI_s + 0.1 * SWN_S + 0.8 * MA_s \\ 0.2 * PMI_s + 0.8 * Manual_s \end{Bmatrix}$$

Figure 6 illustrates the dependence of accuracy on the value of β in the movie domain. When β is 0, the hybrid score strongly depends on SentiWordNet. Consequently, the accuracy is 68.14%. If β is 1.0, then the hybrid score strongly depends on the PMI. Consequently, the accuracy is 67%. However, the system achieves an accuracy of 69.36% when β = 0.6 by combining SentiWordNet and PMI.

Table 6 summarizes the experimental results in multiple domains. The overall accuracy is achieved by combining PMI-IR, SentiWordNet and Fuzzy Function.

4 Conclusion

This paper presents an unsupervised approach based on phrase sentiment scoring to classify a review as positive or negative. The algorithm performs sentiment classification by combining information gained from PMI-IR, SentiWordNet and manual assignments, and adjusting the phrase sentiment score when modification is needed.

We have shown that a bit complex technique is a useful way to combine information from PMI-IR and SentiWordNet, improving their overall performance compared to the performance of either in isolation. Furthermore, we not only considered sentiment words, but also their modifiers. A fuzzy function was applied to adjust the sentiment score when modification was needed. Nevertheless, parameter choice (the weight of linear combination and fuzzy function parameters) is important to obtain an effective method for sentiment classification. Consequently, an expert is needed to initialize the sentiment lexicon, and the parameters of the proposed fuzzy function should be derived by an expert who has knowledge of how to increase/decrease the modifier. Second, SentiWordNet should be applied at a more advanced level with refinements such as negation detection, linear scoring and feature selection. Last, a better search engine may improve results.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2006906).

References

1. Dave, S.L.K., Pennock, D.M.: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews (2003)
2. Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: Pulse: Mining Customer Opinions from Free Text. In: Famili, A.F., Kok, J.N., Peña, J.M., Siebes, A., Feelders, A. (eds.) IDA 2005. LNCS, vol. 3646, pp. 121–132. Springer, Heidelberg (2005)

3. Das, S.R., Chen, M.Y.: Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. Management Science 53, 1375–1388 (2007)
4. Devitt, A., Ahmad, K.: Sentiment Analysis in Financial News: A Cohesion-based Approach. In: Proceedings of the Association for Computational Linguistics (ACL), pp. 984–991 (2007)
5. Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool (2012)
6. Hatzivassiloglou, V., Wiebe, J.M.: Effects of adjective orientation and gradability on sentence subjectivity. In: Proceedings of the 18th Conference on Computational Linguistics, vol. 12000, pp. 299–305. Association for Computational Linguistics, Saarbrucken (2000)
7. Wiebe, J.: Learning Subjective Adjectives from Corpora. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 735–740. AAAI Press (2000)
8. Wiebe, J., Wilson, T., Bell, M.: Identifying Collocations for Recognizing Opinions. In: Proc. ACL 2001 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation, pp. 24–31 (2001)
9. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics, Philadelphia (2002)
10. Turney, P.D., Littman, M.L.: Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus, p. 11. Information Retrieval (ERB-1094) (2002)
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 102002, pp. 79–86. Association for Computational Linguistics (2002)
12. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 625–631. ACM, Bremen (2005)
13. Esuli, A., Sebastiani, F.: SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In: Proceedings of the 5th Conference on Language Resources and Evaluation, LREC 2006, pp. 417–422 (2006)
14. Ohana, B., Tierney, B.: Sentiment classification of reviews using SentiWordNet. In: 9th IT&T Conference, October 22-23, Dublin Institute of Technology, Dublin (2009)
15. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Calzolari, N., et al. (eds.) LREC. European Language Resources Association (2010)
16. Thelwall, M., et al.: Sentiment in short strength detection informal text. J. Am. Soc. Inf. Sci. Technol. 61, 2544–2558 (2010)
17. Taboada, M., et al.: Lexicon-based methods for sentiment analysis. Comput. Linguist. 37, 267–307 (2011)
18. Santorini, B.: Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision, 2nd printing), Department of Linguistics, University of Pennsylvania, Philadelphia, PA, USA (1990)
19. Turney, P.: Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL (2001)
20. Zadeh, L.: A fuzzy set-theoretic interpretation of linguistic hedges. Journal of Cybernetics 2, 4–34 (1972)
21. Ho, N.C., Nam, H.V.: An algebraic approach to linguistic hedges in Zadeh's fuzzy logic. Fuzzy Sets Syst. 129, 229–254 (2002)
22. Nadali, S., Murad, M.A.A., Kadir, R.A.: Sentiment classification of customer reviews based on fuzzy logic. In: 2010 International Symposium on Information Technology, IT-Sim (2010)

Interestingness Measures for Classification Based on Association Rules

Loan T.T. Nguyen[1], Bay Vo[2], Tzung-Pei Hong[3], and Hoang Chi Thanh[4]

[1] Faculty of Information Technology
VOV Broadcasting College II
Ho Chi Minh, Viet Nam
nguyenthithuyloan@vov.org.vn
[2] Information Technology College
Ho Chi Minh City, Viet Nam
vdbay@itc.edu.vn
[3] Department of CSIE
National University of Kaohsiung
Kaohsiung City, Taiwan, R.O.C
tphong@nuk.edu.tw
Hoang Chi Thanh
[4] Department of Informatics
Ha Noi University of Science
Ha Noi, Viet Nam
thanhhc@vnu.vn

Abstract. This paper proposes a new algorithm for classification based on association rule with interestingness measures. The proposed algorithm uses a tree structure for maintenance of related information in each node, thus making the process of generating rules fast. Besides, the proposed algorithm can be easily extended to integrate some measures together for ranking rules. Experiments are also made to show the efficiency of the proposed approach for different settings. The mining time for different interestingness measures is varied only a little when ten measures are integrated.

Keywords: accuracy, classification, class-association rulem, interestingness measure, integration.

1 Introduction

Class-association Rule (CAR) mining was proposed by Liu et al. in 1998 [8]. It finds classification rules based on association rule mining. Several approaches for mining CARs have then been proposed in recent years such as CPAR [24], CMAR [7], CBA [8], MMAC [16], ACME [17], ECR-CARM [22]. Classifiers based on CARs were shown more accurate than traditional methods such as C4.5 [14] and ILA [18-19] in both the theories [20-12] and experimental results [8].

Interestingness measures play an important role in association rule mining. It can be used for ranking association rules. Tan et al. [15] addressed that there was no measure better than others in all application domains. A suitable interestingness

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 383–392, 2012.

measure can thus be chosen depending on the dataset. Therefore, proposing a general algorithm for mining CARs and their interestingness measure values is an important work in choosing an appropriate measure for a given dataset.

In this paper, we propose an efficient algorithm for mining all CARs along with their measure values for any interestingness measure. The proposed algorithm uses a tree structure for maintenance of related information for fast computing any measure value in each node. Besides, the proposed algorithm can also be extended to integrate multiple interestingness measures. Experimental results also show that the execution time for computing ten measures is nearly the same as that for computing one measure.

2 Related Work

The first method for mining CARs was proposed by Liu et al. in 1998 [8]. It first generated all 1-rule items, where a rule item has the form <*condset*, *y*> with *condset* including a set of items and *y* is a class label. It then generated all candidate 2-ruleitems from the frequent 1-ruleitems and then found the large 2-ruleitems. The same process was then repeated until no more candidates could be obtained. The authors then proposed a heuristic algorithm for building a classifier. Li et al. used the FP-tree structure to speed up the CBA mining process [7]. It scanned a dataset only twice and used the tree structure to compress the dataset. It also used the tree-projection technique to find frequent itemsets. Vo and Le [22] then developed a tree structure called ECR-tree (Equivalence Class Rule-tree) and proposed an algorithm named ECR-CARM for mining CARs. The algorithm was based on the intersection of object identifications to fast compute the support of itemsets, and thus needed to scan the dataset only once. Nguyen et al. proposed a lattice-based approach for pruning efficient redundant rules based on lattices [10].

An interestingness measure is a metric to measure the strength of a rule. Several interestingness measures have been designed for ranking rules in recent years. Depending on domain applications, a suitable measure can be chosen for a given dataset. In the past, Piatetsky-Shapiro [12] applied the statistical independence as an interestingness measure. Several measures for association rules have then been proposed since that. For example, Agrawal and Srikant [1] proposed the support and the confidence measures for mining association rules. Hilderman and Hamilton [3], Tan et al. [15] then compared different interestingness measures. Lee et al. [5] and Omiecinski [11] pointed out that the confidence, coherence and cosine measures were a good effect on mining correlation rules in transaction databases. Tan et al. [15] then discussed the properties of twenty-one objective interestingness measures and analyzed the impact on candidate pruning based on the support threshold. There was no measure better than another in all application domains and some of the measures were correlated to each other [4, 15]. Some studies discussed how to choose appropriate measures for a given database [2, 6, 15]. Vo and Le were also proposed an algorithm for fast mining interesting association rules by combining lattices and hash tables [23].

3 Preliminary Concepts

Let D be a set of training data with n attributes $A_1, A_2, \ldots, A_n$ and $|D|$ objects (cases). Let $C = \{c_1, c_2, \ldots, c_k\}$ be a set of class labels. Some definitions used in the paper are defined below.

Definition 1: An itemset is a set of m attribute-value pairs, denoted $\{(A_{i1}, a_{i1}), (A_{i2}, a_{i2}), \ldots, (A_{im}, a_{im})\}$, where A_{ij} is an attribute and a_{ij} is one of the values of A_{ij}.

Definition 2: A class-association rule r has the form of $\{(A_{i1}, a_{i1}), \ldots, (A_{im}, a_{im})\} \rightarrow c$, where $\{(A_{i1},a_{i1}), \ldots, (A_{im}, a_{im})\}$ is an itemset and $c \in C$ is a class label.

Definition 3: The actual occurrence $ActOcc(r)$ of a rule r in D is the number of records in D that match r's condition.

Definition 4: The support of a rule r, denoted $Sup(r)$, is the number of records in D that match r's condition and belong to the class of r.

For example, consider the rule $r = \{(A, a1)\} \rightarrow y$ for the dataset in Table 1. Since there are three records with the attribute A and two of them belong to the class Y, thus $ActOcc(r) = 3$ and $Sup(r) = 2$.

Table 1. An example of training dataset

OID	*A*	*B*	*C*	*class*
1	*a1*	*b1*	*c1*	*y*
2	*a1*	*b2*	*c1*	*n*
3	*a2*	*b2*	*c1*	*n*
4	*a3*	*b3*	*c1*	*y*
5	*a3*	*b1*	*c2*	*n*
6	*a3*	*b3*	*c1*	*y*
7	*a1*	*b3*	*c2*	*y*
8	*a2*	*b2*	*c2*	*n*

An association rule is an expression form $X \xrightarrow{q,vm} Y$, where $X \cap Y = \varnothing$, $q = Sup(X \cup Y)$, and vm is the value of a measure. For example, in traditional association rules, vm is the confidence of the rule, which is evaluated as $vm = Sup(X \cup Y)/Sup(X)$. Let $vm(n, n_X, n_Y, n_{XY})$ be the measure value of rule $X \rightarrow Y$, where the four variables represent the number of objects in the D, the numbers of objects with X, with Y and with $X \cup Y$, respectively. The measure vm can be computed based on n, n_X, n_Y, and n_{XY}.

For example, consider the rule $\{(A, a1)\} \rightarrow y$ obtained from Table 1. For this rule, $X = \{(A, a1)\}$, $Y = y$, $n = 8$ (number of objects), $n_X = 3$, $n_Y = 4$, and $n_{XY} = 2$. Some extended parameters can also be calculated as $n_{\overline{X}} = 5$, $n_{\overline{Y}} = 4$, and $n_{X\overline{Y}} = 1$. Several measures based on these parameters and their values for the example are listed in Table 2.

Table 2. Some measures and their values for the example

No	Measure	Equation	Value for the example
1	Confidence [1]	$\frac{n_{XY}}{n_X}$	$\frac{2}{3}$
2	Cosine [15]	$\frac{n_{XY}}{\sqrt{n_X n_Y}}$	$\frac{2}{\sqrt{3\times 4}} = \frac{1}{\sqrt{3}}$
3	Lift [13]	$\frac{n_{XY} n}{n_X n_Y}$	$\frac{2\times 8}{3\times 4} = \frac{4}{3}$
4	Rule interest [12]	$\frac{n_X n_{\bar{Y}} - n_{X\bar{Y}} n}{n^2}$	$\frac{3\times 4 - 1\times 8}{64} = \frac{1}{16}$
5	Laplace [15]	$\frac{n_{XY}+1}{n_X+2}$	$\frac{3}{5}$
6	Jaccard [15]	$\frac{n_{XY}}{n_X + n_Y - n_{XY}}$	$\frac{2}{3+4-2} = \frac{2}{5}$
7	Phi-coefficient [15]	$\frac{n_{XY} n - n_X n_Y}{\sqrt{n_X n_Y n_{\bar{X}} n_{\bar{Y}}}}$	$\frac{2\times 8 - 3\times 4}{\sqrt{3\times 4\times 5\times 4}} = \frac{1}{\sqrt{15}}$

4 Mining Class-Association Rules

4.1 MECR-Tree Structure

The MECR-tree structure is modified from the ECR-tree structure [22] for mining CARs with support and confidence measures [22]. In the original ECR-tree structure, all itemsets with the same attributes are clustered into one group. Itemsets in the each group will join with all the itemsets that belong to the other groups following it. . It leads to more time consumption for generating and checking candidates. In the proposed MECR-tree structure, each node in the tree contains an itemset along with the following information:

- *Obidset,* which is a set of objects containing the itemset, and
- $count_i$, which is the number of objects containing the itemset and belonging to class i, for $i \in [1, k]$, where k is the number of classes.

For example, consider the node containing itemset $X = \{(A,a2), (B,b2)\}$ for the dataset in Table 1. Since X is contained in Objects 3 and 8, and both of them belong to class n, a node $\underset{38(0,2)}{\{(A,a2),(B,b2)\}}$ is thus generated in the tree to represent the itemset, where 38 represents Objects 3 and 8, and (0,2) represents 0 (no) object belongs to class y and 2 objects belong to class n. The above representation can be further simplified as $\underset{38(0,2)}{AB\times a2b2}$. In real implementation, the bit presentation is used for storing the attributes of an itemset. For example, the itemset AB can be coded as 11, with the first

bit representing the first attribute *A* and the second bit representing *B*. Therefore, the value of the attributes is 3 and node $\underset{38(0,2)}{AB\times a2b2}$ can be rewritten as $\underset{38(0,2)}{3\times a2b2}$. With this representation, bitwise operations can be used to join itemsets fast. With these descriptions, we devide the itemset into two parts atts and vals: atts is bits representation of attributes containing this itemset and vals is a set of values that belong to this itemset.

Vertex in MECR-tree will connect from node *X* to node *Y* if itemset of *X* is prefix of itemset of *Y*. For example, node $\underset{127(2,1)}{1\times a1}$ will conect to node $\underset{1(1,0)}{3\times a1b1}$ but node $\underset{15(1,1)}{2\times b1}$ will not conect to node $\underset{1(1,0)}{3\times a1b1}$.

4.2 Proposed Algorithm

Input: A dataset and a given interestingness measure *vm;*
Output: CARs and their measure values;
Procedure:

CARIM(*P*, *minSup*)
1. CAR = ∅;
2. for all $l_i \in P$ do
3. CAR = CAR ∪ **ENUMERATE_RULE_IM**(l_i); //find the strongest rule from l_i
4. $P_i = \emptyset$;
5. for all $l_j \in P$, with $j > i$ do
6. if l_i.atts ≠ l_j.atts then
7. *l*.atts = l_i.atts ∪ l_j.atts; //use bitwise operation
8. *l*.vals = l_i.vals ∪ l_j.vals;
9. *l.Obidset* = l_i.*Obidset* ∩ l_j.*Obidset;*
10. if |*l.Obidset*| > 0 then // *l*.itemset exists in the dataset
11. for all *ob* ∈ *O.Obidset* do //compute *O.count*
12. *O.count[ob]++;*
13. $P_i = P_i \cup l$; //add *l* into the set of nodes P_i
14. **CARIM**(P_i, *minSup*); //call recursive to generate all children nodes of P_i

ENUMERATE_RULE_IM(*l*)
15. $CAR_l = \emptyset$;
16. for i ∈ [1,k] do //traverse all classes of the dataset
17. if *l.count*[i] > 0 then// l.itemset contains at least one row belongs to class i
18. n_X = |*l.Obidset*|;
19. n_{XY} = *l.count*[i];
20. n_Y = *Count*[i];
21. $CAR_l = CAR_l \cup \{l.\text{itemset} \rightarrow c_i\ (l.count[i], vm(n, n_X, n_Y, n_{XY})\}$;
22. return The rule with highest information from CAR_l;

Fig. 1. The CARIM algorithm for mining CARs

In this section, an algorithm called CARIM (class-association rule with interestingness measure) is proposed for efficiently mining CARs from a given training dataset. The algorithm is stated in Figure 1. It considers each node l_i with all the other node l_j following l_i in L_r (Lines 2 and 5) to generate a candidate children node l. With each pair (l_i, l_j), the algorithm checks whether l_i.atts ≠ l_j.atts (Line 6). If they are different, it will compute the four elements including *atts, vals, Obidset*, and *count* for the new node l (Lines 7-9). If the number of object identifiers is larger than zero (Line 10), then the algorithm computes the count of the objects in each class that contain l.itemset and adds this node to P_i (P_i is initialized empty in Line 4). Finally, CARIM will be recursively called with a new set P_i as its input parameter (Line 14).

The function ENUMERATE_RULE_IM(l) generates interesting rules from the node l. It first traverses each class (Line 16) to generate rules. If the count of this class is larger than zero (Line 17), it means that l can generate rule from l.itemset → c_i. The function will then compute the parameter values for this rule, including n_X, n_Y and n_{XY} (Lines 18-20), where X is l.itemset and Y is class c_i. To get the support of X, the cardinality of its *Obidset* is counted. The support of Y (*Count*[i]) and n (number of objects) can be obtained when the dataset is scanned. After the four elements are obtained, the value of any measure adopted can be easily calculated (Line 21). Finally, the function will return the rule with the highest measure from the rule set CAR_l (Line 22).

4.3 An Example

The example in Table 1 is used here to describe the process of the algorithm with the Jaccard measure. Figure 2 show the MECR-tree constructed from the dataset in Table 1, where the number before the symbol '×' is bit-presentation of the attributes.

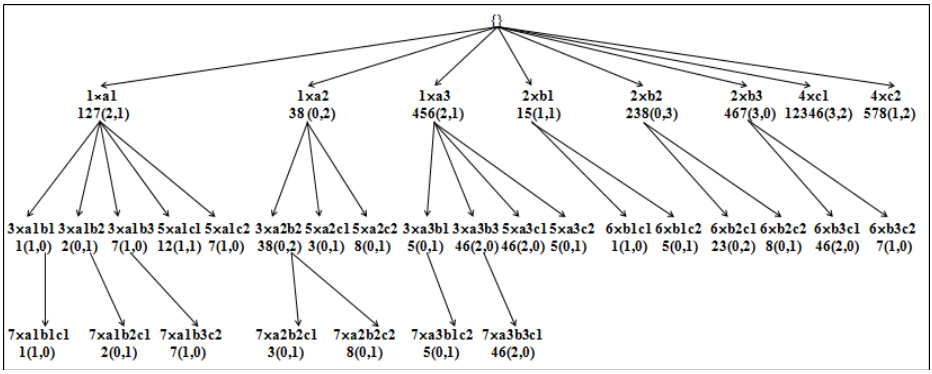


Fig. 2. The MECR-tree constructed from the dataset in Table 1

Consider the process of generated rule from node $\underset{127(2,1)}{1\times a1}$, there are two candidate rules. Consider the first rule "If $A = a1$ then class = y", we have $n_X = 3$, $n_Y = 4$, and $n_{XY} = 2$, the Jaccard measure of this rule is $\frac{2}{3+4-2} = \frac{2}{5}$. Consider the second rule

"If $A = a1$ then class = n" ($n_X = 3$, $n_Y = 4$, and $n_{XY} = 1$), the Jaccard measure of this rule is $\frac{1}{3+4-1} = \frac{1}{6}$. Because 2/5 > 1/6, we choose the first rule for this node.

5 Experimental Results

Experiments were made to show the efficiency of the proposed algorithm. The algorithms used in the experiments were coded on a personal computer with C#2008, Windows 7, Centrino 2×2.53 GHz, and 4MBs RAM. Some datasets obtained from the UCI Machine Learning Repository (http://mlearn.ics.uci.edu) were used in the experiments. Table 3 shows the characteristics of the experimental datasets.

Table 3. The characteristics of the experimental datasets

Dataset	#attributes	#classes	#distinct items	#objects
Breast	12	2	737	699
Lymph	18	4	63	148
Vehicle	19	4	1434	846

Table 4 shows the numbers of rules generated from the datasets in Table 3 for different minimum support thresholds.

Table 4. The numbers of rules generated for different minimum supports

Dataset	*minSup*(%)	#Rules
Breast	1	6016
	0.5	10664
	0.3	15302
	0.1	488356
Lymph	4	1177805
	3	1809130
	2	5783910
	1	14253440
Vehicle	0.8	10645
	0.6	15270
	0.4	41930
	0.2	579970

The results from Table 4 show that some datasets would generate a lot of rules. For example, the Lymph dataset had more than 14 million rules with *minSup* = 1%.

Experiments were then made to compare the execution time of different interestingness measures. The results along with different minimum supports for different datasets are shown in Figures 3 to 5. It could be found from these Figures that the datasets with more numbers of attributes would generate more rules and needed longer execution time.

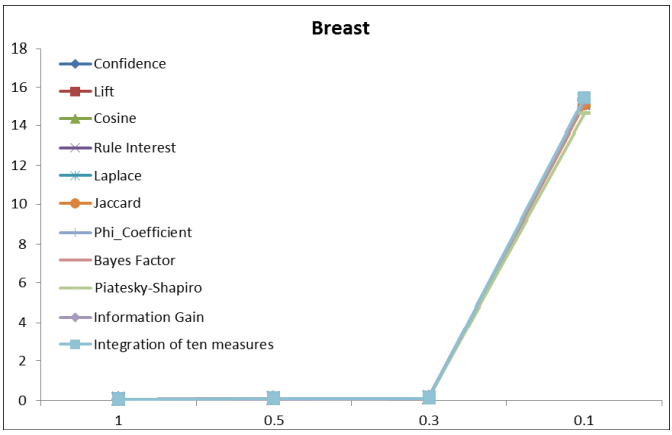


Fig. 3. The execution time of ten different interestingness measures for Breast dataset

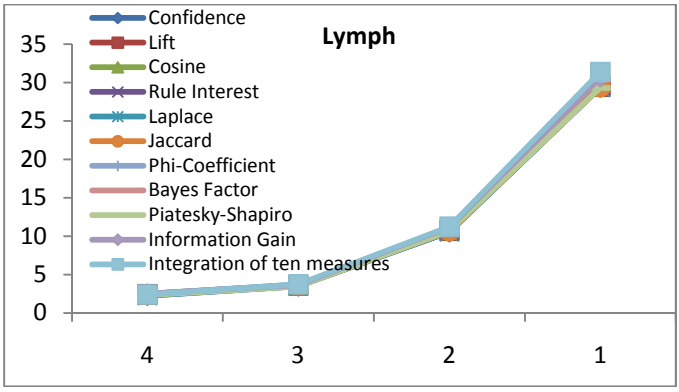


Fig. 4. The execution time of ten different interestingness measures for Lymph dataset

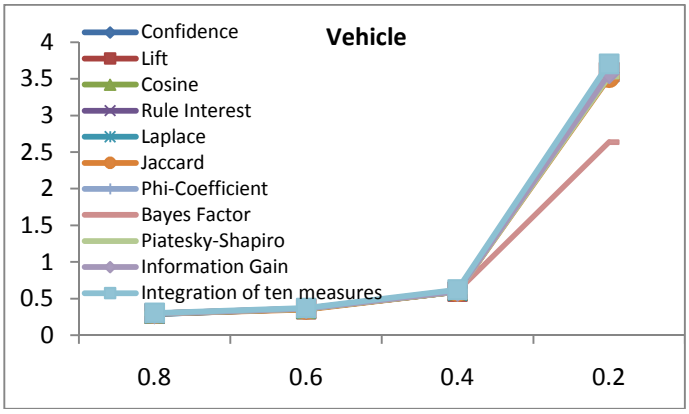


Fig. 5. The execution time of ten different interestingness measures for Vehicle dataset

The experimental results showed that the mining time would increase along with the decrease of the minimum support. Besides, the time for different interestingness measures varied only a little. For example, the minimum execution time for the Breast dataset with *minSup* = 0.1% was 15.1516 seconds while the maximum was 15.4398 seconds. When the ten measures were integrated together, the mining time was 15.4664 seconds.

6 Conclusions and Future Work

This paper has proposed a new algorithm for mining class-association rules with interestingness measures. By using the MECR-tree, the proposed algorithm can mine CARs fast. Furthermore, the proposed algorithm can easily integrate different interestingness measures together for ranking rules. Experimental results show that the execution time for integration of interestingness measures is only slightly more than that for individual measures.

In future, we will study the impact of interestingness measures on accuracy. We will also try to construct other algorithms for ranking rules and building classifiers.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Very Large Data Bases, VLDB 1994, pp. 487–499 (1994)
2. Aljandal, W., Hsu, W.H., Bahirwani, V., Caragea, D., Weninger, T.: Validation-based normalization and selection of interestingness measures for association rules. In: The 18th International Conference on Artificial Neural Networks in Engineering, pp. 1–8 (2008)
3. Hilderman, R., Hamilton, H.: Knowledge discovery and measures of interest. Kluwer Academic Publishers (2001)
4. Huynh, X.-H., Guillet, F., Blanchard, J., Kuntz, P., Briand, H., Gras, R.: A Graph-Based Clustering Approach to Evaluate Interestingness Measures: A Tool and a Comparative Study. In: Guillet, F.J., Hamilton, H.J. (eds.) Quality Measures in Data Mining. SCI, vol. 43, pp. 25–50. Springer, Heidelberg (2007)
5. Lee, Y.K., Kim, W.Y., Cai, Y., Han, J.: CoMine: efficient mining of correlated patterns. In: IEEE International Conference on Data Mining, pp. 581–584 (2003)
6. Lenca, P., Meyer, P., Vaillant, P., Lallich, S.: On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. European Journal of Operational Research 184(2), 610–626 (2008)
7. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: The 1st IEEE International Conference on Data Mining, San Jose, California, USA, pp. 369–376 (2001)
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: The 4th International Conference on Knowledge Discovery and Data Mining, New York, USA, pp. 80–86 (1998)
9. Liu, Y.Z., Jiang, Y.C., Liu, X., Yang, S.L.: CSMC: A combination strategy for multiclass classification based on multiple association rules. Knowledge-Based Systems 21(8), 786–793 (2008)

10. Nguyen, L.T.T., Vo, B., Hong, T.P., Thanh, H.C.: Classification based on association rules: A lattice-based approach. Expert Systems with Applications 39(13), 11357–11366 (2012)
11. Omiecinski, E.: Alternative interest measures for mining associations in databases. IEEE Transaction on Knowledge and Data Engineering 15(1), 57–69 (2003)
12. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248 (1991)
13. Piatetsky-Shapiro, G., Steingold, S.: Measuring lift quality in database marketing. SIGKDD Explorations 2(2), 76–80 (2000)
14. Quinlan, J.R.: C4.5: program for machine learning. Morgan Kaufmann (1992)
15. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Information Systems 29(4), 293–313 (2004)
16. Thabtah, F., Cowling, P., Peng, Y.: MMAC: A new multi-class, multi-label associative classification approach. In: The 4th IEEE International Conference on Data Mining, Brighton, UK, pp. 217–224 (2004)
17. Thonangi, R., Pudi, V.: ACME: An Associative Classifier Based on Maximum Entropy Principle. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 122–134. Springer, Heidelberg (2005)
18. Tolun, M.R., Abu-Soud, S.M.: ILA: An inductive learning algorithm for production rule discovery. Expert Systems with Applications 14(3), 361–370 (1998)
19. Tolun, M.R., Sever, H., Uludag, M., Abu-Soud, S.M.: ILA-2: An inductive learning algorithm for knowledge discovery. Cybernetics and Systems 30(7), 609–628 (1999)
20. Veloso, A., Meira Jr., W., Zaki, M.J.: Lazy associative classification. In: IEEE International Conference on Data Mining, ICDM 2006, Hong Kong, China, pp. 645–654 (2006)
21. Veloso, A., Meira Jr., W., Goncalves, M., Almeida, H.M., Zaki, M.J.: Calibrated lazy associative classification. Information Sciences 181(13), 2656–2670 (2011)
22. Vo, B., Le, B.: A Novel Classification Algorithm Based on Association Rules Mining. In: Richards, D., Kang, B.-H. (eds.) PKAW 2008. LNCS (LNAI), vol. 5465, pp. 61–75. Springer, Heidelberg (2009)
23. Vo, B., Le, B.: Interestingness measures for association rules: Combination between lattice and hash tables. Expert Systems with Applications 38(9), 11630–11640 (2011)
24. Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: SIAM International Conference on Data Mining, SDM 2003, San Francisco, CA, USA, pp. 331–335 (2003)

MSGPs: A Novel Algorithm for Mining Sequential Generator Patterns

Thi-Thiet Pham[1,2], Jiawei Luo[1,*], Tzung-Pei Hong [3,4] , and Bay Vo[5]

[1] School of Information Science and Engineering, Hunan University, Yuelu District, Changsha City, Hunan Province, 410082, Republic of China.
luojiawei@hnu.edu.cn
[2] Faculty of Information Technology, Ho Chi Minh City University of Industry, Ho Chi Minh City, Vietnam
phamthithiet@hui.edu.vn
[3] Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, 811, Taiwan, R.O.C.
tphong@nuk.edu.tw
[4] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C.
[5] Information Technology College, Ho Chi Minh, Vietnam
vdbay@itc.edu.vn

Abstract. Sequential generator pattern mining is an important task in data mining. Sequential generator patterns used together with closed sequential patterns can provide additional information that closed sequential patterns alone are not able to provide. In this paper, we proposed an algorithm called MSGPs, which based on the characteristics of sequential generator patterns and sequence extensions by doing depth-first search on the prefix tree, to find all of the sequential generator patterns. This algorithm uses a vertical approach to listing and counting the support, based on the prime block encoding approach of the prime factorization theory to represent candidate sequences and determine the frequency for each candidate. Experimental results showed that the proposed algorithm is effective.

Keywords: sequential pattern, sequential generator pattern, sequence database, prefix tree.

1 Introduction

Sequential pattern mining, which was first proposed by Agrawal in [1], has played an important role in data mining task The algorithms for mining sequential patterns [2,5,11,12,17] had good performance in databases with short frequent sequences. However, when mining long frequent sequences containing a combinatorial number of frequent subsequences, such mining will generate an explosive number of frequent subsequences for long patterns, or when using very low support thresholds to mine sequential patterns, that is costly in both time and space. So, the performance of such

* Corresponding author.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 393–401, 2012.

algorithms is significantly reduced. To overcome this problem, recent years, mining closed sequential patterns, sequential generator patterns or maximum sequences have been proposed. A sequential pattern *S* is called maximum sequential pattern if there do not exist its super sequences which are frequent. MSPS algorithm [9] could mine maximum sequential patterns effectively. A sequential pattern *S* is called closed if there do not exist super sequences of *S* which have the same support in the database. Several studies [3,6,13-15] were recently proposed to mine closed sequential patterns. In a sequence database, the sequential generator pattern relates to patterns without any subsequence with the same support. Sequential generator patterns used together with closed sequential patterns can provide additional information that closed sequential patterns alone are not able to provide. Li *et al* [7] showed that sequential generator pattern is the minimal member and preferable over all sequential patterns and closed sequential patterns for association rule presentation, web page and product review classification because the length of sequential generator patterns are shorter than closed sequential patterns, so it is necessary to find sequential generator patterns for mining sequential rules from them to reduce redundancies. Recent years, several sequential generator mining methods [4,8,15-16] had been proposed. However, in our opinion, these algorithms have been proposed to generate the different types of pattern separately that is sequential generator patterns can only be generated after sequential patterns found. In this paper, based on the characteristics of sequential generator patterns mentioned in section 4.1, we propose an efficient algorithm called MSGPs for **M**ining all of **S**equential **G**enerator **P**attern**s** at the process of generating sequential patterns. Experimental results show that our proposed algorithm outperforms better than existing algorithms as FSGP [16].

The rest of this paper is organized as follows. Section 2 discusses related works to mining sequential generator pattern. Section 3 presents some problem definitions related to mining sequential generator patterns. Section 4 discusses characteristics of sequential generator patterns and proposes an algorithm for mining sequential generator. Section 5 presents experimental results, and section 6 discusses conclusions and future work.

2 Related Work

Recent years, the problem of mining sequential generator patterns, closed sequential patterns and maximum sequences were proposed to solve the difficulties of mining sequential patterns problem when mining long frequent sequences or when using very low support thresholds. Sequential generator patterns used together with closed sequential patterns can provide additional information that closed sequential patterns alone are not able to provide. Several studies were recently proposed to mine closed sequential patterns such as CloSpan [15], BIDE [13-14], IMCS [3], FCSM-PD ([6] and so on.

In 2008, Lo *et al* [8] proposed the GenMiner method for mining sequential generator patterns, which was the first sequential generator mining algorithm. The GenMiner method extracted sequential generator in a three-step compact-generate-and-filter approach. The first step, it traversed all the sequential patterns and produced a compact representation of the space of frequent patterns in a lattice format in [15].

The second step, it retrieved a set of candidate generators which was a super-set of all generators from the compact lattice and pruned the sub-search spaces containing non-generators by using the unique characteristics of sequential generators [8] to ensure that the candidate generator set is not too large. In the final step, all non-generators from the candidate set were filtered away. The FEAT algorithm was introduced by Gao *et al* [4]. The FEAT was based on sequential patterns growth with forward and backward pruning strategy, along with sequential generator checking technique to speed up the mining process. However, it was costly enormous time for pruning the non-generator sequences which should be pruned since it caused many useless the database projection operations and the comparison of the projected databases in the pruning strategy. To avoid the costly enormous time for pruning, the FSGP algorithm [16] was proposed. In FSGP, a safe pruning strategy was given on the basis of the inclusion relationship between a sequence and its subsequence. Each valid frequent sequential pattern was checked by the sequential generator checking theorem from the set of the valid frequent sequential patterns, then the non-generators were removed, and the result set of the sequential generators was generated. However, algorithms have been proposed to mine sequential generator patterns as introduced in the above that generated the different types of pattern separately is that sequential generator patterns can only be generated after sequential patterns found. In this paper, we propose an efficient algorithm called MSGPs for mining all of sequential generator patterns at the process of generating sequential patterns. This algorithm uses a vertical approach for enumeration and support counting, based on the novel notion of prime block encoding [5], which in turn is based on prime factorization theory. The MSGPs algorithm also applies both the super sequence frequency-based pruning and non-generator-based pruning to reduce much more search space than other algorithms.

3 Problem Definitions

Given a set of items $I=\{i_1, i_2, \ldots, i_n\}$ and a sequence database *SD* contains a set of sequences $S=\{s_1, s_2, \ldots, s_m\}$, where each sequence s_x is an ordered list of itemsets $s_x=\{x_1, x_2, \ldots, x_n\}$, and x_1 occurs before x_2, which occurs before x_3, and so on, such that $x_1, x_2, \ldots, x_n \subseteq I$. The size of a sequence is the number of itemsets in the sequence. The length of a sequence is the number of instances of items in the sequence. A sequence has length l called an *l-pattern*. For example, the length of sequence $\langle ababd \rangle$ is 5 and called 5-pattern.

Sequences $\alpha = \langle \alpha_1\ \alpha_2 \ldots \alpha_n \rangle$ is called a subsequence of $\beta = \langle \beta_1\ \beta_2 \ldots \beta_m \rangle$ and β is a supersequence of α *(where* α_i, β_j *are itemsets)*, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ $(n \leq m)$ such that $\alpha_1 \subseteq \beta_{j1}$, $\alpha_2 \subseteq \beta_{j2}$, . . . , $\alpha_n \subseteq \beta_{jn}$. For example, if $\alpha = \langle (ab), d \rangle$ and $\beta = \langle (abc), (de) \rangle$, where *a, b, c, d*, and *e* are items, then α is a subsequence of β and β is a supersequence of α. The support of a sequence α *sup(*α*)* is the number of sequences in the database containing α. Sequence α is a frequent sequence in a sequence database *SD* if the support *sup(*α*)* of the sequence $\alpha \geq$ *minSup* (minimum support threshold which is a user-specified value). A frequent sequence is called a sequential pattern. Sequential pattern α is called a sequential generator pattern *SGP(*α*)* if and only if $\neg\exists\, \beta$ such that $\alpha \supset \beta$ (i.e., α contains β) and *sup(*α*) = sup(*β*).*

Table 1. A sequence database

SID	Sequence
1	⟨(ab)(b)(b)(ab)(b)(ac)⟩
2	⟨(ab)(bc)(bc)⟩
3	⟨(b)(ab)⟩
4	⟨(b)(b)(bc)⟩
5	⟨(ab)(ab)(ab)(a)(bc)⟩

Sequence α is a prefix of β if and only if $\alpha_i = \beta_i$ for all $1 \leq i \leq n < m$. After the prefix part α is removed from sequence β, then the remaining part of β is called a postfix of β. Sequence α is an incomplete prefix of β if and only if $\alpha_i = \beta_i$ for all $1 \leq i \leq n-1$, $\alpha_n \subset \beta_n$ and all items in (β_n - α_n) are lexicographically after those in α_n. From the above definition, it can be derived that a sequence of size k has $(k-1)$ prefixes. For example, a sequence ⟨(a)(bc)(d)⟩ has 2 prefixes: ⟨(*a*)⟩ and ⟨(*a*)(*bc*)⟩. Therefore, ⟨(*bc*)(*d*)⟩ is the postfix for prefix ⟨(*a*)⟩, and ⟨(*d*)⟩ is the postfix for prefix ⟨(*a*)(*bc*)⟩. However, neither ⟨(*a*)(*b*)⟩ nor ⟨(*bc*)⟩ is considered as a prefix of given sequence, but ⟨(*a*)(*b*)⟩ is a incomplete prefix of given sequence.

A sequence α can be extended in two ways are that itemset extension and sequence extension [5]. In sequence extension, we add a single frequent item from I to the sequence α as a new itemset, so the size of sequence-extended sequence always increases. Sequence α becomes the prefix of all sequence-extended sequences of α. In itemset extension, an item is added to the last itemset in the sequence α such that this added item must be greater than all items in the last itemset. The size of itemset-extended sequences does not change and α is an incomplete prefix of all sequences that extended from these itemset-extended sequences. For example, sequences ⟨(*a*)(b)⟩ and ⟨(*a*)(*c*)⟩ are sequence-extended sequences of ⟨(*a*)⟩, and ⟨(*ab*)⟩ is an itemset-extended sequence of ⟨(*a*)⟩. Sequence ⟨(*a*)⟩ is a prefix of sequences ⟨(*a*)(b)⟩, ⟨(*a*)(*c*)⟩ and an incomplete prefix of ⟨(*ab*)⟩.

A prefix tree is similar to a lexicographic tree, which starts from the tree root at level 0. The root is set with a null sequence ∅, each child node stores a sequential pattern. At level 1, each node is set with a frequent item; at level k, each node is set with a k-pattern. Recursively, we have nodes at the next level $(k+1)$ after extending a k-pattern with a frequent item.

Given a minimum support threshold *minSup* and a sequence database *SD*, the problem of mining sequential generator patterns is to find the all of sequential generator patterns in *SD*.

4 Mining Sequential Generator Patterns

4.1 Unique Characteristics of Sequential Generator Patterns

Property 1: SGP(i) = i, ∀i ∈ I and i be a sequential 1-pattern.

Proof: Suppose that S' is a sequential generator pattern of S which contains a single item, since $S' \neq \varnothing$ and $S' \subseteq S$, so $S' = S$.

Property 2. (Apriori Property of Itemset Generators) [7]
If an itemset is a generator, then all its subsets are also generators. One might be tempted to use this property to mining sequential generator patterns problem. That is, "If a sequential pattern is a generator, then all its subsequences are also generators". Unfortunately, this property does not hold for sequential generators. So, on the other hand, the following apriori property can be possessed for non-generators of sequential patterns which can help to speed up the mining of sequential generator patterns.

Property 3. (Apriori Property of Non-Generators)
Given a sequential pattern S_1 is a sequential generator pattern, if there exist another pattern S_2 such that they have the same support and S_1 is the incomplete prefix of S_2 then S_2 and any extension of S_2 are not also generators.

Proof: That S_2 is not a generator is obvious from the premises. Next, since S_1, S_2 have the same support and S_1 is an incomplete prefix of S_2, if we can extend pattern S_2 (resp. S_1) by a series of events S_x, we can always extend pattern S_1 (resp. S_2) by the same S_x. Thus, for any series of events S_x, the support of S_2++ S_x will always be the same as S_1++S_x where the symbol "++" means concatenation of two sequences. Hence, all extensions of S_2 (*i.e.,* S_2++ S_x) are not generators.

***Remark 1*:** Given two sequences α and β, if α is a prefix of β (i.e., α is a proper subsequence of β) and $sup(\alpha) = sup(\beta)$, then any extension to β , which has the same support as β, cannot be a generator.

Remark 2: A sequence $S = s_1, s_2, ..., s_n$ is a generator if and only if $\neg\exists i$ such that $1 \leq i \leq n$ and $sup(S) = sup(s_{(i)})$.

4.2 Proposed Algorithm

Based on the characteristics of sequential generator patterns in section 4.1 and extension of sequence on the prefix tree by doing depth-first search, we propose an algorithm to generate all of the sequential generator patterns as shown in Fig.1. Using the prefix tree, we can easily create new sequences which are parent nodes by appending an item to the last position of child node as itemset extension or sequence extension. Our proposed algorithm will use the prime block encoding approach to represent candidate sequences and join operations over the prime blocks in PRISM algorithm [5] to determine the frequency for each candidate.

4.3 An Example

Consider the sequence database presented in table 1, with *minSup* = 50%. The all of sequential patterns and sequential generator patterns are shown in table 2, where the complete set of sequential generator patterns consists of only 8 sequences while the whole set of sequential patterns consists of 21 sequences. Consider the process of extending subtree and finding sequential generator patterns at sequence node {⟨(*a*)⟩:4} on the prefix tree. Extending this node based on itemset extension and sequence extension, we have sequential patterns: {⟨(*ab*)⟩:4, ⟨(*a*),(*b*)⟩:3, ⟨(*a*),(*c*)⟩:3}. Since

sequences $\langle(a)\rangle$, $\langle(ab)\rangle$ have the same support and satisfy property 3, then sequence $\langle(ab)\rangle$ is a non-generator pattern and all another extension of $\langle(ab)\rangle$ are also non-generator patterns so we don't need to check the generator property for all sub nodes of $\langle(ab)\rangle$. We repeat the above process for all sub nodes of $\langle(a)\rangle$ containing all nodes except the non-generator pattern nodes which satisfy property 3 to extend tree and find sequential generator patterns from these sub nodes. We also repeat the above process for all remaining nodes on the prefix tree and results are presented in Table 2.

Input: Sequence database *SD*, *minSup*.
Output: Set of the sequential generator patterns: *SGPs*.
Method:
MSGPs(SD, minSup)
dbpat ← all frequent sequence 1-pattern;
SGPs ← all frequent sequence 1-pattern;
for each Pattern *P* in *dbpat*
EXTEND_TREE(P, dbpat, SGPs);//extending tree with root node is P.
return *SGPs*;
//.....
EXTEND_TREE(Root, dbpat, SGPs)
For each *1*-pattern *P* in *dbpat*
Let *Pitems_ext* is a new pattern that creating by itemset extension of *Root* and using block encoding based on prism factorization in [5] to count the support;
If (*sup(Pitems_ext)* ≥ *minSup* and *sup*(*P*) = *sup*(*Pitems_ext*))
Set *Pitems_ext* is non-generator pattern;
Else
Checking whether *Pitems_ext* is generator pattern or not. If *Pitems_ext* is a generator pattern then add *Pitems_ext* into *SGPs* else set *Pitems_ext* is non-generator pattern;
Add *Pitems_ext* into the itemset extended set of *P*;
Let *Pseq_ext* is a new pattern that creating by sequence extension of *Root* and using block encoding based on prism factorization in [5] to count the support;
If (*sup(Pseq_ext)* ≥ *minSup* and *sup*(*P*) = *sup*(*Pseq_ext*))
Set *Pseq_ext* is non-generator pattern;
Else
Checking whether *Pseq_ext* is generator pattern or not. If *Pseq_ext* is a generator pattern then add *Pseq_ext* into SGPs else set *Pseq_ext* is non-generator pattern;
Add *Pseq_ext* into the sequence extended set of *P*;
For each node P_i is an itemset extension of *Root*
If P_i is a generator pattern
EXTEND_TREE(P_i, dbpat, SGPs);// recursively call to extend tree with root node be P_i.
For each node P_s is a sequence extension of *Root*
EXTEND_TREE(P_s, dbpat, SGPs); // recursively call to extend tree with root node be P_s.

Fig. 1. MSGPs algorithm for generating the set of sequential generator patterns

Table 2. The list of sequential patterns and sequential generator patterns

Nodes	Sequential patterns	Sequential generator patterns
a	⟨(a)⟩: 4, ⟨(a)(b)⟩: 3, ⟨(a)(c)⟩: 3, ⟨(ab)⟩: 4, ⟨(a)(b)(b)⟩: 3, ⟨(a)(b)(c)⟩: 3, ⟨(ab)(b)⟩: 3, ⟨(ab)(c)⟩ : 3, ⟨(ab)(b)(b)⟩: 3, ⟨(ab)(b)(c)⟩: 3	⟨(a)⟩: 4 ⟨(a)(b)⟩: 3 ⟨(a)(c)⟩: 3
b	⟨(b)⟩: 5, ⟨(b)(a)⟩: 3, ⟨(b)(b)⟩: 5, ⟨(b)(c)⟩: 4, ⟨(bc)⟩: 3, ⟨(b)(ab)⟩: 3, ⟨(b)(b)(b)⟩: 4, ⟨(b)(b)(c)⟩: 4, ⟨(b)(bc)⟩: 3, ⟨(b)(b)(bc)⟩: 3	⟨(b)⟩: 5 ⟨(b)(a)⟩: 3 ⟨(bc)⟩: 3 ⟨(b)(b)(b)⟩: 4
c	⟨(c)⟩: 4	⟨(c)⟩: 4

5 Experimental Results

To evaluate the performance of the MSGPs algorithm for mining sequential generator patterns, we are comparing its performance with a second algorithm as FSGP [20]. All experiments are performed on PC machine with dual-core 2.81 GHz, 2 GBs RAM, running Windows XP professional, and all algorithms are implemented using C# (2008). We perform the experiments on synthetic and real databases include C6T5S4I4N1kD1k, Chess, and Mushroom. C6T5S4I4N1kD1k was generated using the synthetic data generator provided by IBM to mimic transactions in a retail environment with the following parameters: *C* is the average number of itemsets per sequence that is set to 6 (denoted as *C6*), *T* is the average number of items per itemset that is set to 5 (denoted as *T5*), *S* is the average number of itemsets in maximal sequences that is set to 4 (denoted as *S4*), *I* is the average number of items in maximal sequences that is set to 4 (denoted as *I4*), *N* is the number of distinct items that is set

Table 3. Experimental results in databases

Database	Minsup (%)	Number of Sequential Patterns	Number of Sequential Generator Patterns	Time for mining		Scale (2)/(1) %
				FSGP (1)	MSGPs (2)	
Chess	80	8227	5113	240.57	222.40	92.45
	75	21000	11598	823.97	744.44	90.35
	70	49020	24763	2336	2041.75	87.40
	65	112103	53309	6329.37	5436.29	85.89
	60	254110	113097	18806.69	15193.98	80.79
C6T5S4I4 N1kD1k	0.6	20644	20391	2200.96	2172.23	98.69
	0.5	31311	30692	3445.71	3386.48	98.28
	0.4	54566	52017	6418.37	6251.07	97.39
	0.3	124537	108085	17672.59	16556.23	93.68

to 1,000 (denoted as *N1k*), and *D* is the number of sequences include 1,000 sequences (denoted as *D1k*). Chess database was downloaded from http://fimi.ua.ac.be/data/ where each itemset in a sequence is a single item. Chess database includes 3196 sequences with 76 distinct items.

Table 3 shows all of sequential patterns, sequential generator patterns and the execution time of the two algorithms, which are comparing together, in different databases corresponding to their minimum supports. The experimental results in table 3 show that the number of sequential generator patterns is always smaller the number of sequential patterns and the run time for mining using our MSGPs algorithm is always faster than that of the FSGP algorithm in all cases, because MSGPs algorithm applied both the supersequence frequency-based pruning and non-generator-based pruning on the prefix tree to reduce the search space. The time scale was calculated as follows: (mining time on MSGPs / mining time on FSGP) *100%. For example, consider the Chess database with *minSup* is 60%, there are 254,110 sequential patterns and 113,097 sequential generator patterns. The mining time based on the MSGPs was 15,193.98, and based on the FSGP was 18,806.69, such that the time scale was (15,193.98/18,806.69)*100%, which was 80.79%.

6 Conclusions and Future Works

In this paper, we proposed an algorithm called MSGPs, which based on the characteristics of sequential generator patterns and sequence extensions by doing depth-first search on the prefix tree, to generate all of the sequential generator patterns. This algorithm used a vertical approach for enumeration and support counting, based on the notion of prime block encoding, which in turn is based on prime factorization theory to determine the frequency for each candidate [5]. To reduce search space, MSGPs algorithm also applied both the super sequence frequency-based pruning and non-generator-based pruning on the prefix tree.

The experimental results in synthetic and real databases showed that performance runtime for mining sequential generator patterns in our proposed algorithm is better than existing algorithms as FSGP [16]. In the future, we will research and produce the methods for mining the set of closed sequential patterns and minimal generator patterns of them at the same process based on the prefix tree. After that, we will generate non-redundant rules from these sets. Several relations between objects as reflexive relation, coherent relation, equivalence relation and so on were used for conflict resolution in the classification of ontology conflicts [10]. So, we will consider and study how to apply the one of these relations for mining sequential pattern problem in the future.

Acknowledgements. The work is supported by the National Natural Science Foundation of China (Grant no. 60873184) and the Science and Technology Planning Project of Hunan Province (Grant no. 2011FJ3048).

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of 11th Int'l Conf. Data Engineering, pp. 3–14 (1995)
2. Ayres, J., Gehrke, J., Yiu, T., Flannick, J.: Sequential pattern mining using a bitmap representation. In: Proc. of ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 429–435 (2002)
3. Chang, L., Wang, T., Yang, D., Luan, H., Tang, S.: Efficient algorithms for incremental maintenance of closed sequential patterns in large databases. Data and Knowledge Engineering 68(1), 68–106 (2009)
4. Gao, C.C., Wang, J.Y., He, Y.K., Zhou, L.Z.: Efficient mining of frequent sequence generators. In: Proc. of the 17th International Conference on World Wide Web, Beijing, China, pp. 1051–1052 (2008)
5. Gouda, K., Hassaan, M., Zaki, M.J.: PRISM: A primal-encoding approach for frequent sequence mining. Journal of Computer and System Sciences 76(1), 88–102 (2010)
6. Huang, G., Yang, F., Hu, C., Ren, J.: Fast discovery of frequent closed sequential patterns based on positional data. In: Proc. of the 9th International Conference on Machine Learning and Cybernetics, Qingdao, vol. 1, pp. 444–449 (2010)
7. Li, J., Li, H., Wong, L., Pei, J., Dong, G.: Minimum description length (MDL) principle: Generators are preferable to closed patterns. In: Proc. of the 21th National Conference on Artificial Intelligence, AAAI 2006, Boston, Massachusetts, USA, pp. 409–414 (2006)
8. Lo, D., Khoo, S.-C., Li, J.: Mining and ranking generators of sequential patterns. In: SIAM Conference on Data Mining, SDM 2008, Atlanta, Georgia, USA, pp. 553–564 (2008)
9. Luo, C., Chung, S.M.: A scalable algorithm for mining maximal frequent sequences using a sample. Knowledge and Information Systems 15(2), 149–179 (2008)
10. Nguyen, N.T.: A Method for Ontology Conflict Resolution and Integration on Relation Level. Cybernetics and Systems 38(8), 781–797 (2007)
11. Pei, J., et al.: Mining sequential patterns by pattern-growth: The PrefixSpan approach. IEEE Transaction on Knowledge and Data Engineering 16(10), 1424–1440 (2004)
12. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
13. Wang, J., Han, J.: BIDE: Efficient mining of frequent closed sequences. In: Proc. of the 20th Int' Conf. on Data Engineering, ICDE 1995, pp. 79–91. IEEE Computer Society Press (2004)
14. Wang, J., Han, J., Li, C.: Frequent closed sequence mining without candidate maintenance. IEEE Transaction on Knowledge and Data Engineering 19(8), 1024–1056 (2007)
15. Yan, X., Han, J., Afshar, R.: CloSpan: Mining closed sequential patterns in large datasets. In: Proc. of the 3rd SIAM International Conference on Data Mining, pp. 166–177. SIAM Press, San Francisco (2003)
16. Yia, S.W., Zhao, T.H., Zhang, Y.Y., Ma, S.L., Che, Z.B.: An effective algorithm for mining sequential generators. In: Procedia Engineering, CEIS 2011, vol. 15, pp. 3653–3657 (2011)
17. Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal 42(1-2), 31–60 (2001)

A Genetic Algorithm with Elite Mutation to Optimize Cruise Area of Mobile Sinks in Hierarchical Wireless Sensor Networks

Mong-Fong Horng[1], Yi-Ting Chen[1], Shu-Chuan Chu[2], Jeng-Shyang Pan[1,3], Bin-Yih Liao[1,*], Jang-Pong Hsu[4], and Jia-Nan Lin[4]

[1] Department of Electronics Engineering,
National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan
[2] School of Information Science, Engineering and Mathematics,
Flinders University, Adelaide, Australia
[3] Innovative Information Industry Research Center (IIIRC),
Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
[4] Advance Multimedia Internet Technology Inc., Tainan, Taiwan
byliao@cc.kuas.edu.tw

Abstract. In this paper, a new genetic algorithm with elite mutation is proposed for optimization problems. The proposed elite mutation scheme (EM) improves traditional genetic algorithms with a better ability to locate and to approach fast to optimal solutions, even in cases of huge data set. The proposed EM is to select elite chromosomes and mutate according to the similarity between elite chromosomes and selected chromosomes. The designed similarity guides effectively the search toward optimal solutions with less generation. The proposed EM is applied to optimize the cruise area of mobile sinks in hierarchical wireless sensor networks (WSNs). Numeric results show that (1) the proposed EM benefits the discovery of optimal solutions in a large solution space; (2) the approach to optimal solutions is more stable and faster; (3) the search guidance derived from the chromosome similarity is critical to the improvements of optimal solution discovery. Besides, the minimization of cruise are been proved to have the advantages of energy-saving, time-saving and reliable data collection in WSNs.

Keywords: Genetic algorithm, Elite mutation, Cruise area optimization, Mobile data sinks, Hierarchical wireless sensor networks.

1 Introduction

During the last years, wireless sensor networks (WSNs) have attracted a lot of attentions from academics and industry due to its variety of application. WSN is employed in many military and civilian applications, including smart home and internet of thing (IoT). The main application of WSN is to monitor and sense the environmental change through measurement such as temperature, humidity, lighting and so on. Battery is the only power supply for sensors in a WSN. And the restricted power limits definitely network lifetime. Hence, Effective power consumption of sensor nodes is a

* Corresponding author.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 402–412, 2012.

highly important issue when improving network lifetime. As well known, the power consumption of sensor nodes is mainly related to the distance between source nodes and destination nodes. Cluster-based WSNs consume less power because of hop-by-hop transmission. Such a hierarchical WSN architecture was confirmed to reduce power consumption and to prolong network lifetime.

In a static hierarchical WSN, all sensor nodes are organized through clustering technique [1-2]. Each sensor in a cluster delivers the sensed data to corresponding cluster head. Cluster heads forward data from sensors to data sinks. However, the cluster heads will have a lot of workload and fast exhaust power. Therefore, in order to reduce the power consumption of cluster heads, mobile sinks are designed to cruise among sensor nodes to collect data. The mobile sink is able to move to the deployed area of cluster heads to collect date by shortening the transmission range between cluster head and data sink. Mobile data sink benefits WSNs with (1) shorter transmission distances between cluster heads and data sinks; (2) a reduction of signal interference between nodes causing by long distance transmission; (3) secure data delivery over a short transmission; (4) considerable energy-saving of nodes due to short transmission and (5) longer network lifetime contributed by high energy efficiency in nodes. In a hierarchical WSN, all nodes deliver their measurement to their cluster heads and cluster heads are the targets which mobile sinks intend to visit. Thus, how to select the cluster heads in different clusters to minimum cruise area of mobile sink is an interest issue. This issue is a typical NP-hard problem to select the cluster heads with minimum total distance between clusters. In other words, the discovery of the cluster head set with minimum summation of head-to-head distances is an intensive computational problem. A traditional computational approach did not meet the requirement of real-time applications. However, evolutionary computing is a possible and feasible solution to this problem. Traditional evolutionary computing scheme, such as Genetic Algorithm (GA), cannot guarantee the reliable and efficient search for optimal solutions. Especially in a huge solution space, how to guide the search toward to optimal solutions is still an open question. Local optimal is a challenge in an evolution of solution discovery. Traditional GA applies mutation operation to overcome the local optimal problem. Nevertheless, legacy mutation lacks of a guided search feature to fasten the approach to the optimal solution. In fact, during the genetic evolution, the analysis of elite chromosomes is helpful to derive the premium evolution direction for offspring. Hence, a new genetic algorithm with elite mutation (GAEM) is proposed to realize a fast and efficient genetic evolution to converge to optimal solutions. Based on the analysis of elite chromosomes in generations, the chromosome mutation of offspring is designed to be random and guided. By the integration of random and guided mutation, an efficient mutation is proved to be beneficial to the discovery of optimal solutions.

The rest of this study is organized as follows. In Section 2, a survey of wireless sensor network and genetic algorithm is discussed. In Section 3, the application scenario and a genetic algorithm with elite mutation to optimize cruise area of mobile sink are presented. In Section 4, there are two sparse and close patterns of sensor nodes distribution to verify the stability and accuracy of proposed algorithm. And the search ability of proposed algorithm is discussed in this section. Finally, in Section 5, we will summarize the contribution of this study and present the future work on this issue.

2 Related Work

A wireless sensor network is composed of a number of heterogeneous or homogeneous sensor nodes, routers and data sinks. Sensor nodes are low-cost, low-power and multi-functional. These sensor nodes communicate each other within short distance in a network through wireless radio. In hierarchical WSNs [3-5], cluster heads deliver the collected data from sensor nodes to data sinks. Data sinks analyze these collected data to observe the deployed environmental information. This transmission architecture easily causes that the power of cluster heads are exhausted very quickly. Recently, several WSN architectures based on mobile sinks were proposed [6]. Mobile sink visits sensor nodes in a network to collect data. Mobile sink represents the endpoints of data collection in a mobile WSM (MWSN) [7-9]. Deepak Puthal *et al.* proposed a Mobile Sink Wireless Sensor Network (MSWSN) model [10] to design mobile sinks to collect data from the static nodes in a network. MSWSN model focuses on the sink to move with the relative distance, direction and speed to increase the delivery ratio, residual energy and network lifetime. Luo *et al.* proposed a routing protocol, MobiRoute for the path planning of mobile sinks to improve network lifetime and packet deliver ratio, where the sink sojourns at some anchor points and the pause time is much longer than the movement time. However sensor nodes in MobiRoute need to know the topological changes caused by the sink mobility [11]. Heinzelman *et al.* [12] proposed sink mobility to minimize the data loss during the transition of mobile sink from its current location to its next location. Papadimitriou *et al.* proposed a novel linear programming to maximize the network lifetime that can be achieved by solving optimal sink sojourn time at different locations and route data towards sinks. The mobile sink can move between different places during network operations and the data routing is performed across multiple hops with different transmission energy requirements. This approach is good at a fair balancing of the energy depletion amount the sensor nodes [13].

In a large-scale WSN, the network may be not a full reachability topology due to randomly deployment or limited transmission range. Hence, the collected data by sink is incomplete the power consumption is high. In order to improve these drawbacks, the mobile sink is one of the feasible solutions to solve this issue. Mobile sink can visit each sensor in network to collect data and decrease power consumption of sensors to improve the WSN lifetime [14, 15]. The route planning is required before a trip through all the sensors to increase the efficiency of mobile sink. However, the energy saving problem of mobile sink is also considered. The energy consumption of mobile sink is affected through visit routes. Long route will increase the energy of mobile sink to lead to increase charge times. Hence, a shortest route for the mobile sink is necessary to visit through all sensors. This issue is also called mobile sink routing problem. However, the cost of visiting all sensors in network is high for mobile sink. In this study, a cluster-based network is considered. These networks will construct many clusters through cluster technique. The cluster heads gather all information from sensors within its cluster. Mobile sink only focuses on cluster heads to collect whole data of all sensors. This method can minimize the cruise area, decrease computing complexity through route planning of mobile sink. In addition, how to plan route under restricted conditions is a difficult issue. Hence, a genetic algorithm with elite mutation is

proposed to optimize cruise area of mobile sink in hierarchical wireless sensor network based on cluster topology with full reachability in this study. The detail description of application scenario and proposed algorithm is illustrated in Section 3.

3 A Genetic Algorithm with Elite Mutation to Optimize Cruise Area of Mobile Sinks

3.1 Service Scenario of WSNs with Mobile Sinks

In this study, a genetic algorithm with elite mutation is proposed to optimize cruise area of mobile sink in hierarchical wireless sensor networks. All sensors in deployed network are clustered through cluster technique [16]. Then each cluster assumes adjustment model of the transmission range with a minimum node degree to establish a full-reachability topology [17]. In a full-reachability topology, there is at least a route between any node pair in a cluster. In order to decrease the power consumption of cluster heads (routers), the mobile sink is adopted in a wireless sensor network to collect data for cluster heads. The mobile sink will move in cruise area because each cluster in network is full-reachability topology.

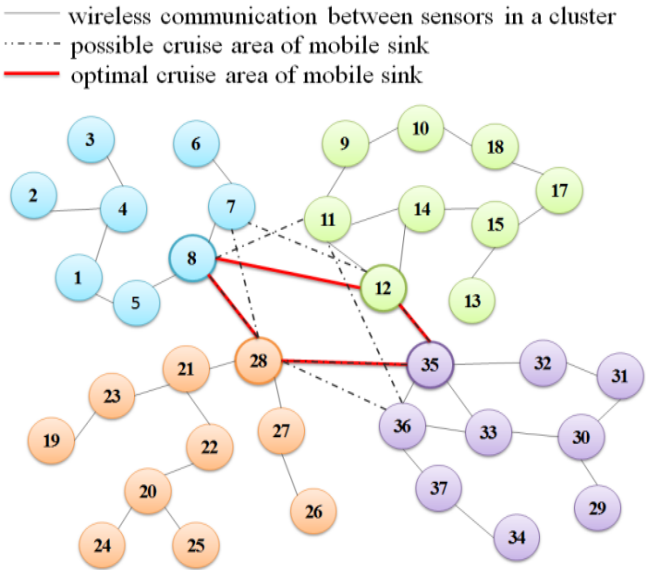


Fig. 1. An optimal cruise area of mobile in a hierarchical wireless sensor networks

To ensure the quality of collected data and reduce the workload of mobile sink, the minimum cruise area of a mobile sink is necessary requirement. Each of clusters will have a sensor node to be a cluster head to gather data from the other nodes in a cluster and forward the data to a mobile sink. The minimum cruise area can decrease the

energy cost of mobile sink and improve the energy efficiency of nodes in a WSN. The application scenario is described in Fig. 1. The cruise area is derived from the locations of cluster heads. In this example, sensor 7, sensor 28, sensor 36 and sensor 11 form a cruise area of a mobile sink. However, this area derived from sensor 7, sensor 28, sensor 36 and sensor 11 bigger than the area from sensor 8, sensor 28, sensor 35 and sensor 12, will cause large power consumption of a mobile sink. As a result, the cruise area formed by sensor 8, sensor 28, sensor 35 and sensor 12 is the optimal cruise area for mobile sink in this network topology. Finally, the optimal cruise area is determined by the cluster heads in which the distances between cluster heads is minimum. This criterion will be applied to find the set of cluster heads in the next section.

3.2 A Genetic Algorithm with Elite Mutation (GAEM)

Genetic Algorithm (GA) is a global search heuristic optimization technique [18-20]. This algorithm searches the global optimum solution based on function definition of the problem through evolution procedure. Fitness value determines the quality of the individual on the basis of the defined Fitness function. The fundamental part of a genetic algorithm is explained as follows.

(1) Initialization: The individual is called as chromosome and consisted of random gene with a sequence of 0s and 1s. All individuals may be toward the optimum solution through repetitive evolution process such as selection, crossover and mutation in a generation. In this phase, there are many parameters will be set. For example: population size, chromosome length, generation times and so on.
(2) Fitness: The fitness function is defined according to application issue. All chromosomes are evaluated through fitness function to obtain fitness value. The fitness value is indicated the survival ability for each chromosome. The fate of chromosomes depends on its fitness value. The chromosome will have a higher survival chance to product new offspring when the fitness value of chromosome is better.
(3) Selection: During each generation, all new offspring are generated by selected chromosomes of the current generation after crossover and mutation procedure. The fitter chromosomes are almost always selected to lead to the search direction to search best solution. There are several selection methods: Roulette-Wheel Selection, Rank Selection and Tournament Selection. Tournament Selection is used in this study.
(4) Crossover: The genes of chromosomes will recombine through selected chromosome (parents) to product new chromosome (offspring). Crossover is a simulation of the sexual reproductive process for transfer of genetic inheritance. There are many version of crossover operation is designed for different cases. The same pair of chromosome will produce different offspring through various crossover operations. The simplest is the single-point crossover through a point is chosen at random. The two parent chromosome will exchange gene after the point.
(5) Mutation: Mutation procedure is performed to overcome the problem that the search direction falls into local optimum. Mutation operation can avoid premature

convergence through occasional random alternation. The randomly selected genes in a chromosome are changed to produce highly different chromosome material. This procedure can expand search area in solution space to help algorithm fast obtain best solution (optimal solution).

The proposed GA with elite mutation is as shown in Fig. 2. First, the population in generation k is denoted as $X^k = \{x_1^k, x_2^k, \ldots, x_n^k\}$ and the chromosome $x_i^k = \{s_1, s_2, \ldots, s_c\}$ where $s_i = \lceil log_2 S \rceil$, S is the number of nodes in a cluster and c is the cluster number in a WSN. Then, each chromosome is evaluated by the following fitness function defined by

$$fit(x_i^k) = d\_sum(x_i^k) \tag{1}$$

$$d_{sum(x_i^k)} = \frac{1}{2}\sum_{u}^{c}\sum_{v}^{c} dist(s_u, s_v) \quad u \neq v \tag{2}$$

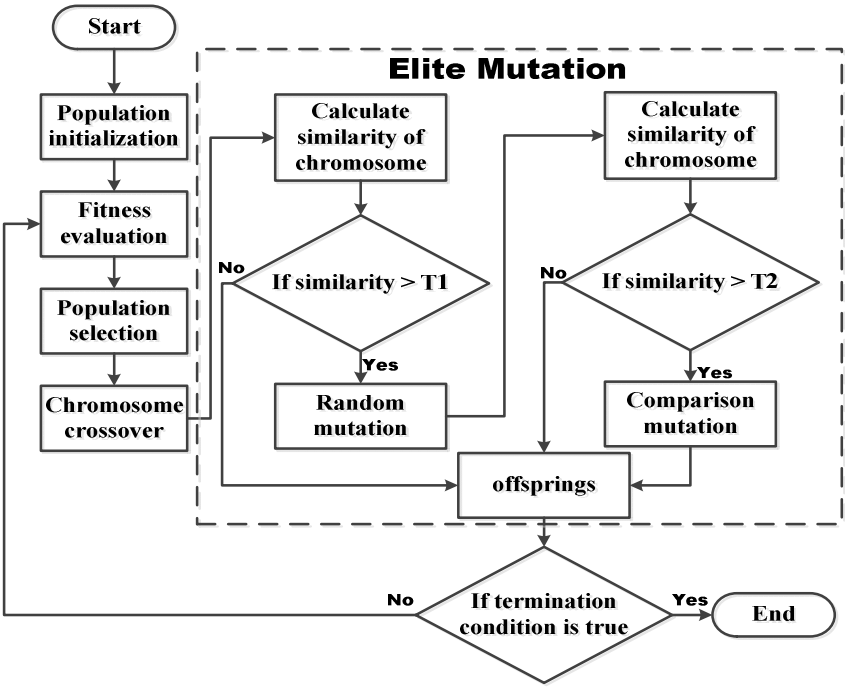


Fig. 2. The flowchart of a genetic algorithm with elite mutation

The fitness function is defined to evaluate the distance summation of all cluster heads. An ordered list of chromosomes according to fitness is obtained as

$$fit(x_i^k) \leq fit(x_{i+1}^k) \leq \cdots \leq fit(x_n^k)$$

Then, the selection rate (SR) is set to 0.5 to keep a half of population denoted as $X_{sur}^{k}$for crossover.

$$X_{sur}^{k} = \{\hat{x}_{i}^{k}, \hat{x}_{i+1}^{k}, \ldots, \hat{x}_{SR\times PS}^{k}\} \tag{3}$$

$$if\ fit(x_{1}^{k}) < fit(x_{1}^{k-1}) \tag{4}$$

A two-point cross over is applied in this work to have offspring with high diversity as follows,

$$(x'^{k}_{i}, x'^{k}_{j}) = corssover(\hat{x}_{i}^{k}, \hat{x}_{j}^{k}) \tag{5}$$

where $x'^{k}_{i} = \hat{x}_{i}^{k}(1{:}P_1) \oplus \hat{x}_{j}^{k}(P_1{:}P_2) \oplus \hat{x}_{i}^{k}(P_2{:}l)$ and $x'^{k}_{j} = \hat{x}_{j}^{k}(1{:}P_1) \oplus \hat{x}_{i}^{k}(P_1{:}P_2) \oplus \hat{x}_{j}^{k}(P_2{:}l)$ *for* $1 < P_1, P_2 < l$.

In mutation, we propose an elite mutation to offer a better search direction based on a similarity analysis of elite chromosomes. At first, each chromosome $\dot{x}_{i}^{k}$ is evaluated by the similarity compared to the best chromosome $\dot{x}_{1}^{k}$. Those chromosomes with a similarity greater than a threshold *T1* will be selected for ransom mutation. The obtain offspring is evaluated and reserved as X_{elite1}^{k}. The offspring produced from the mutation will be selected again by the similarity with a high threshold, *T2* to obtain the elite chromosomes denoted as X_{elite2}^{k}. And the final population selected for survival is given by

$$X^{k+1} = X^{k} \cup X_{elite1}^{k} \cup X_{elite2}^{k} \tag{6}$$

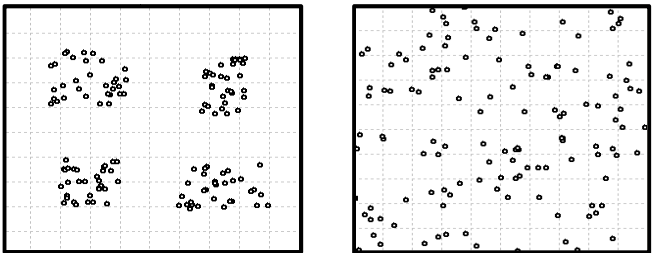
By repeating the evolution given by Eq. (1-6), the proposed GA with EM effectively discovers the optimal solutions. Then, we will apply the proposed approach to find the optimal cruise area for mobile sink in a WSN.

4 Simulation and Performance Evaluation

In performance evaluation, we simulate various parameters of genetic algorithm and patterns of node distribution. Then, a representative simulated result is selected to verify the performance of proposed algorithm. This simulation result is expressed in this section. For the relation between parameters, we will discuss and analyze in future work. There are four typical patterns of node distribution for simulations including close and sparse nodes to verify the effectiveness and stability of proposed genetic algorithm. There are 128 sensors deployed in an area of 300km x 300km. And all sensors are clustered four clusters. The network topologies are display in Fig 2. The first test case is close network pattern. The second test case is sparse network pattern. In initialization, the binary code is used to design chromosome construction. The algorithm parameters include: population size=20, chromosome length=$\lceil log_2^{128} \rceil \times 4$, generation=10,000 and selection rate=0.5. The crossover rates are 0.2 and 0.5 in TGA and GAEM respectively. The mutation rate is 0.3 in TGA. In GAEM, the mutation rate is flexible and the two thresholds are 0.5 and 0.8 respectively. The simulated results of

traditional genetic algorithm and proposed GAEM in close and sparse pattern are shown in Fig. 3. In the case of close pattern, the proposed GAEM can obtain the global optimal solution (fitness value=532.79). And GAEM has converged in 6,500 generation. In other case of sparse pattern, GAEM finds the global optimal solution (fitness value=270.97) in 6,500 generation. However, TGA has fallen into the local optimal solution (fitness value=366.02) in 5,000 generation. For the effectiveness of algorithm, GAEM increase 7.67% and 26% effectiveness than TGA in different patterns.

Next, the stability of algorithm will be discussed. The same simulation is repeated to validate the stability in terms of convergence time. The stabilities of TGA and GAEM for sparse pattern are shown in Table 1. For sparse pattern, the maximum, minimum and average convergence times of GAEM in ten simulations are the same (532.79) in 10,000 generation. The stability of GAEM is 100%. However, the maximum and minimum convergence time of TGA is 626.05 and 532.79 respectively. The stability of TGA is about 85%. For other case with close pattern, the stability of GAEM is still 100%. The stability of TGA is 55%. And TGA cannot obtain the global optimal solution. Hence, these simulated results prove the GAEM is more stability than TAG in these two patterns.



(a) close network pattern (b) sparse network pattern

Fig. 3. The network topoloies of close and sparse four clusters

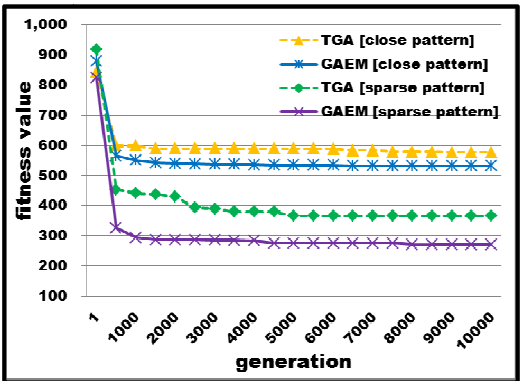


Fig. 4. A comparison of the simulated results for tradition genetic algorithm and proposed genetic algorithm in different patterns

Table 1. Convergence Comparison of traditional GA and the proposed GA with Elite Mutation in sparse and close patterns of node distribution

	Close pattern				Sparse Pattern			
	TGA		GAEM		TGA		GAEM	
Gen.	1	10000	1	10000	1	10000	1	10000
Max.	955.97	626.05	1051.65	532.79	1054.50	568.87	952.00	270.97
Avg.	842.98	576.99	879.89	532.79	918.52	453.63	825.33	270.97
Min.	694.65	532.79	738.81	532.79	755.81	314.94	710.96	270.97

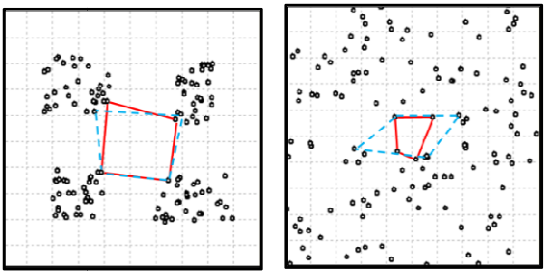


Fig. 5. The optimal cruise area of TGA and GAEM

The optimal cruise areas found through TGA and GAEM are shown in Fig. 5, respectively. The red solid line indicates the optimal cruise area by GAEM. The blue dotted line implies the optimal cruise area by TGA. The red cruise area is narrower than blue cruise area obviously. Hence, the mobile sink only focuses on these cluster heads to collect data within this red area.

5 Conclusions

In this paper, a new genetic algorithm with elite mutation is proposed for the problem of optimizing the cruise area of mobile sinks in WSN. Mobile sinks are designed to cruise among cluster heads to collect data. Mobile data sinks benefits WSNs with interference, secure data delivery, considerable energy-saving and longer network lifetime. However, how to find an optimal cruise area for mobile sinks is a time-consuming problem. Although traditional GA contributes a solution to this problem, local optimal and long convergence affect the performance of discovering the optimal solution. The proposed elite mutation scheme (EM) improves traditional genetic algorithms with a better ability to locate and to approach fast to optimal solutions, even in cases of huge data set. The proposed EM is applied to optimize the cruise area of mobile sinks in hierarchical wireless sensor networks (WSNs). Numeric results show that (1) the proposed EM benefits the discovery of optimal solutions in a large solution space; (2) the approach to optimal solutions is more stable and faster; (3) the search guidance derived from the chromosome similarity is critical to the improvements of optimal solution discovery. Besides, the minimization of cruise are been proved to have the advantages of energy-saving, time-saving and reliable data

collection in WSNs. In the future, the relation between parameters of genetic algorithm will be discussed and analyzed for this case. And stability and reliability of the proposed genetic algorithm will be verified through different problem kinds.

Acknowledgements. The authors would like to express their sincere thanks to the National Science Council, Taiwan for their financial support under the grants NSC- 98-2221-E-151-029-MY2, NSC-99-2623- E-110-002-D, and NSC-100-2623-E-006-009-D.

References

1. Abbasi, A.A., Younis, M.: A survey on clustering algorithms for wireless sensor networks. Computer Communication 30, 2826–2841 (2007)
2. Hong, T.P., Wu, C.H.: An Improved Weighted Clustering Algorithm for Determination of Application Nodes in Heterogeneous Sensor Network. Journal of Information Hiding and Multimedia Signal Processing 2(2), 173–184 (2011)
3. Lung, C.H., Zhou, C.J.: Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach. Ad Hoc Networks 8(3), 328–344 (2010)
4. Manisekaran, S.V.: Energy Efficient Hierarchical clustering for sensor networks. In: 2010 International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–11. IEEE Press, India (2010)
5. Slavik, M.: Analytical model of energy consumption in hierarchical wireless sensor networks. In: 2010 High-Capacity Optical Networks and Enabling Technologies (HONET), pp. 84–90. IEEE Press, Florida (2010)
6. Francesco, M.D., Das, S.K.: Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey. ACM Transactions on Sensor Networks 8(1), 7:1–7:31 (2011)
7. Chen, X.H.: Research on hierarchical mobile wireless sensor network architecture with mobile sensor nodes. In: 2010 3rd International Conference on Biomedical Engineering and Informatics (BMEI), pp. 2863–2867. IEEE Press, Lanzhou (2010)
8. Duan, Z.F.: Shortest Path Routing Protocol for Multi-layer Mobile Wireless Sensor Networks. In: 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC), pp. 106–110. IEEE Press, Nanchang (2009)
9. Chen, C.F.: Mobile Enabled Large Scale Wireless Sensor Networks. In: 8th International Conference Advanced Communication Technology (ICACT), pp. 333–338. IEEE Press, Beijing (2006)
10. Puthal, D., Sahoo, B., Sharma, S.: Dynamic Model for Efficient Data Collection in Wireless Sensor Networks with Mobile Sink. International Journal of Computer Science and Technology 3(1), 623–628 (2012)
11. Luo, J., Panchard, J., Piórkowski, M., Grossglauser, M., Hubaux, J.-P.: MobiRoute: Routing Towards a Mobile Sink for Improving Lifetime in Sensor Networks. In: Gibbons, P.B., Abdelzaher, T., Aspnes, J., Rao, R. (eds.) DCOSS 2006. LNCS, vol. 4026, pp. 480–497. Springer, Heidelberg (2006)
12. Heinzelman, W.B., Murphy, A.L., Carvalho, H.S., Perillo, M.A.: Middleware to Support Sensor Network Applications. IEEE Network 18(1), 6–14 (2004)
13. Papadimitriou, I., Georgiadis, L.: Energy-aware Routing to Maximize Lifetime in Wireless Sensor Networks with Mobile Sink. In: 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 141–151 (2005)

14. Liang, W.F.: Prolonging Network Lifetime via a Controlled Mobile Sink in Wireless Sensor Networks. In: 2010 IEEE Global Telecommunications Conference (GLOBECOM), pp. 1–6. IEEE Press, Canberra (2010)
15. Wu, X.B.: Dual-Sink: Using Mobile and Static Sinks for Lifetime Improvement in Wireless Sensor Networks. In: 16th International Conference on Computer Communications and Networks (ICCCN), pp. 1297–1302. IEEE Press, Nanjing (2007)
16. Horng, M.F., Chen, Y.T., Chu, S.C., Pan, J.S., Liao, B.Y.: An Extensible Particles Swarm Optimization for Energy-Effective Cluster Management of Underwater Sensor Networks. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010, Part I. LNCS(LNAI), vol. 6421, pp. 109–116. Springer, Heidelberg (2010)
17. Chen, Y.T., Lo, C.C., Shieh, C.S., Horng, M.F., Pan, J.S.: An optimization of adaptive transmission with guarantee connection degree for wireless sensor networks. In: 2011 IEEE International Conference on Granular Computing (GrC), pp. 121–126. IEEE Press (2011)
18. Guo, P.F.: The enhanced genetic algorithms for the optimization design. In: 3rd International Conference on Biomedical Engineering and Informatics (BMEI), pp. 2990–2994. IEEE Press (2010)
19. Jiang, W.J.: Hybrid genetic algorithm research and its application in problem optimization. In: 5th World Congress on Intelligent Control and Automation (WICIA), pp. 2122–2126. IEEE Press (2004)
20. Guo, L.J.: An Improved Routing Protocol in WSN with Hybrid Genetic Algorithm. In: 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC), pp. 289–292. IEEE Press (2010)

An Algebraic Structure for Duration Automata

Bui Vu Anh[1] and Phan Trung Huy[2]

[1] Faculty of Mathematics, Mechanics and Informatics, VNU University of Science
vuanh@vnu.edu.vn
[2] School of Applied Mathematics and Informatics, Hanoi University of Science and Technology
huypt-fami@mail.hut.edu.vn

Abstract. Algebraic structures such as group, lattice and fuzzy algebra play an important role in the field of information technology, especially in modelling systems. The previous researches [1, 2, 3, 4] have introduced a formal tool named duration automata, by which we modelled systems and developed algorithms to solve problems on scheduling jobs for a cluster computer, routing on priority network or dealing with uncertain processing time jobs. Here we introduce a new weak semigroup with the aim to combine duration automata together with algebraic structures to solve some optimization problems.

Keywords: Duration automaton, duration automata, formal method, modelling.

1 Introduction

Using algebra in information models are not only increasing the reliability and precision of modeled systems, but also the computing power, decreasing the cost of searching algorithms. The weak algebraic structures such as semi-group, lattice, semi-lattice, groupoid, fuzzy algebra are flexible and appropriate for reflecting information of systems. Beside using graph theory as a background to research on duration automata in [1, 2, 4], this paper finds and selects a suitable algebra structure, in which durations are considered as the elements. Here we propose a weak semigroup with the aim to combine duration automata together with algebraic structures to solve some optimization problems with time restrictions.

2 Algebra of Durations

Let $\mathcal{D} = \{[l, u] \mid l, u \in \mathbb{Z}, l \leq u\}$ be a set of durations. For $d_1, d_2 \in \mathcal{D}$, $d_1 \cap d_2 \neq \emptyset$ iff d_1 overlaps d_2 (and vice verse). A value t is said belonging to $d = [l, u]$ if $l \leq t \leq u$. A new type of product between two durations is defined as follows:

Definition 1. *Given $d_1, d_2 \in \mathcal{D}$, $d_1 = [l_1, u_1], d_2 = [l_2, u_2]$. The duration product of d_1 and d_2 is given by:*

$$d_1 \sqcap d_2 = \begin{cases} [l_1, u_2] & \text{if } d_1 \cap d_2 \neq \emptyset \\ \emptyset & \text{(not defined) otherwise} \end{cases}$$

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 413–422, 2012.

This product in $\mathcal{D}$ is closed because it is a continuous duration with two end points that are also integer numbers, but it is not commutative.

$$d_1 \sqcap d_2 = \begin{cases} [l_1, u_2] & \text{if } d_1 \cap d_2 \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

$$d_2 \sqcap d_1 = \begin{cases} [l_2, u_1] & \text{if } d_1 \cap d_2 \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

We will explain the meaning of this product later when we make a connection to the duration automata. The next are some basic results needed to investigate the associative property of the product.

Theorem 1. *If* $d_1, d_2, d_3 \in \mathcal{D}$ *and* $d_1 \cap d_2 \neq \emptyset$, $d_2 \cap d_3 \neq \emptyset$ *then* $d_1 \sqcap (d_2 \sqcap d_3) = (d_1 \sqcap d_2) \sqcap d_3$.

Proof. Suppose $d_1 = [l_1, u_1],\ d_2 = [l_2, u_2],\ d_3 = [l_3, u_3]$. For two durations, there are 4 cases of non-empty intersection. Because we only consider the relationship between $(d_1,\ d_2)$ and $(d_2,\ d_3)$, there are $2^4 = 16$ cases in total. This theorem can be proved by directly checking these 16 cases. Among 16 cases, there are 2 cases in which both left and right associated products are empty results.

Theorem 2. *Let* d_1, d_2, d_3 *be durations in* $\mathcal{D}$. *If* $(d_1 \sqcap d_2) \sqcap d_3 \neq \emptyset$ *and* $d_1 \sqcap (d_2 \sqcap d_3) \neq \emptyset$ *then* $(d_1 \sqcap d_2) \sqcap d_3 = d_1 \sqcap (d_2 \sqcap d_3)$.

Proof. By remarking the relationship between left and right bounds of 3 durations $d_1 = [l_1, u_1], d_2 = [l_2, u_2], d_3 = [l_3, u_3]$, we only take care of the order of these $l_i, u_i,\ i = 1..3$. Thus, we can choose a representative value in each duration. From logical aspect, there are limited cases, so we can use discrete checking method by regarding $l_i, u_i,\ i = 1..3$, as integer numbers in durations. For a duration $[l_1, u_1]$, there are at most 16 possibilities to put 2 points $l_2, u_2\ (l_2 \leq u_2)$ into the intervals defined by l_1 and u_1. For each 4 points configuration of interval defined by $l_1,\ u_1,\ l_2,\ u_2$ with random order ($l_1 \leq u_1, l_2 \leq u_2$), we have at most 49 cases to put $l_3,\ u_3$ into these intervals. Because the role of l_1, l_2, l_3 and u_1, u_2, u_3 are the same, hence the values of these end points can be considered as integer numbers from 1 to 14. The truth of this theorem can be proved by running a simple program to check for all cases as follow. The results of running a simple program as shown below give us a confirmation for the proof of this theorem.

Let $\mathcal{D} = \{[l, u] \mid l, u \in [1, 14], l \leq u, l, u \in \mathbb{Z}^+\}$ be a set of durations with integer end points and values from 1 to 14. The pseudo code below is to check for associative property of the product of 3 durations as mentioned in the theorem 2:

```
Function check() {
ok=true;
for (d1 ∈ D) and ok
    for (d2 ∈ D) and ok
        for (d3 ∈ D) and ok
          if prod(prod(d1, d2),d3) ≠ ∅ and prod(d1,prod(d2, d3)) ≠ ∅ then
            if prod(prod(d1, d2),d3) ≠ prod(d1,prod(d2, d3)) then ok=false;
return(ok); }
```

Remark: Function $prod$ will return the product of two durations.

In the following example, we present the cases that may happen between the product of 3 durations when checking for associative property.

Example 1.

$([2,5] \sqcap [4,6]) \sqcap [1,3] = [2,6] \sqcap [1,3] = [2,3]$ but $[2,5] \sqcap ([4,6] \sqcap [1,3]) = [2,5] \sqcap \emptyset = \emptyset$

$([4,6] \sqcap [1,3]) \sqcap [2,5] = \emptyset \sqcap [2,5] = \emptyset$ but $[4,6] \sqcap ([1,3] \sqcap [2,5]) = [4,6] \sqcap [1,5] = [4,5]$

$$([1,3] \sqcap [2,5]) \sqcap [4,6] = [1,5] \sqcap [4,6] = [1,6]$$

and

$$[1,3] \sqcap ([2,5] \sqcap [4,6]) = [1,3] \sqcap [2,6] = [1,6]$$

$$([1,2] \sqcap [3,5]) \sqcap [4,6] = \emptyset \sqcap [4,6] = \emptyset$$

and

$$[1,2] \sqcap ([3,5] \sqcap [4,6]) = [1,2] \sqcap [3,6] = \emptyset$$

We can see that if the products of 3 durations with some associations are defined, then they are equal. A question raises: is the defined product of a given sequence of durations with any association unique? The answer as follows.

Theorem 3. *Suppose* $d_1 = [l_1, u_1],\ d_2 = [l_2, u_2],\ \ldots, d_n = [l_n, u_n]$. *If a product* δ *of* $d_1, d_2, \ldots, d_n$ *with an association is defined then* $\delta = [l_1, u_n]$.

Proof. We will prove by induction on n.
- For the case of $n = 2$, this confirmation is true.
- Suppose the confirmation is true for $n = k > 2$ durations. We denote the product of a sequence of durations for some associations as $\delta_k = [d_1, d_2, \ldots d_k]$. In case this product is defined, the product is unique, otherwise it is $\emptyset$. For $x_k = d_1 d_2 \ldots d_k,\ d_i = [l_i, u_i], i = 1..k$, $\delta_k = [l_1, u_k]$. Suppose $d = [l, u] \in \mathcal{D}$, there are 3 possibilities to multiply d with x_k.
+ Left multiply: $d \sqcap x_k = [l, u] \sqcap [l_1, u_k] = [l, u_k]$, if it is defined.
+ Right multiply: $x_k \sqcap d = [l_1, u_k] \sqcap [l, u] = [l_1, u]$ if it is defined.
+ Middle multiply: Suppose the sequence $x_{k+1} = d_1 \ \ldots \ d_i \ d \ d_{i+1} \ldots \ d_k$ with a product for some association $\delta_{k+1} = [d_1 \ \ldots \ d_i \ d \ d_{i+1} \ldots \ d_k]$ is defined. Because δ_{k+1} is defined, there is at least one separation such that $\delta_l = [d_1, \ldots, d_i, d]$, $\delta_r = [d_{i+1}, \ldots, d_k]$ are both defined and $d_l \sqcap d_r \neq \emptyset$. $\delta_l \sqcap d = [l_l, u]$, $\delta_r = [l_{i+1}, u_k]$ and $d_l \sqcap d_r \neq \emptyset \Rightarrow [l_l, u] \cap [l_{i+1}, u_k] \neq \emptyset \Rightarrow d_l \sqcap d_r = [l_l, u] \sqcap [l_{i+1}, u_k] = [l_1, u_k]$.

Note that the new duration can be a result of no more than other k durations.

Corollary 1. *If δ_1 and δ_2 are two products of a given sequence $d_1, d_2, \ldots, d_n$, which are defined for any two associations then $\delta_1 = \delta_2$.*

Corollary 2. *If the product of a sequence $d_1, d_2, \ldots, d_n$ for any association is defined, then the result is a continuous duration.*

Example 2. Consider the sequence $\widetilde{x} = [1,6][5,8][2,4][3,7]$:

$$(([1,6] \sqcap [5,8]) \sqcap [2,4]) \sqcap [3,7] = ([1,8] \sqcap [2,4]) \sqcap [3,7] = [1,4] \sqcap [3,7] = [1,7]$$
$$[1,6] \sqcap ([5,8] \sqcap ([2,4] \sqcap [3,7])) = [1,6] \sqcap ([5,8] \sqcap [2,7]) = [1,6] \sqcap [5,7] = [1,7]$$
$$[1,6] \sqcap ([5,8] \sqcap [2,4]) \sqcap [3,7] = [1,6] \sqcap \emptyset \sqcap [3,7]) = \emptyset \text{ (not defined)}$$

We can see in example 2 that if two defined products of the same sequence with different associations then they are equal.

Below, we look at some natural orders which are defined by left bound, right bound, bandwidth orders, and combination of these to provide complex orders.

Definition 2. *Given $d_1 = [l_1, u_1]$ and $d_2 = [l_2, u_2]$ are two durations. We define new weak orders as below*
- Left bound order: $d_1 <_l d_2$ iff $l_1 \leq l_2$
- Right bound order: $d_1 <_r d_2$ iff $u_1 \leq u_2$
- Bandwidth order: $d_1 <_b d_2$ iff $(u_1 - l_1) \leq (u_2 - l_2)$
And combination orders:
- $d_1 <_{lr} d_2$ iff $(l_1 < l_2) \vee (l_1 = l_2 \wedge u_1 \leq u_2)$
- $d_1 <_{rl} d_2$ iff $(u_1 < u_2) \vee (u_1 = u_2 \wedge l_1 \leq l_2)$
- $d_1 <_{lb} d_2$ iff $(l_1 < l_2) \vee (l_1 = l_2 \wedge u_1 - l_1 \leq u_2 - l_2)$
- $d_1 <_{bl} d_2$ iff $(u_1 - l_1 < u_2 - l_2) \vee ((u_1 - l_1 = u_2 - l_2) \wedge l_1 \leq l_2)$
- $d_1 <_{br} d_2$ iff $(u_1 - l_1 < u_2 - l_2) \vee ((u_1 - l_1 = u_2 - l_2) \wedge u_1 \leq u_2)$
- $d_1 <_{rb} d_2$ iff $(u_1 < u_2) \vee (u_1 = u_2 \wedge u_1 - l_1 \leq u_2 - l_2)$
- By mean order: $d_1 <_m d_2$ iff $u_1 - l_1 \leq u_2 - l_2 \vee (u_1 - l_1 = u_2 - l_2 \wedge l_1 \leq l_2)$

Depending on the criteria of the optimization problems, we can choose a weak order among those (or define a new one) as a comparison operator for selecting a good (or the best) solution(s) .

3 Duration Language

Definition 3. *A duration automaton (DA) is a tuple $M = \langle S, \Sigma, \triangle, \nabla, q, R, F \rangle$ where:*

- *S is a finite set of states. $q \in S$ is an initial state.*
- *$\Sigma, \triangle, \nabla$ are internal, input and output alphabets of actions (or labels). We denote the set of actions of the DA by $\mathcal{A} = \Sigma \cup \triangle \cup \nabla$. There is an empty action $\varepsilon \in \Sigma$.*
- *$R \subseteq S \times \mathcal{A} \times \mathcal{D} \times S$ is a set of transitions, where $\mathcal{D}$ is shown in the section 2. For each transition $e = (s, a, d, s') \in R$, the label a will be the output action of the state s and the input action of the state s' as well. If $s = s'$ then e is a loop.*
- *$F \subseteq S$ is a set of final (or accepted) states.*

A configuration of M is a couple (s,t), where $s \in S$, $t \in \mathbb{R}^+$, which shows that the automaton M reaches the state s and stays there at the time t. For a state s, we denote $\Sigma(s), \triangle(s), \nabla(s)$ are internal, input and output actions of the state s respectively. The initial configuration of M is $(q,0)$. As the time passes, the changes of M are of the following forms:
- *Time-change*: $(s,t) \xrightarrow{a,\sigma} (s,t+\sigma)$ where $\sigma \in \mathbb{R}^+$, $a \in \Sigma(s)$. Automaton stays at the state s and does its *internal* actions.
- *State-change:* $(s,t) \xrightarrow{a,\sigma} (s',t+\sigma)$ where $\sigma \in \mathbb{R}^+, a \in \triangle(s)$, using a transition $e = (s,a,d,s') \in R(M)$, $\sigma \in d$. The transition e can take place if the time constraint d is satisfied, and we say e take M from the state s to the state s'. We make an extra assumption that *internal* actions can finish in a small enough time thus it can be ignored.

Definition 4. *Let M be a DA and $\widetilde{p}$ be a durations sequence $(s_0, d_0 = [0,\infty))(s_1,d_1)(s_2,d_2)\ldots(s_n,d_n)$, denoted by $\widetilde{p} = (s_i,d_i)_{i=0..n}$ in short, be a sequence of changes.*
- If s_0 is the initial state, and for each $i:\ 1 \le i \le n,\ \exists e_i \in R : e_i = (s_{i-1}, a_i, d_i, s_i)$ which makes M move from the state s_{i-1} to the state s_i, then $\widetilde{p}$ is called a d-path from s_0 to s_n in M.
- Suppose $\widetilde{p}$ is a d-path. If there is a strictly increasing sequence $t_0 = 0, t_1, t_2, \ldots, t_n$ such that for all $i : 1 \le i \le n$, $t_i - t_{i-1} \in d_i$ then the sequence $\overline{p} = (s_0,0)(s_1,t_1)(s_2,t_2)\ldots(s_n,t_n)$, denoted by $(s_i,t_i)_{i=0..n}$ in short, is called a t-path satisfying $\widetilde{p}$. In case $s_n \in F$, $\widetilde{p}$ is called a successful path of M; the sequence $\widetilde{w} = (a_1,d_1)(a_2,d_2)(a_3,d_3)\ldots(a_n,d_n)$, denoted by $(a_i,d_i)_{i=1..n}$ in short, where a_i is the action in the transaction e_i of $\widetilde{p}$, is a d-word and the sequence $\overline{w} = (a_1,t_1)(a_2,t_2)(a_3,t_3)\ldots(a_n,t_n)$, denoted by $(a_i,t_i)_{i=1..n}$ in short, is called a t-word satisfying $\widetilde{w}$. Each couple $(a_i,d_i) \in (\mathcal{A},\mathcal{D})$, $\varepsilon_\infty = (\varepsilon,[0,+\infty))$ and $(a_i,t_i) \in (\mathcal{A},\mathbb{R}^+)$ are called a d-label, the empty word and a t-label in turn. Each $x = (a_1\ldots a_k,d) \in (\mathcal{A}^*,\mathcal{D})$ and $y = (a_1\ldots a_k,t) \in (\mathcal{A}^*,\mathbb{R}^+)$ are called a d-string and a t-string in succession.
- A d-word $\widetilde{w}$ is acceptable (or d-word of M) if there is at least one t-word $\overline{w}$ satisfying $\widetilde{w}$, otherwise, we say $\widetilde{w}$ unacceptable. When $\widetilde{w}$ is acceptable, we say $\overline{w}$ takes M from the state s_0 to the state s_n and $\widetilde{w}$ can take M from the state s_0 to the state s_n.
- The d-word $\widetilde{w}$ is accepted by M if it is a d-word of a successful path. A set of all accepted d-words of M is called a d-language of M, denoted by $\mathcal{L}(M)$.

Definition 5. *Given $(a,d_1),(a',d_2)$ are two d-labels and $x = (a_1a_2\ldots a_n, d_1'), y = (b_1b_2\ldots b_m, d_2')$ are two d-strings. The products of two d-labels and two d-strings are given by:*
- $(a,d_1).\varepsilon_\infty = (a,d_1)$
- $(a,d_1).(a',d_2) = (aa', d_1 \sqcap d_2)$
- $x.\varepsilon_\infty = x$
- $x.y = (a_1a_2\ldots a_nb_1b_2\ldots b_m, d_1' \sqcap d_2')$

Remark: $(aa',\emptyset)$ and $(a_1\ldots a_k,\emptyset)$ are d-strings without satisfying t-strings. Furthermore, if we remove the time constraints, these products will become the traditional products of two strings in the automata theory.

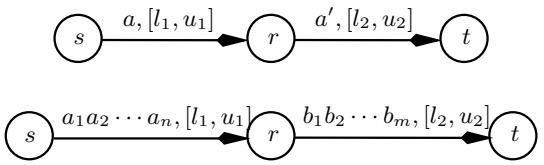


Fig. 1: Duration product of two d-labels and two d-strings

Product of two d-labels (or two d-strings) shows that if an action happens in two sequential phases: on the first phase, it moves from the state s to the state r under the duration constraint $[l_1, u_1]$, in the next phase, it moves from the state r to the state t under the duration constraint $[l_2, u_2]$ (see Fig. 1) then $[l_1, u_1]$ and $[l_2, u_2]$ must be intersected, and the action can not be started earlier than l_1, finished later than u_2.

Definition 6. *Given* $x \in \mathcal{A}^*$, $\widetilde{x} = (a_1, d_1)(a_2, d_2)\dots(a_n, d_n) \in (\mathcal{A}^*, \mathcal{D})^*$. *The left and right meaning functions* φ_l *and* φ_r *are defined by:*

$$\begin{aligned}
\varphi_l, \varphi_r &: (\mathcal{A}^*, \mathcal{D})^* \to (\mathcal{A}^*, \mathcal{D}) \\
\varphi_l((x, d)) &= (x, d) \\
\varphi_l(\widetilde{x}(a, d)) &= \varphi_l(\widetilde{x}) \sqcap (a, d) \\
\varphi_r((x, d)) &= (x, d) \\
\varphi_r((a, d)\widetilde{x}) &= (a, d) \sqcap \varphi_r(\widetilde{x})
\end{aligned}$$

Suppose $\widetilde{x} = (a_1, d_1)(a_2, d_2)\dots(a_n, d_n)$ *and* $\pi(d_1, d_2, \dots d_n)$ *is an arbitrary association of the sequence* $d_1, d_2, \dots d_n$ *which gives us a duration or undefined. We call* $\varphi_\pi(\widetilde{x}) = (a_1 a_2, \dots\ a_n, \pi(d_1, d_2, \dots d_n)$ *is the meaning of* $\widetilde{x}$ *with association* π, *where* $\pi(d_1, d_2, \dots d_n)$ *is the product of the sequence* $d_1, d_2, \dots d_n$ *with the association* π.

Lemma 1. *If a d-word* $\widetilde{x}$ *is acceptable then* $\varphi_r(\widetilde{x}) = \varphi_l(\widetilde{x})$.

Proof. Suppose $\widetilde{x} = (a_1, d_1)(a_2, d_2)\dots(a_n, d_n)$.

$$\begin{aligned}
\varphi_l(\widetilde{x}) =& \varphi_l((a_1, d_1)(a_2, d_2)\dots(a_n, d_n)) \\
=& \varphi_l((a_1, d_1)(a_2, d_2)\dots(a_{n-1}, d_{n-1})) \sqcap (a_n, d_n) \\
=& (\varphi_l((a_1, d_1)(a_2, d_2)\dots(a_{n-3}, d_{n-3})(a_{n-2}, d_{n-2})) \\
& \sqcap (a_{n-1}, d_{n-1})) \sqcap (a_n, d_n) \\
=& \varphi_l((a_1, d_1)(a_2, d_2)\dots(a_{n-3}, d_{n-3})(a_{n-2}, d_{n-2})) \\
& \sqcap ((a_{n-1}, d_{n-1}) \sqcap (a_n, d_n)) \text{ (theorem 1)} \\
=& (\varphi_l((a_1, d_1)(a_2, d_2)\dots(a_{n-3}, d_{n-3})) \sqcap (a_{n-2}, d_{n-2})) \\
& \sqcap \varphi_r((a_{n-1}, d_{n-1})(a_n, d_n))
\end{aligned}$$

$$
\begin{aligned}
=&(\varphi_l((a_1,d_1)(a_2,d_2)\dots(a_{n-3},d_{n-3})(a_{n-3},d_{n-3}))\\
&\sqcap(a_{n-2},d_{n-2}))\sqcap\varphi_r((a_{n-1},d_{n-1})(a_n,d_n))\\
=&\varphi_l((a_1,d_1)(a_2,d_2)\dots(a_{n-3},d_{n-3})(a_{n-3},d_{n-3}))\\
&\sqcap\varphi_r((a_{n-2},d_{n-2})(a_{n-1},d_{n-1})(a_n,d_n))\\
=&\dots\\
=&\varphi_r((a_1,d_1)(a_2,d_2)\dots(a_n,d_n))=\varphi_r(\widetilde{x})
\end{aligned}
$$

4 A Shortest D-Path Problem

Problem: Given a duration automaton $M\langle S,\Sigma,\triangle,\nabla,q,R,F\rangle$. s and t are two states of M, find an optimal path (using a given weak order for comparison) from the state s to the state t.

Remark: Due to the theorem 3, we can define the product of a sequence of durations before taking the product. We can see in the Fig. 2a), any path that going out of s on e_1 and coming to t on e'_1 will have the duration product as $[l_1,u'_1]$ if that product defined. Among all d-paths from the state s to the state t, there are some successful paths but some are not. Even though we know the product in advance, but we don't know whether it is a product of a successful path or not. Thus, we can sort all the products we may have from the best to the worse to make a list, then we try to find if there is a successful path corresponding to the product in the sorted list. In the Fig. 2, we will sort the arc lists $L_s = e_1,e_2,\dots,e_k$ and $L_t = e'_1,e'_2,\dots,e'_l$ such that if we take the product for each $e \in L_s$ (from the first to the last element) with each $e' \in L_t$ (from the first to the last element), these products follow the given order. The trying order for finding a successful path is from the best to the worse in the products list, therefore the first successful path will be the best solution. All other the best solutions will be the same in values, so the algorithm can stop.

We will develop an algorithm which uses graph technique. An arc-graph $G_a = (V,E)$ based on the automaton is built as follows:

- Vertexes: Each arc of the automaton M corresponds with a vertex in the G_a. Use the arcs of automaton to name these vertexes.

$$V=\{e_M \mid e_M \in E_M\}$$

where is E_M is the set of arcs of the duration automaton M.

- Arcs: If two adjacent arcs of the automaton M with non empty intersection between two constraints on the arcs then between two corresponding vertexes of G_a, there is an arc from the first vertex to the second.

$$E=\{(e,e') : e=(s,a,d,s'),e'=(s',a',d',s'');e,e' \in E_M \mid d\cap d' \neq \emptyset\}$$

By using G_a, the above problem leads to a new one: finding paths between the vertex s to the vertex t in the G_a. In the automaton M, there may exist many arcs going out from the state s, and there may exist many arcs coming to the state t. So that, in the arc-graph, we have to start from different vertexes (corresponding to the arcs going out of the state s) and go to the different vertexes in the arc-graph (corresponding to the

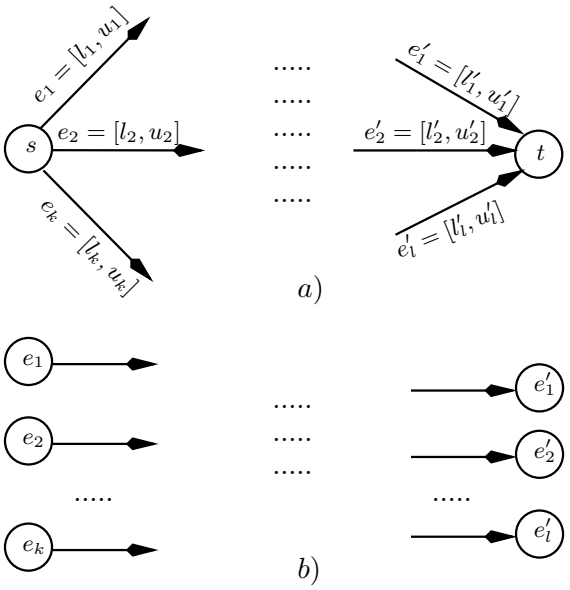


Fig. 2: Many going out and coming in arcs

arcs coming to the state t) (see in the Fig. 2). Because we can have the results of the products before finding a real d-path for the different tries, we will sort the arcs going out from the s and the arcs coming to the t, corresponding to the start and the terminal vertexes in the G_a, using the given order, such that the products of an arc going out from the state s with an arc coming to the state t follow the given order. The comparison operator depends on the requirements of the problem. We can use these orders for trying to find the best solutions using the orders we have prepared. Be aware that each path in G_a is an acceptable d-word in the automaton M.

Algorithm to find the best path as follows:

Input: Automaton $M = \langle S, \Sigma, \triangle, \nabla, q, R, F\rangle$, two states s, t and the order on durations for comparing.

Output: One of the best path.

1. Build these lists: $L_s = [e \mid \exists s' \in S, a \in \mathcal{A}, d \in \mathcal{D} : e = (s, a, d, s') \in R]$,
   $L_t = [e \mid \exists s \in S, a \in \mathcal{A}, d \in \mathcal{D} : e = (s, a, d, t) \in R]$.
2. Sort L_s, L_t such that the products of a duration in L_s with a duration in L_t follows given order.
4. Build arc-graph G_a based on M.
5. Choose a start vertex in L_s, a terminal vertex in L_t in sorted order.
6. On G_a, using a depth first search algorithm to find a path between that two vertexes. If there exists a path, stop algorithm and return the path; otherwise try an other vertexes couple in step 5 until the end of the both lists.
7. If all possibilities have been tried without finding suitable path then return($(\varepsilon, \emptyset)$) (there is no acceptable path between the two).

The algorithm travels each arc of M no more than one time, so that the complexity is $\mathbf{O}(m)$. This problem can be extended for the case of finding the optimal path(s) which pass the smallest number of arcs (or vertexes in G_a). This problem is also solved by finding the shortest paths between two vertexes in G_a and comparing if the products of durations constraints on the start and the terminal vertexes are equal to the optimal one. If the answer is yes, then it is an optimal path which passes the smallest number of arcs.

Example 3. Finding an optimal path from the state s to the state t in the automaton below as shown in the Fig. 3, using $\leq_m$ order.
The trying orders of outgoing arcs from the state s should be $1, 2, 3$, and the trying orders of the incoming arcs to the state t should be $11, 10$.

We can see the corresponding arc-graph in the Fig. 4: start vertexes $L_s = [1, 2, 3]$ and the target vertexes $L_t = [11, 10]$ after sorting. The path $[1, 5, 9, 10]$ is thea first satisfied founded path in the arc-graph, and the algorithm can stop with the product result is $[1, 10]$. All other paths will have the product which is greater than or equal to (using the given order) the optimal one, such as $([2\ 9\ 10], [2, 10])$: $([1\ 5\ 9\ 10], [1, 10]) \leq_m ([2\ 9\ 10], [2, 10])$.

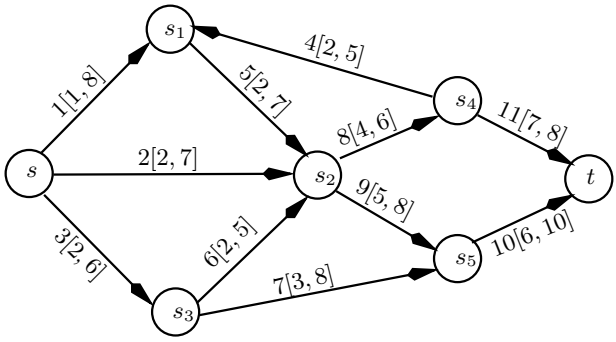


Fig. 3: An DA example

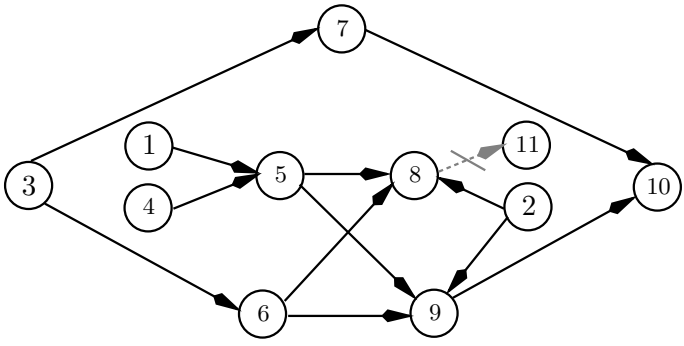


Fig. 4: Arc-graph

5 Conclusion and Future Work

This paper introduced the term of ordered durations and built the algebra on durations. Using duration algebra, we propose algorithm to find the optimal paths on the automaton using different comparison orders with time complexity as $\mathbf{O}(m)$ where m is the number of arcs. Depending on the problem, we can choose a suitable order or define a new one and using this approach to find the best solutions.

References

1. Bui Vu Anh: Duration automaton in scheduling programs for a cluster computer system. Journal of Computer Science and Cybernetics 27(3), 218–228 (2011)
2. Bui Vu Anh: A schedule algorithm for works with uncertainly finish time on a cluster computer. Addendum of RIVF 2012 (Poster), pp. 40–44 (2012)
3. Van Hung, D., Bui Vu Anh: Model Checking Component Based Systems with Black-box Testing. Technical report at International Institute for Software Technology, United Nation University, Macao, pp. 76–79. Published in RTCSA (2005); 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2005) (2005) ISSN: 1533-2306, ISBN: 0-7695-2346-3
4. Bui Vu Anh: Nondeterministic duration automata in modeling priority network. In: Proceeding of National Conference on Discovery Knowledge from Data, Vietnam, August 5-6, vol. 210156B00, pp. 315–325 (2009)
5. Merritt, M.: Time-Constrained Automaton. In: Groote, J.F., Baeten, J.C.M. (eds.) CONCUR 1991. LNCS, vol. 527, pp. 408–423. Springer, Heidelberg (1991)
6. Alur, R.: A theory of Timed Automata (1999)
7. Alur, R., Henginger, T.A.: A really temporal logic. In: Proceedings of 30th IEEE Symposium on Foundation of Computer Science (FOCS 1989), pp. 164–169 (1989); Extended version appeared in The Journal of the ACM 41, 181–204 (1994)
8. Lynch, N., Tuttle, M.: An Introduction to Input/Output automata. CWI-Quarterly 2(3), 219–246 (1989)

Study of the Migration Scheme Influence on Performance of A-Teams Solving the Job Shop Scheduling Problem

Piotr Jędrzejowicz and Izabela Wierzbowska

Department of Information Systems, Gdynia Maritime University,
Morska 81-87, 81-225 Gdynia, Poland
{pj,iza}@am.gdynia.pl

Abstract. The paper investigates the impact of migration scheme on performance in systems with A-Teams working in parallel, in the architecture designed for solving difficult combinatorial optimization problems and used for solving Job Shop Scheduling Problem. A-Teams, or islands, belonging to the team of A-Teams, cooperate through exchange of intermediary computation results, following certain migration scheme. The number of results forwarded from one A-Team to another, e.g. migration size, is experimentally investigated in combination with different numbers and sizes of the islands.

Keywords: Parallel A-Teams, agent cooperation, combinatorial optimization, migration parameters, JSS.

1 Introduction

As it has been observed in [2] the techniques used to solve difficult combinatorial optimization problems have evolved from constructive algorithms to local search techniques, and finally to population-based algorithms. 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 [13], [14], [15], [16].

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 [3], [7] and [9]. The JABAT middleware was built with the use of JADE (Java Agent Development Framework), a framework proposed by TILAB [5]. JABAT complies with the requirements of the next generation A-Teams

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 423–432, 2012.

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 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 [11] JABAT environment has been extended through integrating the team of asynchronous agent paradigm with the island-based genetic algorithm concept first introduced in [6]. A communication, that is information exchange, between cooperating A-Teams (islands) has been introduced. It has been shown [4] that applying this model may have a positive impact on the results.

The impact of the topology on computation results in other models was considered in, for example, [12], where the Island Model was considered. Also, in [10] several known migration models have been compared. It has been shown, that a model called *Randomized*, outperforms other known models. This paper investigates how the choice of the migration size, in combination with different numbers and sizes of the islands, influences results obtained by the Team of A-Teams solving instances of the Job Shop Scheduling Problem.

The paper is constructed as follows: Section 2 describes an implementation of the Team of A-Teams (*TA-Teams*) concept. Section 3 gives details of the specialised *TA-Teams* implementation, designed to solve instances of Job Shop Scheduling Problem, as well as computational experiment settings. Section 4 contains results of the computational experiment that has been subsequently carried out. Finally some conclusions and suggestions for future research are drawn.

2 Team of A-Teams Implementation

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 one of the current solutions kept in the A-Team common memory, and attempts to improve it. Afterwards, successful or not, the result is sent 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. The process is supervised by the agent called *SolutionManager*.

In JABAT a number of A-Teams may run in parallel providing the required computational resources are available (Fig. 1). If these A-Teams solve the same task, by exploring different regions of the search space, then the added process of communication capability 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. The A-Teams with the added communication are called islands. 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, [17] or [19].

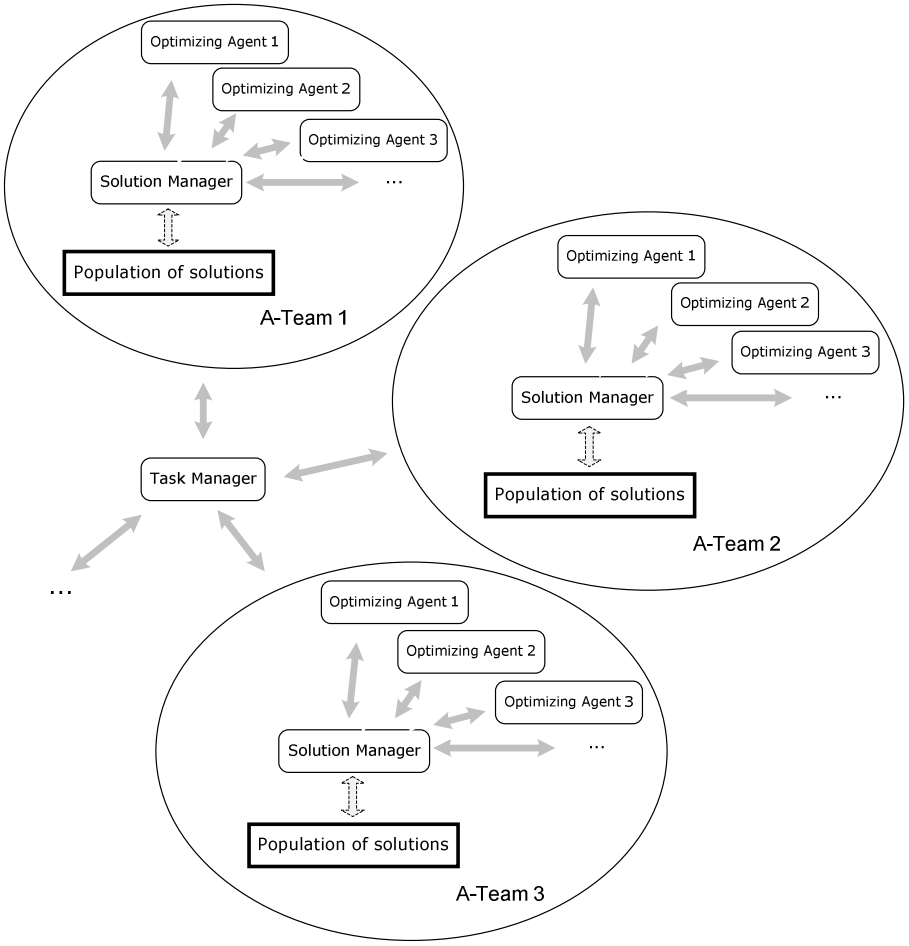


Fig. 1. JABAT with several A-Teams

In the discussed JABAT implementation of the *TA-Teams* concept the process of communication between common memories is supervised by a specialized agent called *MigrationManager* controlling execution of the particular migration scheme. Such a scheme consists of the following settings:

– *Migration architecture* – an architecture defining which A-Team receives communication from another A-Team, which A-Team it sends communication to, and when (or how often) the communication takes place.

- *Migration size* – number of individuals sent between common memories of A-Teams in a single cycle,
- *Migration policy* – a rule determining how the received solution is incorporated into the common memory of the receiving A-Team.

The migration used in JABAT *TA-Teams* is asynchronous. *MigrationManager* sends messages to islands, pointing out to which islands current best solution should be sent to, then each A-Team (that is an island) receiving *MigrationManager* message, after reading it, sends current best solution to indicated island or islands.

3 Team of A-Teams Solving the Job Shop Scheduling Problem and Computational Experiment Design

3.1 Job Shop Scheduling Problem

For the purpose of this paper computation results obtained from solving one of the best known combinatorial optimization problems - Job Shop Scheduling problem (JSS) using a dedicated *TA-Teams* have been analyzed and compared.

In the job shop scheduling problem n jobs $J_1, J_2, ..., J_n$ of varying sizes are to be scheduled on m identical machines. The resulting schedule should have the minimal makespan, defined as the total length of the schedule.

In the computational experiment *TA-Teams* implementation has been used to solve several instances of JSS, namely ft10, ft20, la16, la18 and la20. All instances have been taken from well-known benchmark dataset library: OR-library [18].

3.2 Optimizing Agents

To solve instances of JSS the following optimization algorithms has been used as the inner algorithms of optimizing agents implemented within the system:

- 3-opt algorithm,
- path relinking algorithm,
- harmony search algorithm,
- algorithm recombinating two input solutions,
- crossover and mutation algorithms.

3.3 Working Strategy

The process of solving a single task 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, which contains a population of solutions called individuals.

In [2] a set of different strategies has been investigated with respect to selecting solutions to be improved and replacing the solutions stored in the common memory by the improved ones. The respective strategies are shown in Table 1.

Table 1. The working strategies investigated in [2]

Strategy	Procedures used
RM-RR	Random move selection + Random replacement
RM-RW	Random move selection + Replacement of the worse
RM-RE	Random move selection + Replacement of the worse with exchange
RM-SUS	Random move selection + Addition with the stochastic universal sampling
RM-TS	Random move selection + Addition with the tournament selection
RB-RR	Random move selection with blocking + Random replacement
RB-RW	Random move selection with blocking + Replacement of the worse
RB-RE	Random move selection with blocking + Replacement of the worse with exchange
RB-SUS	Random move selection with blocking + Addition with the stochastic universal sampling
RB-TS	Random move selection with blocking + Addition with the tournament selection

To choose the best working strategy for the JSS problem implementation an experiment was carried out, in which a single A-Team with population of 500 individual solutions was used to solve two instances: ft10 and ft20. The results are shown in Table 2 and it appears that in terms of respective relative errors the most promising strategy, generating solutions of good quality, is RM-RW. Following the above finding, the RM-RW strategy has been chosen for further computations.

Table 2. Results for different working strategies, one A-Team

Strategy	ft10	ft20	avg
RM-RR	3,80%	2,24%	3,02%
RM-RW	2,73%	1,27%	2,00%
RM-RE	3,36%	1,64%	2,50%
RM-SUS	6,80%	7,49%	7,15%
RM-TS	2,68%	1,70%	2,19%
RB-RR	3,34%	2,52%	2,93%
RB-RW	2,78%	1,40%	2,09%
RB-RE	2,81%	1,70%	2,25%
RB-SUS	6,30%	7,27%	6,79%
RB-TS	3,58%	1,69%	2,63%

Thus selected working strategy RM-RW has the following features:

- All individuals in the initial population of solutions are generated randomly, the individuals are feasible solutions of the instance to be solved.

- Selection of individuals for improvement is a random move.
- Returning individual replaces the first found worse individual.
- 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.

3.4 Migration Schemes

In [10] the migration scheme called *Randomized* has been shown as producing good results in comparison with other models. In *Randomized*, whenever an island needs a new solution, it sends appropriate message to *MigrationManager* and then receives the current best solution from another island, chosen at random by *MigrationManager*. An island asks for a new solution when the current best solution has not changed in half of the *no improvement time gap.*

There are two other migration scheme that may influence results: *migration size* and *migration policy*. In this paper we consider the following settings:

- *migration size* $\in \{1, 5, 10, 15\}$: in one cycle the given number of current best solutions is sent from the common memory of an A-Team to the common memory of another A-Team,
- *migration policy*: the best solutions, taken from the source population, are added to the destination population according to the working strategy. It means that in the case of considered strategy RM-RW a solution may replace a worse solution or, if there is no worse solution, may be rejected.

3.5 Other Settings

Different values of the numbers of islands and population sizes have been investigated. These settings are summarized in Table 3.In each case the total number of solutions within the system is close to 500.

Table 3. Island settings used in the experiment

Case	Number of islands	Population size
1	1	500
2	2	250
3	3	166
4	4	125

Case	Number of islands	Population size
5	5	100
6	6	83
7	8	62
8	10	50

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 machines and connecting them to the main platform. Then agents may migrate from the main platform to these containers. Each instance used in the reported experiment was solved with the use of 5 nodes of the cluster - one for the main platform and four for the optimising agents to migrate to.

For all runs for each setting from Table 3 and *migration sizes*, 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 at the end averaged.

4 Computational Experiment Results

In [1] and [8] similar ATEAM architectures have been considered and used for solving instances of the JSS problem. Table 4 compares these results, in terms of respective relative errors, to the results obtained in JABAT with a single A-Team with population of 500 solutions and with *TA-Teams* consisting of four A-Teams, each working on a population of 125 solutions and using 10 solutions to migrate.

Table 4. Results obtained from various A-Teams

Problem	Aydin' ATEAM [1]	PLA Team [8]	JABAT A-Team	JABAT *TA-Teams*
ft10	3.05%	2,37%	2,62%	**2,31%**
la16	**0.50%**	2,03%	1,06%	0,80%
la18	1,02%	0,79%	**0.41%**	**0,41%**
la20	**0,52%**	0,55%	0,62%	**0,52%**

Table 5 presents results obtained for different numbers of islands, populations sizes and migration sizes, and - in the last column- averaged over all tasks considered. The best results for each number of islands are highlighted in bold. The results differ little for consecutive settings. A single combination of the settings, best for all tasks under investigation, cannot be pointed out, though it may be seen that results for 4 and 5 islands are slightly better.

In case of *migration size* it also may be significant how the incoming solutions are incorporated into the target population. The results from Table 5 correspond to adding these solutions in accordance to the overall strategy RM-RW, discussed in Subsection 3.3. In this case returning individual replaces a worse individual, which means that some of the solutions send as the part of communication may be discarded – when in the target population no worse solution can be found. So, the strategy has been modified to mRM-RW, in which the incoming solutions (but only these coming from

the communication between islands) are prevented from being discarded and replace a random solution in the target population. Tables 6 an 7 present results of these two strategies for the cases of 4 islands with population of 125 and 5 islands of 100 solutions. and the results are slightly better, but the differences are indeed very small.

Table 5. Results obtained from *TA-Teams* with different settings of numbers of islands, and population and migration sizes

Islands	Population size	Migration size	ft10	la16	la18	la20	Avg
1	500	1	2,62%	1,06%	0,41%	0,62%	1,18%
2	250	1	2,20%	1,41%	0,60%	0,62%	1,21%
		5	2,89%	0,95%	0,31%	0,60%	1,19%
		10	2,58%	1,03%	0,42%	0,63%	1,17%
		15	2,43%	1,13%	0,43%	0,59%	1,15%
3	166	1	2,42%	1,25%	0,57%	0,69%	1,23%
		5	2,60%	0,99%	0,49%	0,67%	1,19%
		10	2,62%	1,12%	0,52%	0,58%	1,21%
		15	2,52%	**0,71%**	0,52%	0,62%	1,09%
4	125	1	2,19%	0,89%	0,37%	0,64%	1,02%
		5	**2,13%**	0,93%	0,59%	0,60%	1,06%
		10	2,31%	0,80%	0,41%	0,52%	**1,01%**
		15	2,18%	1,00%	0,38%	0,58%	1,04%
5	100	1	2,49%	0,80%	0,35%	0,55%	1,05%
		5	2,51%	0,75%	0,35%	0,61%	1,06%
		10	2,55%	1,05%	0,32%	0,55%	1,12%
		15	2,37%	1,16%	**0,26%**	0,55%	1,09%
6	83	1	2,27%	1,71%	0,52%	**0,52%**	1,26%
		5	2,43%	0,96%	0,47%	0,54%	1,10%
		10	2,59%	0,92%	0,52%	0,58%	1,15%
		15	2,48%	0,96%	0,37%	0,61%	1,11%
8	62	1	3,02%	1,07%	0,55%	0,61%	1,31%
		5	2,82%	1,02%	0,57%	0,56%	1,24%
		10	2,78%	1,23%	0,49%	0,62%	1,28%
		15	2,50%	1,43%	0,62%	0,62%	1,29%
10	50	1	3,03%	0,99%	0,49%	0,59%	1,27%
		5	2,57%	1,24%	0,36%	0,58%	1,19%
		10	3,14%	1,26%	0,51%	0,57%	1,37%
		15	2,90%	1,16%	0,41%	0,54%	1,25%

Table 6. Results for strategies mRW-RW and RW-RW, 4 islands

mRM-RW						RM-RW
Migration size	**la18**	**la20**	**ft10**	**la16**	**Avg**	**Avg**
1	**2,41%**	0,97%	0,45%	**0,59%**	**1,11%**	1,02%
5	2,64%	1,35%	**0,45%**	0,66%	1,27%	1,06%
10	2,57%	0,89%	0,47%	0,62%	1,14%	**1,01%**
15	2,73%	**0,84%**	0,49%	0,62%	1,17%	1,04%

Table 7. Results for strategies mRW-RW and RW-RW, 5 islands

mRM-RW						RM-RW
Migration size	**la18**	**la20**	**ft10**	**la16**	**Avg**	**Avg**
1	2,57%	**0,95%**	0,38%	0,56%	1,12%	**1,05%**
5	2,40%	1,51%	0,49%	0,60%	1,25%	1,06%
10	2,46%	1,06%	**0,30%**	0,55%	**1,09%**	1,12%
15	**2,36%**	1,02%	0,57%	**0,54%**	1,13%	1,09%

5 Conclusions

The discussed experiment results confirmed that the choice of the number of islands and population and migration sizes may influence results obtained by the Team of A-Teams, with the impact of the number of islands greater than that of the others considered settings. However, results suggest that the choice as to which settings should be chosen may be difficult, and even within a single problem (in our case Job Shop Scheduling) the best settings may be different for different instances.

The observations are valid only to one problem considered in this paper, that is Job Shop Scheduling, and cannot be generalised. Future research may focus on evaluating effects of the choice of migration parameters in solving other problems.

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: research project no. N N519 576438 for years 2010-2013.

References

1. Aydin, M.E.: Metaheuristic Agent Teams for Job Shop Scheduling Problems. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS (LNAI), vol. 4659, pp. 185–194. Springer, Heidelberg (2007)
2. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: Influence of the Working Strategy on A-Team Performance. In: Szczerbicki, E., Nguyen, N.T. (eds.) Smart Information and Knowledge Management. SCI, vol. 260, pp. 83–102. Springer, Heidelberg (2010)

3. Barbucha, D., Czarnowski, I., Jędrzejowicz, 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, ISDA, Jinan Shandong China, pp. 144–149. IEEE, Los Alamos (2006)
4. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: Parallel Cooperating A-Teams. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 322–331. Springer, Heidelberg (2011)
5. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper. Exp. 3(3), 6–20 (2003)
6. 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, Hillsdale (1987)
7. Czarnowski, I., Jędrzejowicz, 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)
8. Jędrzejowicz, J., Jędrzejowicz, P.: Agent-Based Approach to Solving Difficult Scheduling Problems. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 24–33. Springer, Heidelberg (2006)
9. 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, Part III. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
10. Jędrzejowicz, P., Wierzbowska, I.: Impact of Migration Topologies on Performance of Teams of A-Teams. In: Czarnowski, I., et al. (eds.) ABOSCI Book. Springer, Heidelberg (to appear, 2012)
11. Jędrzejowicz, P., Wierzbowska, I.: Parallel Cooperating A-Teams Solving Instances of the Euclidean Planar Travelling Salesman Problem. In: O'Shea, J., Nguyen, N.T., Crockett, K., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2011. LNCS, vol. 6682, pp. 456–465. Springer, Heidelberg (2011)
12. Ruciński, M., Izzo, D., Biscani, F.: On the impact of the migration topology on the Island Model. Parallel Computing 36(10-11), 555–571 (2010)
13. Talukdar, S.N.: Collaboration Rules for Autonomous Software Agents. Decision Support Systems 24, 269–278 (1999)
14. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Cooperation Schemes for Autonomous Agents. Journal of Heuristics 4(4), 295–321 (1998)
15. Talukdar, S.N., Pyo, S.S., Giras, T.: Asynchronous Procedures for Parallel Processing. IEEE Trans. on PAS PAS-102(11), 3652–3659 (1983)
16. Talukdar, S.N., de Souza, P., Murthy, S.: Organizations for Computer-Based Agents. Engineering Intelligent Systems 1(2), 56–69 (1993)
17. 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)
18. OR-library, http://people.brunel.ac.uk/~mastjjb/jeb/info.html
19. 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)

A New Cooperative Search Strategy for Vehicle Routing Problem

Dariusz Barbucha

Department of Information Systems
Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
barbucha@am.gdynia.pl

Abstract. Cooperation as a problem-solving strategy is a widely used approach to solving complex hard optimization problems. It involves a set of highly autonomous programs (agents), each implementing a particular solution method, and a cooperation scheme combining these autonomous programs into a single problem-solving strategy. It is expected that such a collective of agents can produce better solutions than any individual members of such collective. The main goal of the paper is to propose a new population-based cooperative search approach for solving the Vehicle Routing Problem. It uses a set of search procedures, which attempt to improve solutions stored in a common, central memory. Access to a single common memory allows exploitation by one procedure solutions obtained by another procedure in order to guide the search through a new promising region of the search space, thus increasing chances for reaching the global optimum.

Keywords: cooperative search, population-based methods, multi-agent systems, vehicle routing problem.

1 Introduction

In the recent years technological advances enabled development of various parallel and distributed versions of the hybrid methods with cooperation paradigm embedded in them for solving computationally difficult optimization problems [3]. *Cooperative search* consists of a search performed by agents that exchange information about states, models, entire sub-problems, solutions or other search space characteristics [3]. Although the cooperation search paradigm may take different forms, they all share two main features [4]: a set of highly *autonomous programs*, each implementing a particular solution method, and a *cooperation scheme* which combines these programs into a single consistent problem-solving strategy.

A set of autonomous programs may include exact methods, like for example branch and bound, but in most cases different approximate algorithms (local search, variable neighborhood search, guided local search, tabu search, etc.) are engaged in finding the best solution. A cooperation scheme, has to provide the

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 433–442, 2012.

mechanism for effective communication between autonomous programs allowing them to dynamically exchange the important pieces of information which next is used by each of them to support the process of search for a solution.

The key challenge of cooperation is to ensure that *meaningful* information is exchanged in a *timely* manner yielding a global parallel search that achieves a better performance than the simple concatenation of the results of the individual threads, where performance is measured in terms of computing time and solution quality [5]. Toulouse, Crainic, and Gendreau [15] have proposed a list of fundamental issues to be addressed when designing cooperative parallel strategies for meta-heuristics: What information is exchanged? Between what processes is it exchanged? When is information exchanged? How is it exchanged? How is the imported data used? Crainic and Toulouse [4] have completed the above list by adding the issue what each autonomous program does with the received information and whether new information and knowledge is to be extracted from the exchanged data to guide the search.

Cooperative search methods can be viewed as hybrid and/or parallel metaheuristics. Referring to the hierarchical taxonomy of hybrid metaheuristics presented by Talbi [14], it is easy to see that a class of *teamwork* hybrids represents cooperative search strategies. On the other hand, considering classification of parallel metaheuristics provided by Crainic and Toulouse [5], cooperative search strategies belong to the *pC/KS* or *pC/C* or *pC/KC* groups of parallel metaheuristics. It means that the global problem solving process is controlled by several processes, and they implement the information sharing cooperation mechanism specifying how the independent methods interact within the global search behavior. All cooperative strategies may start the search threads from the same or different solutions and may use of the same or different search strategies.

Several cooperative search schemes have been proposed last years for solving different optimization problems. An overview of the most representative of them the reader can find, for example, in [4] or [11].

This paper aims at proposing a new population-based cooperative approach for solving the vehicle routing problem, where a set of different search programs (agents) working in parallel collectively solve instances of the problem using a common central memory used for storing a pool of solutions. Additionally, an adaptive mechanism implemented in the approach alows one to define a few population management strategies which are dynamically changed during the search according to the performance of the approach at the current stage of computation. The presented approach is implemented in a multi-agent environment which provide a convenient and effective mechanism for solving the problem in parallel and for cooperation between agents. Section 2 defines the vehicle routing problems and relates to existing cooperative aproaches to solve it. Section 3 includes a description of the proposed approach. Computational experiment, which has been carried out in order to validate the proposed approach is presented in Section 4. And conclusions and suggestions of future work presented in Section 5 end the paper.

2 Vehicle Routing Problem

The classical Vehicle Routing Problem (VRP) 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 $i \in V \setminus \{0\}$ 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.

A review of the different variants of VRP and state-of-the-art methods for solving them, can be find, for example, in [7]. Among cooperative approaches dedicated for solving VRP two of them are worth mentioning here because of their similarity to those proposed in the paper.

The first one, proposed by Le Bouthillier et al. [9], presents a parallel cooperative multi-search method for the vehicle routing problem with time windows, in which several search threads cooperate by asynchronously exchanging information on the best solutions identified. The exchanges are performed through a solution warehouse mechanism, which holds and manages a pool of solutions. Each of these independent processes implements an evolutionary algorithm or a tabu search procedure.

In the second one, Meignan et al. [12] present a self-adaptive and distributed metaheuristic called Coalition-Based Metaheuristic, which is based on the Agent Metaheuristic Framework and hyper-heuristic approach. In their approach, several agents, forming a coalition, concurrently explore the search space of a given instance of VRP. Each agent modifies a solution with a set of operators. The selection of these operators is determined by heuristic rules dynamically adapted by individual and collective learning mechanisms.

3 A Cooperative Search Approach for VRP

The proposed cooperative search approach (CSA) for solving VRP belongs to the class of the cooperative population-based methods, and is implemented in a multi-agent environment [1]. Its main functionality focuses on organizing and conducting the process of search for the best solution with using a set of *search procedures* (implemented as software agents) executed in parallel, where each search program is an implementation of a single-solution method. During their

execution, the search procedures comunicate asynchronously with each other but the communication between them is performed indirectly via a common, sharable *memory* (also called warehouse or pool of solutions). Each individual stored in the memory is represented in a form that reflects the characteristics of the problem being solved, as well as which is convenient to handle the calculations performed on it by search procedures.

The whole process of search is organized as a sequence of steps, including initialization and improvement phases. At first the initial population of solutions is generated and stored in the memory. Next, at the following computation stages, individuals forming the initial population are improved by autonomously acting search procedures. A specialy designed program, called solution manager, acts as an intermediary between common memory and search programs. It maintains the common memory and its role is to read a particular individual from the memory and to send it periodically to search procedures, which have already announced their readiness to act, and to update the memory by storing in it a possibly improved solution obtained from the search program. Thus, the memory succesively evolves from the initially generated pool of solutions through intermediate trial solutions obtained during the search process up to the stage when the stopping criterion is met, and the best solution stored in the population is taken as the final solution of the given problem instance.

The above presented activities performed on memory are managed in accordance with the *population management strategy*, which defines: how the inital population is created and how many solutions does it include, how to choose solutions which are to be sent to the search programs for improvement, how to merge the improved solutions returned by the search procedures with the whole population and when to stop the process of searching?

Figure 1 presents the main part of the architecture of the proposed approach.

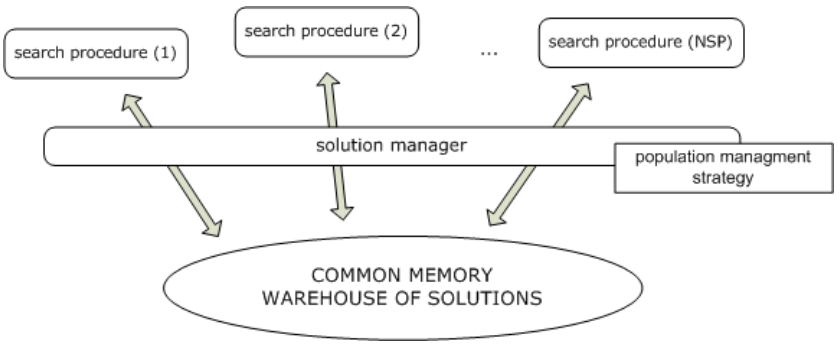


Fig. 1. Main part of the architecture of the proposed approach

In the proposed approach for VRP, each individual in population is a permutation *form* of N numbers (representing customers) with additional '0' delimiters denoting division of permutation into the routes. A part of individual between

'0' delimiters reflects the order in which customers are visited by one vehicle within selected route.

At the begining of search, the initial population of individual solutions is generated randomly. An initial random permutation of N numbers is next, completed by '0' delimiters in places which divide the permutation into the separeted parts (routes) assigned to each vehicle. The places for delimiters insertion is calculated in such a way that total capacity of vehicle assigned to the current route and the maximal route length are not exceeded. The process of creating the whole initial population is repeated until $popSize$ (population size) individuals have been generated.

Each individual from the population is evaluated using the *fitness* function, which value is calculated as a sum of the costs related to each permutation part (vehicle's route).

Four *search procedures* exploring the search space have been proposed:

- $SP(1)$ - an implementation of the *3-opt* procedure [10] operating on a single route.
- $SP(2)$ - a modified implementation of the dedicated *local search method* based on main features of λ-*interchange local optimization* method [13], where at most λ customers are moved or exchanged between two selected routes ($\lambda = 2$).
- $SP(3)$ - implementation of the dedicated *local search method* operating on two routes, and based on exchanging or moving selected customers between these routes. Selection of customers to be exchanged or moved is carried out in accordance with their distance to the centroid of their original route. First, a given number of customers from two selected routes for which the distance between them and the centroid of their routes are the greatest are removed from their original routes. Next they are moved to the opposite routes and inserted in them on positions, which give the smallest distance between newly inserted customers and the centroid of this route [2].
- $SP(4)$ - an implementation of the dedicated *local search method* based on moving or exchange the edges between two selected routes until feasible and improved solution is obtained [2].

Till now, all approaches dedicated for solving selected optimization problems, which has been implemented in multi-agent environment presented in [1], used a single population management strategy. Here, basing on the general assumptions about population management strategy and its role in the process of search for the best solution, it has been decided to implement an *adaptive mechanism*, where a few population management strategies are defined and which dynamically changes strategy during the search. A strategy which may assure the convergence to the best solution at the current stage of computation is selected and used in next stages. It is expected that such adaptation may influence the behavior of the proposed approach by diversification or intensification of process of solving the problem, thus increasing chances for reaching the global optimum.

For this reason, three *population management strategies* presented in Table 1 have been developed. Because of the fact that method of creating an initial

population is the same for all these strategies (as described earlier), and the population size has been set to 30 individuals in all strategies, the table includes only elements which differentiate strategies used in the approach.

Table 1. Population management strategies

Strategy(1) (default)	
Read/Select	random solution
Add/Replace	random solution
Stopping criterion	after 0.5 min.
Strategy(2)	
Read/Select	best solution
Add/Replace	worst solution
Stopping criterion	after 1 min.
Strategy(3)	
Read/Select	best solution
Add/Replace	random solution if last consecutive five solutions received from the optimization agents did not improve existing solutions in population, the worst solution is removed from the population and a newly generated one is added to the pool of individuals
Stopping criterion	after 0.5 min. without improvement

In the first (default) strategy, the step of reading/selecting an individual from the memory and next sending it to the optmising agents is implemented as a selection of a random solution. After improvement phase, if the solution currently received from optimization agent has been improved, it replaces random solution from current population. The process of search stops after 0.5 minute of computation. This default population management strategy guarantees a proper diversification of the population of solutions.

The second strategy aims at intensification of the process of search. At each time, when search program is ready to act, the current best solution from population is sent to it. After improvement, solution received from the search program replaces the worst one including in memory.

And the third one, decreases the intensification level, but in case of no improvement of existing solutions in population by search programs in last consecutive five attempts, the worst solution is removed from the population and a newly generated one is added to the pool of individuals. The process of search stops if no improvement is observed within the last 0.5 minute of computation.

Having a finite set of predefined population management strategies $Strategies = \{Strategy(1), Strategy(2), Strategy(3)\}$, all search procedures initially begin their search according to the rules defined in the first strategy ($str = 1$). If the stoping criterion defined in this strategy is reached, then str is increased by one. The general rule for changing strategy is that if stopping criterion is met and no improvement is observed, then str is increased by one, otherwise str is reset to 1 (default strategy).

The whole process of search stops where no improvement is observed for all strategies. A pseudocode including main steps of the proposed approach is presented in Algorithm 1.

Algorithm 1. A Cooperative Search Approach for VRP (CSA)

Require: $popSize$ - population size, N_{SP} - the number of search procedures, $SP = \{SP(1), SP(2), \ldots, SP(N_{SP})\}$ - a set of search procedures, f - fitness function, $strMax$ - the number of population management strategies, $Strategies = \{Strategy(1), Strategy(2), Strategy(strMax)\}$ - set of predefined population management strategies

Ensure: s_best - best solution found

1: Generate an initial population of solutions (individuals) $P = \{s_1, s_2, \ldots, s_{popSize}\}$, and store them in the common memory
2: $s\_best \leftarrow \arg\min_{s_i \in P} f(s_i)$
3: $str \leftarrow 1$
4: **while** $(str \leq strMax)$ **do**
5: $improved \leftarrow$ **false**
6: **while** (stopping criterion of strategy $Strategy(str)$ is not met) **do** {in parallel}
7: Select individual s_k $(k = 1, \ldots, popSize)$ from the common memory
8: Select a search procedure $SP(i)$ $(i = 1, \ldots, N_{SP})$
9: Execute $SP(i)$ on selected individual s_k improving it and return s_k^* as a resulting individual
10: **if** $f(s_k^*) < f(s_k)$ **then**
11: Store s_k^* in the common memory
12: **end if**
13: **if** $f(s_k^*) < f(s\_best)$ **then**
14: $s\_best \leftarrow s_k^*$
15: $improved \leftarrow$ **true**
16: **end if**
17: **end while**
18: **if** $(improved)$ **then**
19: $str \leftarrow 1$
20: **else**
21: $str \leftarrow str + 1$
22: **end if**
23: **end while**
24: **return** s_best

4 Computational Experiment

The main goal of the computational experiment, which has been carried out, was to evaluate to what extent presented approach produces results of good quality, measured as the mean relative error - MRE (in %) from the optimal (or the best known) solution, reported in [8]. Moreover, the influence of implementation of the adaptive mechanism on results, has been also investigated.

The experiment involved 14 instances of Christofides et al. [6], where the number of customers is 50-199. Each instance was repeatedly solved 10 times and the 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.

Results of the experiment are presented in Table 2. For each instance, besides its name and type, the next columns present: best known solution obtained from [8], minimal and maximal solution produced by the proposed approach (CSA) and the mean relative error calculated over all runs. The last line of the table includes MRE averaged over all instances.

Analysis of the results presented in the table allows one to conclude that the proposed approach can be seen as an interesting alternative to existing methods for solving VRP. General observation is that the mean relative error does not exceed 4%, but for majority of instances it reaches at most 1-2% deviation from the best known solution. For six instances the best known results have been reached. The worst results were observed for large instances. Type of constraints does not influence the results.

Table 2. Results obtained by the proposed approach with adaptive mechanism implemented in it

Instance	Customers	Type	Best known	CSA (min)	CSA (max)	MRE
vrpnc1	50	*C*	524.61	524.61	524.61	0.00%
vrpnc2	75	*C*	835.26	838.04	855.76	1.42%
vrpnc3	100	*C*	826.14	826.14	847.26	1.19%
vrpnc4	150	*C*	1028.42	1047.68	1072.84	3.00%
vrpnc5	199	*C*	1291.29	1325.95	1357.78	3.74%
vrpnc6	50	*C, R*	555.43	555.43	560.24	0.43%
vrpnc7	75	*C, R*	909.68	909.68	929.33	1.26%
vrpnc8	100	*C, R*	865.94	868.29	891.91	1.41%
vrpnc9	150	*C, R*	1162.55	1171.88	1202.45	1.85%
vrpnc10	199	*C, R*	1395.85	1418.74	1473.70	3.71%
vrpnc11	120	*C*	1042.11	1046.96	1059.33	1.03%
vrpnc12	100	*C*	819.56	819.56	821.92	0.05%
vrpnc13	120	*C, R*	1541.14	1543.50	1569.49	1.03%
vrpnc14	100	*C, R*	866.37	866.37	877.39	0.36%
Average						1.46%

Note: Type C denotes capacity constraints, R - route constraints

In order to discover the influence of implementation of adaptive mechanism on MRE, three additional runs have been performed. In each of them, only one population management strategy, respectively $Strategy(1)$ or $Strategy(2)$ or $Strategy(3)$, were used, respectively, while system was solving the problem. Figure 2 presents MREs for results obtained for these runs, and additionally, the fourth series refers to the case where adaptive mechanism was implemented.

It is easy to see that strategy where the population of solutions is dynamically managed brings better results in terms of MRE for all instances. Restriction to only one strategy results in the deterioration of the final solution. The biggest increase of MRE is observed for *Strategy*(2) and *Strategy*(3), the smallest increase - for *Strategy*(1).

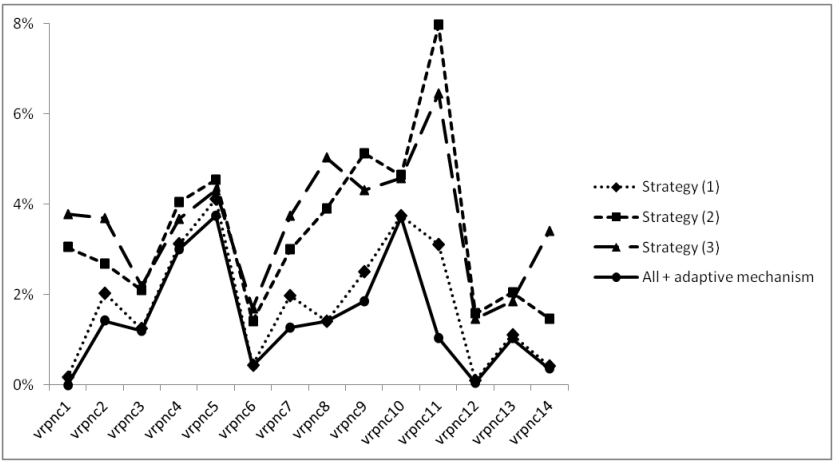


Fig. 2. Results obtained by the proposed approach without and with adaptive mechanism implemented in it

5 Conclusions

A new population-based multi-agent cooperative search approach for solving the Vehicle Routing Problem has been proposed in the paper. It uses a set of search procedures, which attempt to improve solutions stored in a common, central memory. An important part of the presented approach is an adaptive mechanism, where a few population management strategies are defined and which dynamically changes strategy during the search, according to their current performance. The experiment confirmed that implementation of such mechanism increase a performance of the whole approach in comparison with the case where only one population management strategy is used.

The future research will focus on generalization of the proposed approach in such a way that it could be used to solve other optimization problems implemented in multi-agent environment presented in [1]. Another interesting direction of research is investigation of possibility of implementation of other rules which would be used to dynamically switch between population management strategies, for example based on the level of population diversity. One of the measure of population diversity for VRP has been considered in [2].

Acknowledgments. The research has been supported by the Polish National Science Centre grant no. 2011/01/B/ST6/06986 (2011-2013). Calculations have been performed in the Academic Computer Centre TASK in Gdansk, Poland.

References

1. 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)
2. Barbucha, D.: Experimental Study of the Population Parameters Settings in Cooperative Multi-Agent System Solving Instances of the VRP. Submitted to LNCS Transactions on Computational Collective Intelligence (2012)
3. Blum, C., Roli, A.: Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys 35(3), 268–308 (2003)
4. Crainic, T.G., Toulouse, M.: Explicit and Emergent Cooperation Schemes for Search Algorithms. In: Maniezzo, V., Battiti, R., Watson, J.-P. (eds.) LION 2007 II. LNCS, vol. 5313, pp. 95–109. Springer, Heidelberg (2008)
5. Crainic, T.G., Toulouse, M.: Parallel Meta-heuristics. In: Gendreau, M., Potvin, J.-Y. (eds.) Handbook of Metaheuristics. International Series in Operations Research and Management Science, vol. 146, pp. 497–541. Springer, Heidelberg (2010)
6. Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.): Combinatorial optimization. John Wiley, Chichester (1979)
7. Golden, B.L., Raghavan, S., Wasil, E.A. (eds.): The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research Computer Science Interfaces Series, vol. 43. Springer (2008)
8. 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)
9. Le Bouthillier, A., Crainic, T.G.: A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. Computers & Operations Research 32, 1685–1708 (2005)
10. Lin, S.: Computer solutions of the traveling salesman problem. Bell Syst. Tech. J. 44, 2245–2269 (1965)
11. Masegosa, A.D., Pelta, D.A., Verdegay, J.L.: Cooperative Methods in Optimisation. Lambert Academic Publishing (2011)
12. Meignan, D., Creput, J.C., Koukam, A.: Coalition-based metaheuristic: a self-adaptive metaheuristic using reinforcement learning and mimetism. Journal of Heuristics 16(6), 859–879 (2010)
13. Osman, I.H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421–451 (1993)
14. Talbi, E.: A taxonomy of hybrid metaheuristics. Journal of Heuristics 8(5), 541–564 (2002)
15. Toulouse, M., Crainic, T.G., Gendreau, M.: Communication issues in designing cooperative multi thread parallel searches. In: Osman, I.H., Kelly, J.P. (eds.) Meta-Heuristics: Theory & Applications, pp. 501–522. Kluwer, Norwell (1996)

A-Team for Solving the Resource Availability Cost 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 the agent system based on A-Team and E-JABAT architecture for solving the resource availability cost problem (RACP) is proposed and experimentally tested. RACP known also as RIP (resource investment problem) belongs to the NP-hard problem class. To solve this problem an A-Team consisting of an asynchronous agents implemented using E-JABAT middleware have been proposed. Three kinds of optimization agent have been used. Computational experiment involves evaluation of the proposed approach.

Keywords: project scheduling, resource availability cost problem, RACP, resource investment problem, RIP, optimization, A-Team, agent, agent system.

1 Introduction

The paper proposes an agent based approach to solving instances of the resource availability cost problem (RACP) known also as resource investment problem (RIP). The considered problem have attracted less attention then other project scheduling problems, for example resource-constrained project scheduling problem (RCPSP). However, it is of great practical significance. It is used to model, for example, the bridge construction, staff management problems or negotiations the price of a project [17],[1]. In this problem the total costs of using a given amount of resource for the project is minimized. A solution of this problem consists of a set of activity starting times and a set of resource capacities, while respecting a project deadline. The problem is NP-hard.

RACP problem was introduced by Mohring (1984) [17] as the resource investment problem (RIP). He proposed an exact algorithm based on the known procedure for the RCPSP problem to solve it. Demeulemeester (1995) [9] proposed the next exact algorithm based on a branch-and-bound procedure for the RCPSP developed by himself and Herroelen (1992) [7], [8]. Rodrigues and Yamashita (2010) [22] modified the algorithm of Demeulemeester by reducing the search space using new bounds for branching scheme.

A few heuristic and metaheuristic algorithms are proposed to solve the RACP problem in the literature. Drexl and Kimms (2001) [10] develop two lower bounds for this problem using Lagrangean relaxation and column generation techniques,

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 443–452, 2012.

respectively. Both procedures are capable of yielding feasible solutions as well, so they also proposed two optimization guided heuristics. Yamashita et al. (2006) [26] proposed a multi-start heuristic based on the scatter search methodology using dynamic updating of the reference set, frequency-based memory within the diversification generator, and a combination method based on path relinking. Shadrokh and Kianfar (2007) [23] develop a genetic algorithm for the RACP in which the tardiness is permitted with penalty. Ranjbar et al. (2008) [21] developed two algorithms: a path relinking procedure and a genetic algorithm, in which a schedule is created with a precedence feasible priority list given to the schedule generation scheme. Van Peteghem and Vanhoucke (2011) [25] proposed an artificial immune system algorithm inspired by the vertebrate immune system and using new fitness function, the probability function for the composition of capacity lists, and the K-means diversity evaluation function for the preservation of diversity. Additionally the modification of the RACP problem has been proposed in several papers.

Approaches mentioned above to solve the RACP problem produce either approximate solutions or can be only applied for solving instances of the limited size. Hence, searching for more effective algorithms and solutions to the RACP/RIP 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 common feature of the contemporary multiple-agent systems.

The multiple-agent systems are an important and intensively expanding area of research and development. There exists a number of multiple-agent approaches proposed to solve different types of optimization problems. One of them is the concept of an asynchronous team (A-Team), originally introduced by [24]. The idea of A-Team was used to develop the JADE-based environment for solving a variety of computationally hard optimization problems called E-JABAT ([12],[2]). E-JABAT is a middleware supporting the construction of the dedicated A-Team architectures based on the population-based approach. The mobile agents used in E-JABAT allow for decentralization of computations and use of multiple hardware platforms in parallel, resulting eventually in more effective use of the available resources and reduction of the computation time.

In this paper the E-JABAT-based A-Team architecture for solving the RACP problem instances is proposed and experimentally validated. A-Team includes optimization agents which represent heuristic algorithms. The behavior of the A-Team is defined by the, so called, working strategy. In the proposed approach the architecture for the RACP problem implemented using the E-JABAT environment and three optimization algorithms based on local search, path relinking and Lagrangean relaxation, has been proposed.

The paper is constructed as follows: Section 2 of the paper contains the RACP problem formulation. Section 3 gives some information on E-JABAT environment. Section 4 provides details of the proposed A-Teams architecture designed for solving the RACP instances. Section 5 describes settings of the computational experiment carried-out with a view to validate the proposed approach.

Section 6 contains a discussion of the computational experiment results. Finally, Section 7 contains conclusions and suggestions for future research.

2 Problem Formulation

Single resource availability cost problem consists of a set of $n+2$ activities, where 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. The duration of an activity j, $j = 0, \ldots, n+1$ is denoted by d_j where $d_0 = d_{n+1} = 0$. There are r renewable resource types. The availability of each resource type k in each time period is unlimited but using each unit of each resource type costs. There are r cost values, one for each resource c_k, $k = 1, \ldots, r$. Each activity j requires r_{jk} units of resource k during each period of its duration, where $r_{1k} = r_{nk} = 0$, $k = 1, ..., r$.

There are precedence relations of the finish-start type 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 a 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 $0 \in SP_j$, $j = 1, \ldots, n+1$, and $n+1 \in SS_j$, $j = 0, \ldots, n$.

There is also a time limit impose for the project execution as deadline D. All parameters, except costs are non-negative integers.

The objective is to find a schedule S of activities starting times $[s_1, \ldots, s_n]$, where $s_1 = 0$ and $s_{n+1} \leq D$ and resource requirements $[r_1, \ldots, r_k]$, such that the total resource cost is minimized.

Formally, the RACP problem can be described as follows:

$$\min \sum_{k=1}^{r} c_k r_k \tag{1}$$

s.t.

$$s_i + d_j \leq s_j \quad \forall (i,j) \in SA \tag{2}$$

$$\sum_{i \in A_t} r_{ik} \leq r_k \quad \forall k = 1, \ldots r, \; t = 1, \ldots, D \tag{3}$$

where A_t denotes the set of activities processed in time t,

$$s_{n+1} \leq D \tag{4}$$

$$s_0 = 0 \tag{5}$$

$$r_k \geq 0 \quad \forall k = 1, \ldots r \tag{6}$$

The above formulated problem as a generalization of the classical job shop scheduling problem belongs to the class of NP-hard optimization problems [4],[17].

RACP can be denoted as $PS_m, \infty | prec | \sum C_k max \; r_k(S,t)$ [5] or $m, 1 | cpm, \delta_n | rac$, ($rac$ means resource availability costs) [11].

3 The E-JABAT Environment

E-JABAT is a middleware allowing to design and implement A-Team architectures for solving various combinatorial optimization problems, such as the resource-constrained project scheduling problem (RCPSP), the traveling salesman problem (TSP), the clustering problem (CP), the vehicle routing problem (VRP). It has been implemented using JADE framework. The problem-solving paradigm on which the proposed system is based can be best defined as the population-based approach.

E-JABAT produces solutions to combinatorial optimization problems using a set of optimization agents, each representing an improvement algorithm. Each improvement (optimization) algorithm when supplied with a potential solution to the problem at hand, tries to improve this solution. An initial population of solutions (individuals) is generated or constructed. Individuals forming an initial population are, at the following computation stages, improved by independently acting agents. Main functionality of the proposed environment includes organizing and conducting the process of search for the best solution.

To perform the above described cycle two main classes of agents are used. The first class called OptiAgent is a basic class for all optimization agents. The second class called SolutionManager is used to create agents or classes of agents responsible for maintenance and updating individuals in the common memory. All agents act in parallel. Each OptiAgent represents a single improvement algorithm (for example: local search, simulated annealing, tabu search, genetic algorithm etc.).

Other important classes in E-JABAT include: Task representing an instance or a set of instances of the problem and Solution representing the solution. To initialize the agents and maintain the system the TaskManager and PlatformManager classes are used. Objects of the above classes also act as agents.

E-JABAT environment has been designed and implemented using JADE (Java Agent Development Framework), which is a software framework supporting the implementation of multi-agent systems. More detailed information about E-JABAT environment and its implementations can be found in [12] and [2].

4 E-JABAT for Solving the RACP Problem

E-JABAT environment was successfully used by the authors for solving the RCPSP, MRCPSP and RCPSP/max problems ([13],[14],[3]). In the proposed approach the new data representation has been proposed dedicated for the RACP problem. Additionally some modification in order to improve the system efficiency has been implemented.

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:

– RACPTask inheriting from the Task class and representing the instance of the problem,

- RACPSolution inheriting from the Solution class and representing the solution of the problem instance,
- Activity representing the activity of the problem,
- Resource representing the renewable resource,
- TimeUnit representing the time unit in which the activities are processed.

The second set includes classes describing the optimization agents. Each of them includes the implementation of an optimization heuristic used to solve the RCPSP problem. All of them are inheriting from OptiAgent class. In the proposed dedicated A-Team this set includes the following classes:

- OptiLRA denoting the Lagrangean Relaxation Algorithm (LRA),
- OptiLSA denoting the Local Search Algorithm (LSA),
- OptiPRA denoting Path Relinking Algorithm (PRA),

The LRA is an implementation of the heuristic based on the Lagrangean relaxation method proposed by Drexl and Kimms in [10]. The relaxed problem of minimizing the total weighted completion times of the activities subject to precedence constraints is solved after convertion to minimum cut problem [18]. The implementation of the push relabel maximum flow algorithm described in [6] was used. The solution obtained represents a feasible suboptimal solution of the RACP problem.

Additionally, the above mentioned optimization agent and its algorithm based on the Lagrangean relaxation method is used to compute and update lower and upper bound for the processing instance. The bounds values are stored in RACPTask and used to stop computation in case when the lower bound or upper bound is reached by an agent.

The LSA is a local search algorithm which finds the shortest schedule for the considered problem with fixed resource availabilities by making a move. The move is understood as moving one of the activity to a new position in the schedule. All possible places in the schedule are checked in one iteration. For each combination of activities the value of possible solution is calculated. The best schedule is remembered and finally returned. The resource availabilities are calculated as follows:

- for feasible initial solution - the resource availabilities are decreased by x_k coefficient but not less then to the resource availability lower bound:

$$r_k = max(r_k - x_k(r_k - r_k^{LB}), r_k^{LB}), \text{ for } k = 1, \ldots, r, \tag{7}$$

- for infeasible initial solution - the resource availabilities are increased by y_k coefficient but not more then to the resource availability upper bounds:

$$r_k = min(r_k + y_k(r_k^{UB} - r_k), r_k^{UB}), \text{ for } k = 1, \ldots, r. \tag{8}$$

The resource availability lower r_k^{LB} and upper bound r_k^{UB} are calculated initially and updated during computation by the LRA algorithm. The coefficients x_k, for k=1,...,r. are set initially to 10% and updated during computation as well.

The PRA 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 in the case of LSA as moving one of the activities to a new position in the schedule. For each schedule in the path the value of the respective solution is checked using minimal for these two solutions resource availabilities. The best schedule is remembered and finally returned.

An individual is represented as a schedule of activities S. The final solution is obtained from the schedule for fixed resource availabilities by Serial Generation Scheme (serial SGS) procedure [16].

All optimization agents (OptiAgents) co-operate together using their A-Team common memory managed by the SolutionManager. The working strategy of SolutionManager has been defined as follows:

- All individuals in the initial population of solutions are generated randomly, improved by the LRA algorithm and stored in the common memory.
- Individuals for improvement are selected from the common memory randomly and blocked, which means that once selected individual (or individuals) cannot be selected again until all other individuals have been tried.
- Returning individual replaces the first found worse individual. If a worse individual cannot be found within a certain number of reviews (where 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 a randomly generated one.
- The computation time is defined by the no improvement time gap set by the user. If in this time gap no improvement of the current best solution has occurred, the A-Team stops computations.

5 Computational Experiment Settings

To evaluate the effectiveness of the proposed approach and compare the results the computational experiment has been carried out using benchmark instances generated by Yamashita at al. [26] for their computational experiment. The instances of RCPSP for 30, 60, 90 and 120 activities and 4 resource types are taken from the PSPLIB [19], and instances for RCPSP for 6 and 8 resource types has been generated by ProGen [15] using the following settings:

- Resource factor (RF): 0.25, 0.5, 0.75 and 1.0,
- Network complexity (NC): 1.5, 1.8 and 2.1.

Next, the instances has been adopted to RACP problem using Drexl and Kimms methodology [10] by removing the resource availability requirements, adding the costs drawn from a uniform distribution $U[1, 10]$ and adding the deadlines calculated using deadline factor $DF = 1.2$ ($D = DF \max_{i=0}^{n+1} s_i^{CP}$, where s_i^{CP} denotes the earliest starting times taken from the critical path).

The test set includes 144 problem instances. The experiment involved computation with the fixed number of optimization agents, fixed population size, and the limited time indicated by the no improvement time gap.

The proposed A-Team includes 3 optimization agents representing the LRA, LSA and PRA algorithms described in Section 4 - one of each type. The population has included 10 individuals, and the no improvement time gap has been set to 3 minutes. The values of the parameters are chosen on the basis of the previous experiments [12], [13], [14].

The experiment has been carried out using nodes of the cluster Holk of the Tricity Academic Computer Network built of 256 Intel Itanium 2 Dual Core 1.4 GHz with 12 MB L3 cache processors and with Mellanox InfiniBand interconnections with 10Gb/s bandwidth. During the computation one node per three optimization agents was used.

6 Computational Experiment Results

During the experiment the following characteristics of the computational results have been calculated and recorded: mean and maximal relative error (Mean RE) calculated as the deviation from the best solution obtained by Yamashita at al. [26] for three heuristics: scatter search with dynamic update (SSD) and two multi-start heuristic (FMS and RMS), the number of best results obtained, mean computation time required to find the best solution (Mean CT) and mean total computation time (Mean total CT). Each instance has been solved five times and the results have been averaged over these solutions.

Table 1. Performance of the proposed A-Team in terms of the mean relative error and number of the best results obtained

	#Activities				Mean	#Best results
	30	60	90	120		
A-Team for RACP	0.51%	1.23%	1.30%	1.52%	1.14%	96

Table 2. Performance of the proposed A-Team in terms of the mean computation time in seconds

	#Activities				Mean
	30	60	90	120	
A-Team for RACP	89.78s	123.42s	269.30s	315.92s	199.61s

Table 3. Performance of the proposed A-Team in terms of the mean total computation time in seconds

	#Activities				Mean
	30	60	90	120	
A-Team for RACP	141.04s	345.67s	470.53s	537.05s	373.57s

Table 4. Literature reported results [26]. Mean RE from the best known solution obtained by Yamashita at al. [26] and number of the best solutions for three heuristics: SSD, FMS and RMS.

	#Activities				Mean	#Best results
	30	60	90	120		
SSD	0.17%	0.00%	0.00%	0.00%	0.04%	137
FMS	0.42%	0.97%	1.33%	1.51%	1.06%	33
RMS	0.72%	1.77%	1.92%	2.26%	1.67%	31

Table 5. Literature reported results [26]. Mean computation time and mean total computation time in seconds for three heuristics: SSD, FMS and RMS.

	Mean CT	Mean total CT
SSD	1609.55s	3262.01s
FMS	945.09s	3135.13s
RMS	133.92s	3117.85s

Performance of the proposed A-Team is presented in Tables 1, 2 and 3. These results are compared with the results reported in the literature [26] shown in Tables 4 and 5.

The experiment results show that the proposed E-JABAT based A-Team for RACP implementation is effective and the results are comparable with the literature reported results. In each case the 100% of feasible solutions has been obtained. The times obtained in the experiment are quite good, however in the case of the agent based approaches it is difficult to directly compare computation times. The results obtained be a single agent may or may not influence the results obtained by the other agents. Additionally the computation time includes the time used by agents to prepare, send and receive messages.

7 Conclusions

Experiment results show that the proposed implementation based on the dedicated A-Team architecture is an effective and competitive tool for solving instances of the RACP problem. Presented results are comparable with solutions known from the literature. It can be also noted that they have been obtained in a comparable time. Time comparisons in this case might be misleading since the proposed A-Teams have been run using different numbers and kinds of processors. In case of the agent-based environments the significant part of the time is used for agent communication which has an influence on both - computation time and quality of the results.

The presented implementation and experiment is a first approach to construct A-Team for RACP problem. The experiment should be extended to examine the

A-Team behavior for different no improvement time gaps, different numbers of optimization agents and different population sizes. The other optimization algorithms and ideas to improve this implementation should be considered and tested.

Future research will concentrate on implementing more sophisticated procedures and optimization agents, as well as on searching for the best configuration of the heterogenous agents used during computations.

Acknowledgments. The authors are grateful to Proffesor Denise S. Yamashita and Proffesor Sávio B. Rodrigues from Federal University of São Carlos for making available the benchmark datasets and solutions for RACP problem.

The research has been supported by the Ministry of Science and Higher Education grant no. N N519 576438 for years 2010–2013. Calculations have been performed in the Academic Computer Centre TASK in Gdansk.

References

1. Artigues, C., Demassey, S., Néron, E.: Resource-Constrained Project Scheduling. Models, Algorithms, Extensions and Applications. ISTE Ltd. and John Wiley & Sons, Inc. (2008)
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, L.C. (eds.) Intel. Agents in the Evol. of Web & Appl. SCI, vol. 167, pp. 57–86. Springer, Heidelberg (2009)
3. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: Parallel Cooperating A-Teams. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS (LNAI), vol. 6923, pp. 322–331. Springer, Heidelberg (2011)
4. Błażewicz, J., Lenstra, J., Rinnooy, A.: Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics 5, 11–24 (1983)
5. Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E.: Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods. European Journal of Operational Research 112, 3–41 (1999)
6. Cherkassky, B.V., Goldberg, A.V.: On Implementing Push-Relabel Method for the Maximum Flow Problem. In: Balas, E., Clausen, J. (eds.) IPCO 1995. LNCS, vol. 920, pp. 157–171. Springer, Heidelberg (1995)
7. Demeulemeester, E.L.: Optimal Algorithms for Various Classes of Multiple Resource-Constrained Project Scheduling Problems, Ph.D. thesis, Department of Applied Economics, Katholieke Universiteit Leuven, Belgium (1992)
8. Demeulemeester, E.L., Herroelen, W.S.: A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem. Management Science 38, 1803–1818 (1992)
9. Demeulemeester, E.L.: Minimizing resource availability costs in time-limited project networks. Management Science 41, 1590–1598 (1995)
10. Drexl, A., Kimms, A.: Optimization guided lower and upper bounds for the resource investment problem. Journal of the Operational Research Society 52, 340–351 (2001)

11. Herroelen, W., De Reyck, B., Demeulemeester, E.L.: A classification scheme for project scheduling. In: Węglarz, J. (ed.) Handbook of Recent Advances in Project Scheduling, pp. 1–26. Kluwer, Dordrecht (1999)
12. 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, Part III. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
13. Jędrzejowicz, P., Ratajczak-Ropel, E.: New Generation A-Team for Solving the Resource Constrained Project Scheduling. In: Proc. the Eleventh International Workshop on Project Management and Scheduling, Istanbul, pp. 156–159 (2008)
14. 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)
15. Kolisch, R., Sprecher, A., Drexl, A.: Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science 41, 1693–1703 (1995)
16. Kolisch, R.: Serial and parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. European Journal of Operational Research 43, 23–40 (1996)
17. Möhring, R.: Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion Time. Operations Research 32, 89–120 (1984)
18. Möhring, R.H., Schulz, A.S., Stork, F., Uetz, M.: Solving project scheduling problems by minimum cut computations. Management Science 49, 330–350 (2003)
19. PSPLIB, http://129.187.106.231/psplib
20. Radermacher, F.J.: Scheduling of Project Networks. Annals of Operations Research 4, 227–252 (1985)
21. Ranjbar, M., Kianfar, F., Shadrokh, S.: Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. Appl. Math. Comput. 196, 879–888 (2008)
22. Rodrigues, S., Yamashita, D.: An exact algorithm for minimizing resource availability costs in project scheduling. European Journal of Operational Research 206, 562–568 (2010)
23. Shadrokh, S., Kianfar, F.: A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty. European Journal of Operational Research 181, 86–101 (2007)
24. 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)
25. Van Peteghem, V., Vanhoucke, M.: An artificial immune system algorithm for the resource availability cost problem. Flexible Services and Manufacturing Journal, 1936-6582, 1–23 (2011)
26. Yamashita, D., Armentano, V., Laguna, M.: Scatter search for project scheduling with resource availability cost. European Journal of Operational Research 169, 623–637 (2006)

Agent-Based Approach to RBF Network Training with Floating Centroids

Ireneusz Czarnowski and Piotr Jędrzejowicz

Department of Information Systems, Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
{irek,pj}@am.gdynia.pl

Abstract. In this paper the agent-based population learning algorithm designed to train RBF networks (RBFN's) is proposed. The algorithm is used to network initialization and estimation of its output weights. The approach is based on the assumption that a location of the radial based function centroids can be modified during the training process. It is shown that such a floating centroids may help to find the optimal neural network structure. In the proposed implementation of the agent-based population learning algorithm, RBFN initialization and RBFN training based on the floating centroids are carried-out by a team of agents, which execute various local search procedures and cooperate to find-out a solution to the considered RBFN training problem. Two variants of the approach are suggested in the paper. The approaches are implemented and experimentally evaluated.

Keywords: neural networks, radial basis function, RBF network, floating centroids, population learning algorithm, A-Team.

1 Introduction

The RBF networks, introduced by Bromhead and Lowe [5], can be considered as universal approximation tools similarly to multilayer perceptrons (MLPs). However Radial basis function networks (RBFNs) usually achieve faster convergence since only one layer of weights is required [12].

A RBF network is constructed from only one hidden layer, while MLP networks may have one or more hidden layers. RBFN uses different activation functions at each unit. The activation functions in the RBF units compute a distance between the input examples and the centers, while the activation functions of the MLP compute inner products from the input instances and weights. Each hidden unit in the RBFN represents a particular point in the space of the input instances. The output of the hidden unit depends on the distance between the processed instances and the particular point in the input space of instances. The distance is calculated as a value of the activation function. Next, the distance is transformed into a similarity measure that is produced through a nonlinear transformation carried-out by the output function. The output function of the hidden unit is called the radial basis function. Each particular point in the space of the input instances, is called an initial seed point, prototype, centroid or kernel of the basis function.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 453–462, 2012.

The performance of the RBF network depends on numerous factors including the number of the radial basis functions, their dispersion or shapes, the number of centroids and their locations, and other parameters describing the radial basis functions as well as weights and the method used for learning the input-output mapping. In general, training of the RBFN involves two stages:

- At the first stage the RBF parameters including cluster centroids together with their dispersion are estimated. It is a very important stage from the point of view of achieving a good approximation by the RBF network. The process of determining the number of centroids is associated with construction (initialization) of the RBFN.
- At the second stage the weights used to form the linear combination of the outputs of the hidden neurons are estimated.

Each of the above stages can be considered as an independent RBFN training problem. Different approaches to learn the RBF parameters and weights of RBFNs have been developed. Among approaches to determining parameters of the RBF proposed in the literature some of the best known include [15], [3], [10], [21], [16], [4] and [14]. Among classical methods used to RBFN initialization are cluster techniques, such as vector quantization or input-output clustering. Besides clustering methods, the support vector machine or the orthogonal forward selection approaches are used.

To estimate the output weights one can apply similar techniques as used for the MLP training, for example, the back propagation algorithm. Another approach includes point to estimation of the weights by linear least square methods [22]. However, for practical application these approaches are often very time consuming, an extensive research work is being carried-out in order to accelerate this process. Problem with the back propagation methods is a high likelihood of being caught in a local optimum. Hence, researchers look not only for algorithms that train RBF networks quickly but rather for quick algorithms that are not likely, or less likely, to get trapped in a local optimum [23].

None of the approaches proposed so far (see, for example, [13], [20]) can be considered as a superior, guaranteeing optimal results in terms of the learning error reduction or increased efficiency of the learning process. Hence, searching for robust and efficient approaches to RBFN construction and learning is still an important field of research.

Among methods dedicated to RBFN's training are hybrid approaches, where a training is carried out through simultaneous optimization of the location of centroids and widths, that is, in another words, through optimization of the RBFN structure [18], [3].

In a majority of cases it is also assumed that a RBF neural network should be trained based on a fixed centroids. However, fixing centroids decreases chances to find optimal neural network structure [20]. In [20] the Floating Centroid Method (FCM) is proposed to improve RBFN performance by introducing many floating centroids obtained during the optimization process by using the k-means algorithm. Based on the experimental evaluation it has been shown that the FCM improves performance (including generalisation accuracy, training accuracy, training speed) of neural network-based classifiers.

The paper deals with the RBF training. The main contribution of the paper is proposing and evaluating through computational experiment an agent-based population learning algorithm used to determine centroids (prototypes, kernels) of the Gaussian neurons prior to network training and to determine the output weights of the RBF network. Both search processes are carried out in parallel, which is consistent with the idea of the floating centroids. The extended variant of the approach, also introduced in the paper, allows independent optimization of the centroid location during the training process. In general, both new variants named, respectively, *ABRBFN 2* and *ABRBFN 3,* are an extension of the approach introduced in [8] and known as the *ABRBFN 1*. In the *ABRBFN 1* approach an agent-based population learning algorithm was used to locate prototypes within the produced clusters and the backpropagation algorithm for output weights estimation was applied.

The goal of the paper is to show through computational experiment that the proposed agent-based RBFN training with floating centroids can be competitive to its earlier version as presented in [8], as well as to other RBFN training algorithms. To validate the approache, an extensive computational experiment has been carried-out. Performance of the proposed algorithm has been evaluated using several benchmark datasets from the UCI repository [1].

The paper is organized as follows. Section 2 gives a basic account of the RBF networks. Idea of the agent-based population learning algorithm is presented in Section 3. Section 4 explains main features of the proposed implementation of the agent-based population learning algorithm. Section 5 provides details on the computational experiment setup and discusses its results. Finally, the last section contains conclusions and suggestions for future research.

2 RBF Neural Network

A RBF network is constructed from a tree-layer architecture with a feedback. The input layer consisting of a set of source units connects the network to the environment. The hidden layer consists of hidden neurons with radial basis functions [12]. One of the most popular output functions of the RBF hidden units is the Gaussian function [5], which has been chosen to best fit data from each cluster. In such a case the output function takes the following form:

$$G(r,b) = e^{-(\frac{r}{b})^2},$$

where r is a norm function denoted as $r=\|x-c\|$, where x is an input instance, c represents a centroid and b defines a function dispersion (or "width" of the radial function).

The output function of the RBF hidden unit most frequently is calculated using the Euclidean distance although other measures of distance can be also used. Thus, in general case, r refers to the Euclidean norm [11].

The output of the RBF network is a linear combination of the outputs of the hidden units, i.e. a linear combination of the nonlinear radial basis function generating approximation of the unknown function. In case of the classification problems the

output value is produced using the sigmoid function with a linear combination of the outputs of the hidden units as an argument. In general, the RBFN output function has the following form:

$$f(x, w, p) = \sum_{i=1}^{M} w_i G_i(r_i, p_i),$$

where M defines the number of hidden neurons, G_i is a radial basis function associated with i-th hidden neuron, p_i is a vector of parameters, which can include a location of centroids, dispersion or other parameters describing the radial function.

The RBF network initialization is a process, where the set of parameters of the radial basis functions needs to be calculated or drawn. On the other hand, RBFN training involves finding a set of weights of links between neurons such that the network generates a desired output signals. Thus, training process is also considered as adjusting values of these weights using a set of training patterns showing the desired RBFN behavior. In general, the RBFN training process aims at minimizing the learning error $E(W)$:

$$E(W) = \min_{W} \sum_{1}^{n} e(W, x_i, d_i)$$

where W is a set of the respective weights, $e(W,X,D) = (d_i\text{-f}(x_i,w,p))$ is an error function, where X is a set of input instances and D is a set of outputs respectively for n training patterns consisting of input-output pairs $\{(x_1,d_1),(x_2,d_2),...,(x_n,d_n): x_{1,...,n} \in U, d_{1,...,n} \in D\}$.

The RBFN training process can be viewed as solving the optimization task, where the optimization objective is to minimize the value of the target function by finding the optimal values of vector weights and vector of RBF parameters.

Since the RBF neural network training belongs to the class of computationally difficult combinatorial optimization problems [13], it is reasonable to apply to solve this task one of the known metaheuristics. In this paper an agent-based population learning algorithm, proposed originally in [2], is adopted for the purpose of the RBFN training.

3 Agent-Based Population Learning Algorithm

In [2] it has been shown that agent-based population learning search can be viewed as a robust and powerful optimizing technique. In the agent-based population learning implementation both - optimization and improvement procedures are executed by a set of agents cooperating and exchanging information within an asynchronous team of agents (A-Team). The A-Team concept was originally introduced in [19].

The concept of the A-Team was motivated by several approaches like blackboard systems and evolutionary algorithms, which have proven to be able to successfully solve some difficult combinatorial optimization problems. Within an A-Team agents achieve an implicit cooperation by sharing a population of solutions, to the problem to be solved.

An A-Team can be also defined as a set of agents and a set of memories, forming a network in which every agent remains in a closed loop. Each agent possesses some problem-solving skills and each memory contains a population of temporary solutions

to the problem at hand. It also means that such an architecture can deal with several searches conducted in parallel. In each iteration of the process of searching for the best solution agents cooperate to construct, find and improve solutions which are read from the shared, common memory. All agents can work asynchronously and in parallel.

Main functionality of the agent-based population learning approach includes organizing and conducting the process of search for the best solution. It involves a sequence of the following steps:

- Generation of the initial population of solutions to be stored in the common memory.
- Activation of optimizing agents which execute some solution improvement algorithms applied to solutions drawn from the common memory and, subsequently, store them back after the attempted improvement in accordance with a user defined replacement strategy.
- Continuation of the reading-improving-replacing cycle until a stopping criterion is met. Such a criterion can be defined either or both as a predefined number of iterations or a limiting time period during which optimizing agents do not manage to improve the current best solution. After computation has been stopped the best solution achieved so far is accepted as the final one.

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 A-Team implementations are described.

4 An Approach to the RBF Network Training

The paper deals with the problem of RBFN training by an implementation of the agent-based population learning algorithm. The main goal is to produce the optimal RBF network structure with respect to:

- Producing clusters and determining their centroids.
- Finding the output weights of the RBFN.

Under the proposed approach clusters are produced at the first stage of the training process. They are generated using the procedure based on the similarity coefficient calculated as in [7]. Thus clusters contain instances with identical similarity coefficient and the number of clusters is determined by the value of the similarity coefficient (for details see, for example, [7]). In the proposed approach a modified similarity-based clustering procedure suitable for classification problems and function approximation (regression) problems as introduced in [8] has been used.

The second stage involves selection of centroids from thus obtained clusters. An agent-based algorithm with a dedicated set of agents is used to locate centroids within clusters. Thus, an A-Team consists of agents which execute the improvement procedure and cooperate with a view to solve RBF network initialization problem. The second stage also involves estimation of values of output weights of the RBFN. This searching process is related to a different set of agents which execute the improvement procedure with a view to solve a non-linear numerical optimization problem. Both of the above searching processes are carried out in parallel. Such

approach is consistent with the idea of floating centroids where the centroid locations are changed during the process of estimating the output weights of the RBFN.

Most important assumptions behind the *ABRBFN 2* approach, can be summarized as follows:

- Shared memory of the A-Team is used to store a population of solutions to the RBFN training problem.
- A solution is represented by a string consisting of two parts. The first contains integers representing numbers of instances selected as centroids and the second – real numbers, representing weights of connections between neurons of the network under training.
- The initial population is generated randomly.
- Initially, potential solutions are generated through randomly selecting exactly one single centroid from each of the considered clusters.
- Initially, the real numbers representing weights are generated randomly.
- Each solution from the population is evaluated and the value of its fitness is calculated. The evaluation is carried out by estimating classification accuracy or error approximation of the RBFN, which is initialized using prototypes and set of weights indicated by the solution.

To solve the RBFN training problem two groups of optimizing agents have been considered. The first group includes agents executing procedures for centroid selection. To this end the following procedures have been implemented:

- Local search with the tabu list for prototype selection – this procedure 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.
- Simple local search – this procedure 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 of optimizing agents includes procedures for estimation of the output weights. It has been decided to adapt and implement the following procedure, introduced before in [6]:

- Gradient mutation – this procedure modifies two randomly selected elements within a solution by incrementing or decrementing their values. Direction of change (increment/decrement) is random and has identical probability equal to 0.5. The value of change is proportional to the gradient of an individual. If the fitness function value of an individual has improved then the change is accepted.
- Gradient adjustment procedure - the gradient adjustment agent adjusts the value of each element of the solution by a constant value Δ proportional to its gradient. Delta is calculated as $\Delta = \alpha \cdot \xi$, where α is the factor determining a size of the step in direction of ξ, known as a momentum and α takes values from (0, 1]. In the

proposed algorithm its value iterates starting from 1 with the step equal to 0.02. ξ is a vector determining a direction of search and is equal to the gradient of a solution.

The above procedure was extended to *ABRBFN* 3 in the following manner. During the training process the value of the error function is monitored by each optimizing agent. When this value does not decrease after a predefined number of iterations the improvement procedure in a weight dimension is suspended and only a search for a better location of centroids within clusters is carried-out by agents responsible for the centroid selection. The search in a centroid dimension is carried-out by the predefined number of iteration. Subsequently, the weight searching process is resumed.

The above extension resulting in a periodic modifications of solutions only in the centroid dimension has been proposed to enhance the benefits of floating centroids method and to increase chances of finding the optimal RBF network.

5 Computational Experiment

5.1 Computational Experiment Setting

To validate the proposed approach it has been decided to carry out computational experiment. The experiment aimed at answering the following two questions:

- Does the proposed agent-based RBFN training approach (*ABRBFN 2*) perform better than classical methods of RBFN training?
- Does the extended version called *ABRBFN 3* perform better than *ABRBFN 2*?

The aim of the experiment has been also to compare the proposed approaches with the *ABRBFN 1* introduced in [8], where an agent-based learning algorithm has been used only to perform search for a location of centroids.

The evaluation of the proposed approaches and comparison of performance of the *ABRBFN 2* and *ABRBFN 3* with other algorithms are based on two problem kinds i.e. the classification and the regression problems. For both problems the proposed algorithms have been applied to solve respective problems using several benchmark datasets obtained from the UCI Machine Learning Repository [1]. Basic characteristics of these datasets are shown in Table 1.

Table 1. Datasets used in the reported experiment

Dataset	Type of problem	Number of instances	Number of attributes	Number of classes	Best reported results
Forest Fires	Regression	517	12	-	-
Housing	Regression	506	14	-	-
WBC	Classification	699	9	2	97.5% [1] (*Acc.*)
Credit	Classification	690	15	2	86.9% [1] (*Acc.*)
Sonar	Classification	208	60	2	97.1% [1] (*Acc.*)

Each benchmark problem has been solved 50 times, and the experiment plan involved 5 repetitions of the 10-cross-validation scheme. The reported values of the quality measure have been averaged over all runs. The quality measure in case of classification problems was the correct classification ratio – accuracy (*Acc*). The overall performance for regression problems has been computed by the mean squared error (*MSE*) calculated as the approximation error over the test set.

During the experiment population size for each investigated A-Team architecture was set to 60. The process of searching for the best solution in the centroid dimension has been stopped either after 100 iterations or after there has been no improvement of the current best solution for one minute of computation. The number of epochs for procedures designed to determine the output weights has been set to 1000. In the *ABRBFN 3* after 500 epochs without an improvement the search process for the centroid dimension only is initiated with the maximum number of iterations set to 100. Values of the above parameters have been set arbitrarily in the trials and errors procedure. The dispersion of the Radial function has been calculated as a double value of minimum distance between basis functions [14].

The proposed A-Team has been implemented using the middleware environment called JABAT [2].

5.2 Experiment Results

Table 2 shows mean values of the classification accuracy of the classifiers (Acc) and the mean squared error of the function approximation models (MSE) obtained using the RBFN architecture.

Table 2. Results obtained for different variants of the proposed algorithms to the RBNF's training and their comparison with the performance of some different competitive approaches

Problem / Algorithm	Forest fires	Housing	WBC	Credit	Sonar
	MSE			Acc. (%)	
ABRBFN 1 [8]	2.15	35.24	94.56	84.56	82.09
ABRBFN 2	**2.01**	**34.71**	**97.34**	84.96	**84.11**
ABRBFN 3	2.05	35.6	94.9	**87.14**	81.72
Neural network – MLP	2.11 [24]	40.62 [24]	96.7 [9]	84.6 [9]	**84.5** [9]
Multiple linear regression	2.38 [24]	36.26 [24]	-	-	-
SVR/SVM	**1.97** [24]	44.91 [24]	96.9 [9]	84.8 [9]	76.9 [9]
C 4.5	-	-	94.7 [9]	85.5 [9]	76.9 [9]

From Table 2, one can observe that in a strong majority of cases at least one of the *ABRBFN* variants assures competitive results in comparison to other approaches and the finding holds true independently from the type of the problem. The *ABRBFN 2* algorithm and its extended version assure better results than the *ABRBFN 1* algorithm

considering all benchmark problems. In one cases it can be also observed, that the *ABRBFN 2* algorithm does not assure as good results as *ABRBFN 3*. The experiment results also show that the agent-based search is a suitable tool for solving complex optimization problems including the RBFN initialization and estimation of the output weights of the RBFN. The agent-based algorithm is also a suitable tool allowing for discarding the fixed-centroid constraint, which finally can contribute to improvement of the quality of training.

6 Conclusions

In this paper an agent-based population learning algorithm for RBF neural network training with floating centroids is proposed. The role of the proposed agent-based population learning algorithm is to select the appropriate centroids and to estimate the output weights of the RBFN. Both processes are carried out in parallel.

Two variants of the algorithm has been suggested and discussed. The second variant bases on additional modifications of the location of centroids within clusters. In the reported computational experiment the proposed algorithms outperformed other techniques for RBF initialization.

In the future it is planned to carry-out more refined statistical analysis of the results to obtained a better insight into properties of the proposed approach. Future research will also focus on finding more effective configurations of the RBF networks. The additional experiment are planned using additional benchmark datasets.

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.) Intel. Agents in the Evol. of Web & Appl. SCI, vol. 167, pp. 57–86. Springer, Heidelberg (2009)
3. Billings, S.A., Zheng, G.L.: Radial basis function networks configuration using genetic algorithms. Neural Networks 8(6), 877–890 (1995)
4. Bishop, C.: Neural Network for Pattern Recognition. Oxford University Press (1995)
5. Broomhead, D.S., Lowe, D.: Multivariable Functional Interpolation and Adaptive Networks. Complex Systems 2, 321–355 (1988)
6. Czarnowski, I., Jędrzejowicz, P.: An agent-based approach to ANN training. Knowledge-Based Systems 19, 304–308 (2006)
7. Czarnowski, I.: Cluster-based Instance Selection for Machine Classification. Knowledge and Information Systems 30(1), 113–133 (2012)

8. Czarnowski, I., Jędrzejowicz, P.: An Approach to Cluster Initialization for RBF Networks. In: Proceedings of the 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, San Sebastian, Spain (to appear, 2012)
9. Datasets used for classification: comparison of results. Directory of data sets, http://www.is.umk.pl/projects/datasets.html (accessed September 1, 2009)
10. Esposito, A., Marinaro, M., Oricchio, D., Scarpetta, S.: Approximation of continuous and discontinuous mappings by a growing neural RBF-based algorithm. Neural Networks 13, 651–665 (2000)
11. Fasshauer, G.E., Zhang, J.G.: On Choosing "optimal" Shape Parameters for RBF Approximation. Numerical Algorithms 45(1-4), 345–368 (2007)
12. Gao, H., Feng, B., Hou, Y., Zhu, L.: Training RBF Neural Network with Hybrid Particle Swarm Optimization. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 577–583. Springer, Heidelberg (2006)
13. 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)
14. Huang, G.-B., Saratchandra, P., Sundararajan, N.: A Generalized Growing and Pruning RBF(GGAP-RBF) Neural Network for Function Approximation. IEEE Transactions on Neural Networks 16(1), 57–67 (2005)
15. Krishnaiah, P.R., Kanal, L.N.: Handbook of Statistics 2: Classification, Pattern Recognition and Reduction of Dimensionality. North Holland, Amsterdam (1982)
16. Lloyd, S.P.: Least Squares Quantization in PCM. IEEE Trans. Inf. Theory 28, 129–137 (1982)
17. Qasem, S.N., Shamsuddin, S.M.H.: Radial Basis Function Network Based on Multi-Objective Particle Swarm Optimization. In: Proceeding ofthe 6th International Symposium on Mechatronics and its Applications (ISMA 2009), Sharjah, UAE, March 24-26 (2009)
18. Ros, F., Pintore, M., Chretien, J.: Automatic design of growing radial basis function neural networks based on neighbourhood concepts. Chemometrics and Intelligence Laboratory Systems 87, 231–240 (2007)
19. 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)
20. Wang, L., Yang, B., Chen, Y., Abraham, A., Sun, H., Chen, Z., Wang, H.: Improvement of Neural Network Classifier Using Floating Centroids. Knowledge Information Systems 31, 433–454 (2012)
21. Wang, Z., Zhu, T.: An Efficient Learning Algorithm for Improving Generalization Performance of Radial Basis Function Neural Networks. Neural Networks 13, 545–553 (2000)
22. Wei, L.Y., Sundararajan, N., Saratchandran, P.: Performance Evaluation of a Sequential Minimal Radial Basis Function (RBF) Neural Network Learning Algorithm. IEEE Trans. Neural Networks 9, 308–318 (1998)
23. Shang, Y., Wah, B.W.: A global optimization method for neural network training. In: Conference of Neural Networks, vol. 29, pp. 45–54. IEEE Computer (1996)
24. Zhang, D., Tian, Y., Zhang, P.: Kernel-based Nonparametric Regression Method. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 410–413 (2008)

New Differential Evolution Selective Mutation Operator for the Nash Equilibria Problem

Urszula Boryczka and Przemysław Juszczuk

Institute of Computer Science, University of Silesia,
ul.Bedzinska 39, Sosnowiec, Poland
{urszula.boryczka,przemyslaw.juszczuk}@us.edu.pl

Abstract. Differential Evolution (DE) is a simple and powerful optimization method, which is mainly applied to numerical optimization. In this article we present a new selective mutation operator for the Differential Evolution. We adapt the Differential Evolution algorithm to the problem of finding the approximate Nash equilibrium in n person games in the strategic form. Finding the Nash equilibrium 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 the Nash approximation and called ϵ-Nash equilibrium. The fitness function in this approach is based on the max function which selects the maximal value from the set of payoffs. Every element of this set is calculated on the basis of the corresponding genotype part. We propose an approach, which allows us to modify only the worst part of the genotype. Mainly, it allows to decrease computation time and slightly improve the results.

Keywords: Differential Evolution, mutation, ϵ-Nash equilibrium.

1 Introduction

In the last decades many deterministic and stochastic methods of an optimization problem have been developed. However, no universal technique which could give the good results for all optimization problems has been found yet. One of these problems is numerical optimization which is used to test many new methods. Its aim is to find a solution which minimize the quality function $F(x)$:

$$x^* = \arg \min_{x \in X} F(x)$$

where X is a set of feasible solutions and $x \in X$ is a vector $x = [x^1, x^2, \ldots, x^{n_F}]$, n_F is a number of dimensions. Recently we have shown a way to transform the problem of finding Nash Equilibrium into the function optimization problem. Finding the Nash equilibrium in the n-person non-zero sum games is the 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 the equilibrium computation. Determining whether Nash equilibria exist, and effectively computing them, are relevant problems that have attracted much research

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 463–472, 2012.

in computer science [4,13]. Finding the simple, pure Nash equilibrium (in contrast to the mixed Nash equilibrium) is easy problem. 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).

In this article we present a new way to construct the mutation operator in the Differential Evolution algorithm. The fitness function in the problem of finding approximate Nash equilibrium is based on the *max* function. We show, that modifying only a part of the genotype may lead to the overall improvement of the fitness value. Moreover, this approach alows to reduce the algorithm complexity. In every iteration only the part of the genotype is modified at the same time.

Our article is organized as follows: first, we present some basic considerations about Nash equilibria in n-person games. Next we give details about other approaches in constructing mutation schemas. In the next section we give the detailed problem solution, and we present the new mutation schema. Moreover, we describe the Differential Evolution algorithm adapted to the problem of finding approximate Nash equilibrium in n-person game. Finally we show some experiments and results. We summarize with short conclusions.

2 Related Works

Differential Evolution (DE) is a stochastic, population-based search strategy developed by R. Storn and K. Price in 1995, and deeply studied by J. Lampinen, I. Zelinka [7,9,17], and others. The Differential Evolution algorithm was successfully used in many practical applications: neural network train [11], filter design [19] or image analysis [6]. High effectiveness of Differential Evolution leads to different modifications directed into specific problems. The ability of Differential Evolution (DE) to perform well in continuous-valued search spaces is well documented. The algorithm was also successfully used in the binary and discrete problems [1,5]. G. Pampara in [15] proposed a DE algorithm to evolve solutions to binary-valued optimization problems, without having to change the operation of the original DE. On the other hand there are numerous approaches relevant to the hybridization.

The most important part of allowing to improve performance and accelerate the convergence of the algorithm is the mutation operator. We show, that for specific problems like finding Nash equilibria, it is possible, to increase the speed of the algorithm. 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 [10], which is a pivoting algorithm similar to the simplex algorithm for linear programming. It is very often described as the state of the art algorithm, but in [18] it was shown that for some game classes it has exponential time of finding solution. In 1991 an algorithm based on the support enumeration was introduced [3], similar, but more effective algorithm was described in [16]. Both algorithms favor games, in which support of both players is very small. The algorithm proceeds by constructing a triangulated grid

over the space of mixed strategy profiles, and uses a path-following method to compute an approximate fixed point. This approximate fixed point can then be used as a starting point on a refinement of the grid. The algorithm begins with any mixed strategy profile consisting of rational numbers as probabilities. In our experiments we used implementation available in the Gambit software [12]. Without any options, the algorithm begins with the centroid, and computes one Nash equilibrium.

3 The Classical Differential Evolution

As in any other evolutionary algorithms, before the population can be initialized, both upper and lower bounds for each gene of the genotype must be specified. After that, the selection process takes place. During the selection stage, three parents are chosen and they generate a single offspring which competes with a parent to determine who passes to the following generation. Despite some similarities, the DE algorithm differs from evolutionary algorithms in that: mutation is a primary operator, and crossover is an auxiliary operator. The second difference: mutation is applied first to generate a trial vector, next this vector is used in crossover procedure.

The pseudocode of the general DE algorithm is presented below:

Algorithm 1. Basic DE algorithm

1 Create the initial population of genotypes $S^0 = \{\boldsymbol{X}_1^0, \boldsymbol{X}_2^0, ..., \boldsymbol{X}_n^0\}$;
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 $\{f(\boldsymbol{X}_1), f(\boldsymbol{X}_2), ..., f(\boldsymbol{X}_n)\}$;
5 Create the population of trial genotypes V^g based on S^g;
6 Make crossover of genotypes from the population S^g and V^g to create population U^g;
7 Choose the genotypes with the highest fitness function from population U^g and S^g for the next population;
8 $generation = generation + 1$, go to step 4;

the positions of individuals provide valuable information about the fitness landscape. Provided that a good uniform random initialization method is used to construct the initial population. The length of the single genotype depends on the dimension of the problem. In every iteration of the algorithm the following operations are performed: mutation, crossover, selection and fitness computation. There are various strategies, how to create the trial vector. The most popular strategy denoted by abbreviation DE/rand/1/ generates the individual by adding the weighted difference of two points:

$$\forall_i, \forall_j \; U_{i,j} = S_{r_1,j} + F \cdot (S_{r_2,j} - S_{r_3,j}).$$

Individual $\boldsymbol{S_i}$ is denoted as a targer vector. $(\boldsymbol{S_{r_2}} - \boldsymbol{S_{r_3}})$ is the differential vector created from the two random individuals $\boldsymbol{S_{r_2}}$ oraz $\boldsymbol{S_{r_3}}$. Moreover F is mutation parameter. Population U is next used in the crossover process. After the crossover, the child population V is created. Part of the genotype of the individual $\boldsymbol{V_i}$ is taken from the parent vector $\boldsymbol{S_i}$, while the remaining elements are taken from the trial vector $\boldsymbol{U_i}$. This process can be expressed by the formula:

$$\forall_i, \forall_j V_{i,j} = \begin{cases} U_{i,j} \text{ if } rand_j[0,1) < CR, \\ S_{i,j} \text{ other case.} \end{cases}$$

where CR is the crossover parameter. After the creation of the child population V, fitness function value for the all population is calculated. Fitness values for the child individuals are compared with the fitness values for the parent individuals (population S). To the next population $t+1$ is taken an individual with the higher fitness value (from the pair parent-child). The genotype with the lower fitness function value is transferred to the next generation (fitness function equal to 0 is identified as global optimum). Selection in the Differential Evolution is described by the given formula:

$$\forall_i, \boldsymbol{S}_i^{g+1} = \begin{cases} \boldsymbol{U}_i^g \text{ if } f(\boldsymbol{U}_i^g) \leq f(\boldsymbol{S}_i^g), \\ \boldsymbol{X}_i^g \text{ other case.} \end{cases}$$

where $\boldsymbol{S}_i^{g+1}$ is i-th individual from the generation $g+1$, and $f()$ is the fitness function.

4 A New Approach for the Mutation Schema in the Problem of Finding Nash Equilibrium

A non-cooperative game in a strategic form consists of a set of players, and, for each player, a set of strategies available to him as well as the payoff function mapping each strategy profile (i.e. each combination of strategies, one for each player) to a real number that captures the preferences of the player over the possible outcomes of the game:

$$\Gamma = \langle N, \{A_i\}, M \rangle, i = 1, 2, ..., n$$

where:

- $N = \{1, 2, ..., n\}$ is the players set;
- $\{A_i\}$ is the finite set of strategies for the i-th player with m-strategies;
- $M = \{\mu_1, \mu_2, ..., \mu_n\}$ is the finite set of the payoff functions.

Next we define the strategy profile $a = (\boldsymbol{a_1}, ..., \boldsymbol{a_n})$ for all players. Moreover:

$$a_{-i} = (\boldsymbol{a_1}, ..., \boldsymbol{a_{i-1}}, \boldsymbol{a_{i+1}}, ..., \boldsymbol{a_n}),$$

will be the strategy profile excluding i-th player. Mixed strategy for the i-th player will be denoted as:

$$\boldsymbol{a_i} = (P(a_{i_1}), P(a_{i_2}), ..., P(a_{i_m})),$$

where $P(a_{i_1})$ will be probability of chosing strategy 1 by the player i.

Nash equilibrium is a strategy profile such that no deviating player could achieve the payoff higher than the one that the specific profile gives him:

$$\forall_i, \forall_j \ \mu_i(a) \geq \mu_i(a_{i_j}, a_{-i}),$$

where i is the i-th player, j is the number of the strategies for given player, $\mu_i(a)$ is the payoff for the i-th player for the strategy profile a and $\mu_i(a_{i_j}, a_{-i})$ is the payoff for the i-th player using strategy j agaist profile a_{-i}.

Game in the strategic form for the n-playes may be transformed into the genotype. This process may be seen on the fig. 1. The whole individual consists of n parts, where every part is probability distribution over the set of strategies for the one player.

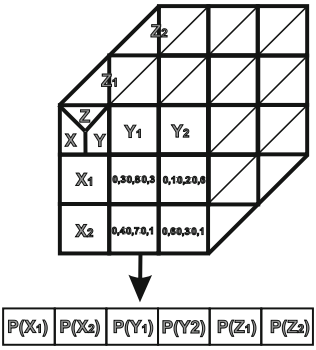


Fig. 1. Creation of the single individual

This transformation allows us to use specific approach, where the fitness function for the algorithm consists of two parts, where first part is given as follows:

$$f_1 = \max\{\max\{u_1(a_{11}, a_{-1}), ..., u_1(a_{1j}, a_{-1})\} - u_1(a), ..., \max\{u_i(a_{i1}, a_{-i}), ..., u_i(a_{ij}, a_{-i})\} - u_i(a)\}$$

where $u_i(a_{ij}, a_{-i})$ is the payoff for the i-th player using strategy j. Second necessary condition is to check, if sum of probabilities for every player is equal to one. The second condition may be described as follows:

$$f_2 = \sum_{i=1}^{n} |1 - \sum_{j=1}^{m} P(a_{ij})|$$

where f_1 means maximal deviation from the optimal strategy - denoted as the worst solution from all players. This value is also denoted as the ϵ value (ϵ Nash equilibrium means the approximate Nash equilibrium). Sum of this two above functions gives the fitness function, which is represented by the formula:

$$f = f_1 + c \cdot f_2,$$

where c is the constant value. Only after the fulfillment of the condition f_2, the f_1 may be minimized. Interesting is a fact, that the f_1 function is the worst case value for all players, so if we would be able to improve the payoff for the worst player, we also improve overall value of the fitness function. Moreover, the payoff value for the single player is calculated on the basis of the part of the genotype, so it is possible to only modify part of the genotype in the iteration. It is worth noting, that payoffs of all players are in some way related to each other, so changing part of the genotype of the selected player changes also payoffs of the remaining players (by the adding value e to the payoff value from the previous iteration). This whole process may be presented in the fig. 2. Despite this, as we show in the experiments, this simple modification slightly improves results.

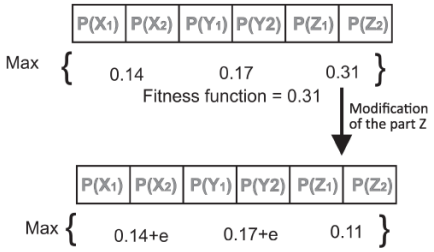


Fig. 2. Selective mutation operator

5 Experiments

The aim of this research work is to determine if the new proposed DE mutation schema is capable to solve problem of finding the approximate Nash equilibrium in the n-person games. We also show, that even for the large games involving more than 10 strategies per player the DE is capable to find optimal solutions faster than classical algorithms like the Simplicial Subidvision. All given parameters were chosen experimentally. Unfortunately because of space limitation we are not able to present the complete parameter tuning. In our experiments we used the following values:

- the population size $N = 50$;
- the crossover parameter $CR = 0.5$;
- the mutation parameter $F = 0.6$;
- for every game program was run 30 times;
- maximum time for generating solution was 300 seconds for both algorithms;
- for the third experiment we selected randomly 10 games with the 5,7, 10, 12, 15, 17 and 20 strategies.

The number of iterations for all experiments were set as follows:

- 2000 iterations for games with 5, 7 and 10 strategies per player;
- 2500 iterations for games with 12, 15 and 17 strategies per player;
- 3000 iterations for games with 20 strategies per player.

We used different number of iterations for games, because number of strategies per players inflicts the genotype size. Moreover, in the two tables, and on the first figures we used the following descriptions: x y str, means the x game with the y strategies per players. All tested games were games for three players with the same number of strategies. We used the famous GAMUT program for generating all test games [14]. In our experiments we used the Simplicial Subdivision implementation from the GAMBIT program.

In the table 1 we can see comparision of the computation time for the DE and the Simplicial Subdivision algorithm (denoted as SimpDiv). It is worth noting, that the Simplicial Subdivision algorithm gives repeatable results for every game - average, minimum and maximum value for every run of the algorithm are the same. In table 1 we put only one time computation value. The biggest advantage for the proposed solution is its ability to solve every game in the approximate same time. The Simplicial Subdivision algorithm (the last column in the table) wasn't able to solve some games, where maximum computation time is equal to 300 seconds. 300 seconds was the highest acceptable computation time, and it is clear to see, that for larger games this algorithm is much slower than the DE. Please note, that the Simplicial Subdivision algorithm finds only the exact Nash equilibrium, but as we can see, it falls to solve larger problems. It this case is better to use the Differential Evolution which gives approximate solution.

In the second table 2 we can see ϵ values for the same games. The Differential Evolution was able to solve every game, but in the other hand, none exact solution was found. The Nash equilibrium should be described as the global optimum, and ϵ equilibrium as the local optimum. We can suspect, that the local and global optima are not near of each other, and the DE falls in the local optimum. Of course given solutions are very good. It is worth noting, that some games seem to be more difficult than other (even with the same number of strategies). For example games with 20 strategies per player $120str$ and $220str$: in the first case, algorithm was able to find solution with ϵ value equal to 0.02, and in the second case we have 0.24. This dispersion should be examined in the future.

On the fig. 3 we can see the worst values from the table 2. This values are compared with the worst case values for the best algorithm for 2-players. Approximate algorithms for two players are mainly based on the transformation of the non-zero sum game into zero sum game (the difference of the two matrices). Those algorithms falls to solve games for more than two players, but it is interesting, that the proposed DE algorithm is able to achieve the better results.

Finally, on the last fig. 4 we can see randomly generated games with the different number of strategies. For every game with the same number of strategies we show number of games which were solved by the two algorithms: the Differential Evolution and the Simplicial Subdivision. For games with the 20 stratiegies, for

Table 1. Average, minimum, maximum, median and standard deviation for the time of computation(in seconds): comparision of the DE and the Simplicial Subdivision for different tested problems

Game	min	max	avg	std dev	median	SimpDiv
1 5 str	1.4	1.6	1.5	0.1	1.5	1.1
2 5 str	1.4	1.6	1.5	0.1	1.4	0.8
3 5 str	1.4	1.5	1.4	0.1	1.4	1.1
1 7 str	3.8	4.3	4.0	0.2	4.1	15.3
2 7 str	4.1	4.5	4.3	0.1	4.4	2.6
3 7 str	3.8	4.1	3.9	0.1	3.9	26.9
1 10 str	10.2	11.1	10.6	0.3	10.7	19.7
2 10 str	10.7	11.6	11.1	0.3	11.2	2.4
3 10 str	11.7	12.5	12.0	0.3	11.9	79.6
1 12 str	18.9	19.8	19.3	0.3	19.3	43.5
2 12 str	18.2	19.4	18.8	0.4	18.9	51.5
3 12 str	18.7	19.4	19.1	0.3	19.1	4.8
1 15 str	45.2	46.3	45.7	0.4	45.7	300
2 15 str	45.6	46.5	46.0	0.4	46.2	300
3 15 str	45.3	46.4	46.2	0.7	46.4	193.4
1 17 str	67.4	68.8	68.4	0.8	68.8	300
2 17 str	68.4	69.5	69.4	0.7	69.5	300
3 17 str	66.5	69.2	69.1	1.7	69.2	300
1 20 str	146.6	149.4	150.3	3.2	149.4	300
2 20 str	149.7	153.4	153.5	2.6	153.4	300
3 20 str	150.4	154.6	154.3	2.6	157.5	300

Table 2. Average, minimum, maximum, median and standard deviation ϵ values for different tested problems

Game	min	max	avg	std dev	median
1 5 str	0.18	0.21	0.19	0.01	0.19
2 5 str	0.15	0.17	0.16	0.00	0.16
3 5 str	0.19	0.24	0.22	0.02	0.22
1 7 str	0.19	0.21	0.19	0.01	0.19
2 7 str	0.11	0.13	0.12	0.00	0.12
3 7 str	0.09	0.12	0.10	0.01	0.10
1 10 str	0.10	0.13	0.11	0.01	0.11
2 10 str	0.13	0.16	0.14	0.01	0.14
3 10 str	0.18	0.26	0.22	0.03	0.21
1 12 str	0.17	0.19	0.18	0.01	0.18
2 12 str	0.15	0.18	0.16	0.01	0.16
3 12 str	0.12	0.14	0.13	0.00	0.13
1 15 str	0.14	0.17	0.15	0.01	0.15
2 15 str	0.11	0.14	0.12	0.01	0.11
3 15 str	0.18	0.22	0.19	0.02	0.19
1 17 str	0.18	0.21	0.19	0.01	0.19
2 17 str	0.10	0.12	0.11	0.00	0.11
3 17 str	0.08	0.14	0.11	0.02	0.11
1 20 str	0.08	0.14	0.11	0.02	0.12
2 20 str	0.18	0.24	0.20	0.03	0.18
3 20 str	0.15	0.21	0.17	0.02	0.16

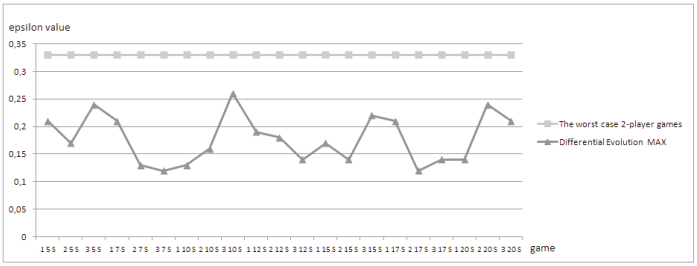


Fig. 3. The worst ϵ value for different games - comparision with the worst ϵ for 2 person games

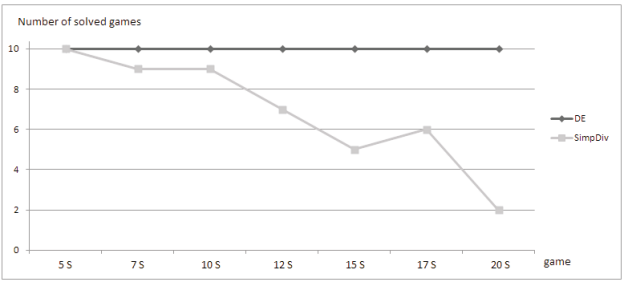


Fig. 4. Number of solved games - the DE and the Simplicial Subdivision comparision

10 different games only 2 were solved by the Simplicial Subidivision algorithm. On the other hand, the Differential Evolution solved all games.

6 Conclusions

In this article we proposed the new mutation schema, which was used in the Differential Evolution algorithm. The proposed method was used to solve the difficult problem of finding the Nash equilibrium for n players, where n is greater than 2. Many existing mathematical methods point to the fact that game theory isa branch of pure mathematics. Moreover, those methods are only effective for the 2 person games and they are strictly directed on the specific classes of the problems. In the literature exist only few algorithms dedicated to n person games. One of those algorithms is the Simplicial Subdivision algorithm. It is one of the main state-of-art algorithms (next to the Lemke-Howson algorithm). Despite the fact that the Lemke-Howson was created 50 years ago, it is still very popular. One of the main disadvantages is the maximum number of players involved, which equals 2. Our method was compared with the Simplicial Subdivision, because this algorithm seems to be the most flexible and general approach. In recent years, evolutionary algorithms and similar methods allow us to partialy solve problems considered purly mathematical. One of our goals was to show a simple and efficient solution based on one of the famous approximate methods. We showed, that the DE is far more efficient than the Simplicial Subdivision for problem of finding the Nash equilibrium. The proposed method found approximate solutions for all prepared games. It is obvious that the Differential Evolution has a lot of potential in the problem of computing the Nash and approximate Nash equilibria. The possibility of simple modification of the fitness function allows us to consider much harder problmes than the classical Nash equilibrium.

Our next task is to compare the proposed DE schema with other existing schemas. The described problem is very specific, and at this moment we can't define more problems, in which proposed mutation schema could be used. In our experiments we used fixed values for the CR and the F parameters. An interesting problem is to find a way to improve results for the proposed solutions. Only few runs of the algorithm were able to find better solution that ϵ equal to 0.1. Still open remains question, if the Differential Evolution is able to find exact solutions for any random n person game. One of our next goals will be detailed comparision of the basic Differential Evolution and the method described in the article. Deep study of the basic DE adapted to the problem of finding approximate Nash equilibria should be very desirable for the future works.

References

1. Deng, C., Zhao, B., Yang, Y., Deng: A Novel Binary Differential Evolution without Scale Factor F. In: 2010 Third International Workshop on Advanced Computational Intelligence (IWACI), pp. 250–253 (2010)

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. Dickhaut, J., Kaplan, T.: A program for finding Nash equilibria, Working papers, University of Minnesota, Department of Economics (1991)
4. 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)
5. Liu, F., Qi, Y., Xia, Z., Hao, H.: Discrete Differential Evolution Algorithm for the Job Shop Scheduling Problem. In: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 879–882 (2009)
6. Kasemir, K.U.: Detecting Ellipses of Limited Eccentricity in Images with High Noise Levels. Image and Vision Computing 21(7), 221–227 (2003)
7. Lampinen, J., Zelinka, I.: Mixed Variable Non-Linear Optimization by Differential Evolution. In: Proceedings of Nostradamus (1999)
8. van der Laan, G., Talman, A.J.J., van Der Heyden, L.: Simplicial Variable Dimension Algorithms for Solving the Nonlinear Complementarity Problem on a Product of Unit Simplices Using a General Labelling. Mathematics of Operations Research, 377–397 (1987)
9. Lampinen, J., Zelinka, I.: On Stagnation of the Differential Evolution Algorithm. In: Proceedings of 6th International Mendel Conference on Soft Computing (2000)
10. Lemke, C.E., Howson, J.T.: Equilibrium Points of Bimatrix Games, vol. 12, pp. 413–423. Society for Industrial and Applied Mathematics (1964)
11. Magoulas, G.D., Vrahatis, M.N., Androulakis, G.S.: Effective Backpropagation Training with Variable Stepsize. Neural Networks 10, 69–82 (1997)
12. 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
13. McLennan, A.: The Expected Number of Nash Equilibria of a Normal Form Game. Econometrica 73, 141–174 (2005)
14. 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)
15. Pampara, G., Engelbrecht, A.P., Franken, N.: Binary Differential Evolution. In: IEEE World Congress on Computational Intelligence, Proceedings of the Congress on Evolutionary Computation, pp. 1873–1879 (2006)
16. Porter, R., Nudelman, E., Shoham, Y.: Simple Search Methods for Finding a Nash Equilibrium. Games and Economic Behavior 63, 642–662 (2008)
17. Price, K., Storn, R., Lampinen, J.: Differential Evolution: a Practical Approach to Global Optimization. Springer (2005)
18. 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)
19. Storn, R.: Differential Evolution Design of an IIR-Filter. In: IEEE International Conference on Evolutionary Computation, ICEC 1996, pp. 268–273 (1996)

Ant Colony Decision Forest Meta-ensemble

Urszula Boryczka and Jan Kozak

Institute of Computer Science, University of Silesia,
Będzińska 39, 41–200 Sosnowiec, Poland
{urszula.boryczka,jan.kozak}@us.edu.pl

Abstract. This paper is devoted to the study of an extension of Ant Colony Decision Tree (ACDT) approach to Random Forests (RF) – an arisen meta-ensemble technique called Ant Colony Decision Forest (ACDF). To the best of our knowledge this is the first time that Ant Colony Optimization is being applied as an ensemble method in data mining tasks. Meta-ensemble ACDF as a hybrid RF and ACO based algorithm is evolved and experimentally shown high accuracy and good effectiveness of this technique motivate us to further development.

Keywords: Ant Colony Optimization, Ant Colony Decision Trees, Decision Forest, Bagging, Random Forests.

1 Introduction

Data mining and machine learning have been the subject of increasing attention over the past 30 years. Ensemble methods, popular in machine learning and pattern recognition, are learning algorithms that construct a set of many individual classifiers, called base learners, and combine them to classify new data points or samples by taking a weighted or unweighted vote of their predictions. It is now well-known that ensembles are often much more accurate than the individual classifiers that make them up. The success of ensemble approaches on many benchmark data sets has raised considerable interest in understanding why such methods succeed and identifying circumstances in which they can be expected to produce good results. This article provide a summary of widely used heuristic methods and their modifications used in different benchmark problems and to identify its development of the random forests in reference to Ant Colony Decision Tree algorithm. Our goal is to design a new algorithm for construction of decision forests, where we obtain better accuracy of classification, especially in case of difficult data sets making explicit reference to Ant Colony Optimization.

This paper is organized as follows: section 2 presents an overview of related works on association classification and decision trees. Section 3 discusses an approach Random Forests as an example of ensemble methods. Section 4 describes the ACDF approach as a tool of meta-ensemble learning approach. Section 5 presents the experimental setup and methodology concerning the ACDF examination. Section 6 discusses results and conclusions are drawn in section 7.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 473–482, 2012.

2 Decision Trees

One of the most efficient and widely applied learning algorithms search the hypothesis (solution) space consisting of decision trees [11,14]. The term hypothesis is understood as a combination of attribute values which determine the way to undertake a specific decision. A decision tree learning algorithm searches the space of such trees by first considering trees that test only one attribute and making an immediate classification. Then they consider expanding the tree by replacing one of the leaves by a test of the second attribute. Various heuristics are applied to choose which test to include in each iteration and when to stop growing the tree [5]. The evaluation function for decision trees will be calculated according to the following formula:

$$Q(T) = \phi \cdot w(T) + \psi \cdot a(T, P) \tag{1}$$

where:

$w(T)$ – the size (number of nodes) of the decision tree T,
$a(T, P)$ – the accuracy of the classification samples from a test set P by the tree T,
ϕ and ψ – constants determining the relative importance of $w(T)$ and $a(T, P)$.

Constructing optimal binary decision trees is an NP–complete problem, where an optimal tree is one which minimizes the expected number of tests required for identification of the unknown samples, as shown by Hyafil and Rivest in [10]. Classification And Regression Tree (CART) approach was developed by Breiman et al. in 1984 [6].

Twoing criterion, firstly proposed in CART, will search for two classes that will make up together more then 50% of the data. Twoing splitting rule maximizes the following change-of-impurity measure which implies the following maximization problem for nodes m_l, m_r:

$$\underset{a_j \leq a_j^R, j=1,\ldots,M}{\arg\max} \left(\frac{P_l P_r}{4} \left[\sum_{k=1}^{K} |p(k|m_l) - p(k|m_r)| \right]^2 \right), \tag{2}$$

where:

$p(k|m_l)$, $p(k|m_r)$ – the conditional probability of the class k provided in node m_l, m_r,
P_l, P_r – the probability of transition samples into the left or right node m_l, m_r,
K – number of decision classes,
a_j – j–th variable, a_j^R is the best splitting value of variable a_j.

3 Decision Forests

A decision forest is a collection of decision trees [5,7,14]. We defined the decision forest by following formula:

$$DF = \{d_j : X \to \{1, 2, ..., g\}\}_{j=1,2,\ldots,J}, \tag{3}$$

where J is a number of decision trees j ($J \geqslant 2$).

In decision forests, predictions of decision trees are combined to make the overall prediction for the forest. Classification is done by a simple voting. Each decision tree votes on the decision for the sample and the decision with the highest number of votes is chosen. The classifier created by a decision forest *DF*, denoted as $dDF : X \to 1, 2, ..., g$, uses the following voting rule:

$$dDF(x) := \arg\max_k N_k(x), \tag{4}$$

where k a decision class, such that $k \in \{1, 2, \ldots, g\}$; $N_k(x)$ is the number of votes for the sample $x \in X$ classification in to class k, such that $N_k(x) := \#\{j : d_j(x) = k\}$.

3.1 Bagging

Ensemble methods work by running a base algorithm multiple times, and forming a vote out of the resulting hypotheses. There are two main approaches to designing ensemble learning algorithms. The first approach is to construct each hypothesis independently in such a way that the resulting set of hypotheses is accurate and diverse. The second approach to designing ensembles is to construct the hypotheses on a coupled fashion so that the weighted vote of the hypothesis gives a good fit to the data. One way to fulfill this task – construct multiple hypotheses is to run the algorithm several times and provide it with somewhat different data (e.g. bootstrap samples) in each run.

Good example of such a method is Bagging – "Bootstrap Aggregating" method [9], firstly introduced by Breiman [4]. This approach works as follows. Given a set of n training data (learning set), Bagging chooses in each iteration a learning set of size n by sampling uniformly with replacement from the original data set. Each element of such a learning set can be chosen exactly with the same probability equal to $\frac{1}{n}$.

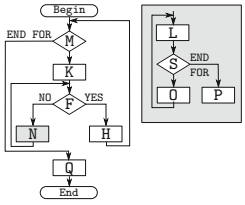
To aggregate the base classifiers in a consensus manner, strategy such as voting is commonly used. Assuming the result of the base classifiers are independent of each other, each of the base classifier give exactly one vote and finally the simple voting decides about the classification the samples (see formula (4)).

3.2 Random Forests

Some ensemble methods such as Random Forests are particularly useful for high-dimensional data sets because increased classification accuracy. It can be achieved by generating multiple prediction models each with a different subset of learning data consisted of attribute subsets [5].

Breiman provides a general framework for tree ensembles called "random forests" [5]. Each tree depends on the values of a randomly chosen attributes, independently for each node or tree and with the same distribution for all trees. Thus, a random forest is a classifier (ensemble) that consists of many decision

trees. Each splitting rule is performed independently for different subset of attributes. As a result it could be chosen m attributes from the p descriptions of the learning samples. Assume that $m \ll p$, and according to the performed experiments, good results should be obtained when $m = \sqrt{p}$. Let we assume, that $\frac{1}{3}$ of samples cannot be chosen to the training sample (in accordance to the probability equal to $(1-n)^n \approx e^{-n}$), so only $\frac{1}{3}$ trees in the analyzed forest are constructed without this sample. In this situation, Breiman proposed that it will be well-grounded to apply the unencumbered estimator of misclassification probability obtained by decision tree [5]. Diagram of the construction decision forest is presented in Fig. 1.

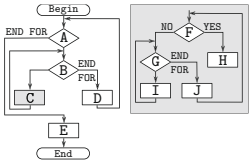


F- the completed tree
H- calculate the accuracy rate
K- create the pseudo-samples
L- create the subset of attributes
M- for each decision tree
N- select the next division
O- calculate the splitting rule value
P- select the best division
Q- the decision forest
S- for pseudo-samples and subset of att

Fig. 1. Diagram of the construction random forest

4 Ant Colony Decision Trees Algorithm

Ant Colony Optimization (ACO) approach has been successfully applied to many difficult combinatorial problems. Ant Colony Decision Trees (ACDT) algorithm is the first ACO adaptation to the task of rule induction and constructing decision trees, but also rule induction approach – Ant-Miner [3,12,13].



A- for each colony of ants
C- build tree
E- the decision tree
G- for all samples and attributes
I- select the division

B- for each angent-ant
D- update pheromone trail
F- the completed tree
H- calculate the accuracy rate
J- calculate the probability of choosing the appropriate split

Fig. 2. Construction the tree by ACDT algorithm

In each ACDT step an ant chooses an attribute and its value for splitting the samples in the current node of the constructed decision tree. The choice is made according to a heuristic function and pheromone values. The heuristic function is based on the Twoing criterion (equ. (2)), which helps ants select an attribute-value pair which well divides the samples into two disjoint sets, i.e. with the intention that samples belonging to the same decision class should be put in the same subset. The best splitting is observed when similar number of

samples exists in the left subtree and in the right subtree, and samples belonging to the same decision class are in the same subtree. Pheromone values indicate the best way (connection) from the superior to the subordinate nodes – all possible combinations are taken into account.

The diagram of the proposed algorithm is presented in Fig. 2. As mentioned before, the value of the heuristic function is determined according to the splitting rule employed in CART approach (see formula (2)). The probability of choosing the appropriate split in the node is calculated according to a classical probability used in ACO [8]:

$$p_{i,j} = \frac{\tau_{m,m_{L(i,j)}}(t)^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{i}^{a} \sum_{j}^{b_i} \tau_{m,m_{L(i,j)}}(t)^{\alpha} \cdot \eta_{i,j}^{\beta}}, \tag{5}$$

where:

$\eta_{i,j}$ – a heuristic value for the split using 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 determined values 1 and 3, respectively.

The initial value of the pheromone trail is determined similarly to the Ant–Miner approach and depends on the number of attribute values. The pheromone trail is updated (6) by increasing pheromone levels on the edges connecting each tree node with its parent node:

$$\tau_{m,m_L}(t+1) = (1-\gamma) \cdot \tau_{m,m_L}(t) + Q(T), \tag{6}$$

where $Q(T)$ is a quality of the decision tree (see formula (1)), and γ is a parameter representing the evaporation rate, equal to 0.1.

5 Ant Colony Decision Forest

A computational problem arises when the proposed algorithm cannot guarantee to find the best hypothesis within hypotheses space. In ACO and RF approaches, the task of finding the suitable hypothesis that best fits the training data is computationally intractable, so more sophisticated method should be employed in this situation. An algorithm ACDF proposed in this paper is based on two approaches: RF and ACDT. The ACDF algorithm can be applied for difficult data sets analysis by adding randomness to the process of choosing which set of features or attributes may be distinguish during the construction decision trees.

In case of the ACDF, agent-ants create a collection of hypotheses in random manner complying the threshold or rule to split on. The challenge is to introduce a new random subspace method for growing collections of decision trees – it means that agents-ants can create the collection of hypotheses from the hypothesis space using random-proportional rules. At each node of the tree agent-ant choose from the random subset (random pseudo-samples) of attributes and then constrain the tree-growing hypothesis to choose its splitting rule from among this subset. Because of the re-labeled randomness proposed in our approach we

Table 1. Variants of ACDF

Version	Trees in forest	Data sets	Attribute sets
ACDF_1 (Fig. 3 b)	local best trees	individual, for agent-ant	individual, for agent-ant
ACDF_2 (Fig. 3 b)	independently, global best trees	individual, for agent-ant	individual, for agent-ant
ACDF_3 (Fig. 3 c)	local best trees	collective, for colony	collective, for colony
ACDF_4 (Fig. 3 d)	independently, global best trees	collective, for colony	collective, for colony
ACDF_5 (Fig. 3 e)	independently, global best trees	independently, for colony	independently, for colony
ACDF_6 (Fig. 3 f)	local best trees	collective, for colony	for each node, as in RF
ACDF_7 (Fig. 3 g)	independently, global best trees	all samples	all attributes

resign from the different subsets of attributes chosen for each agent-ant or colony to favor of greater stability of the undertaken hypotheses. It is a consequence of the proposition firstly used in RF.

A proposed approach that suffer from the representational problem is said to have a good diversity in (random pseudo-samples) training and testing samples and balance in decision making. ACDF is characterized as an algorithm with high diversity, because agents-ants make a cascade of choices consist of attribute and value choosing at each internal node in the decision tree (for creating a special hypothesis). Consequently, ensembles of decision tree classifiers perform better than individual decision trees. It is due to the independently performed exploration/exploitation the subspace of hypotheses.

Seven different variants of ACDF algorithms have been proposed in this article, differ in the way of preparing the pseudo-samples and subsets of attributes and the best decision tree choosing. The proposed versions are presented in Tab. 1.

In ACDF_1 each agent-ant can search different subspaces of hypotheses space. The stability of decision tree can be obtained during this searching process and it is identical as in version ACDF_2. In this version we propose independently creating decision trees by each of colony of ants, so the trees are not very similar. Smaller distribution in hypotheses space for version 3 and 4 is due to the same sets of samples and attributes chosen for each colony of ants. As in these two cases, the version ACDF_5 is also characterized by the smaller distribution in the space, so the pheromone values deposited in the subtrees play more significant role. The most interested version is presented in ACDF_7, where the meaningful similarity to RF is presented. The decision trees are created by agent-ants independently and we may analyzed here only the inner-colony cooperation among agent-ants. Independent colonies have not the possibility to communicate with each other, so in consequence the diversity of decision trees may be obtained.

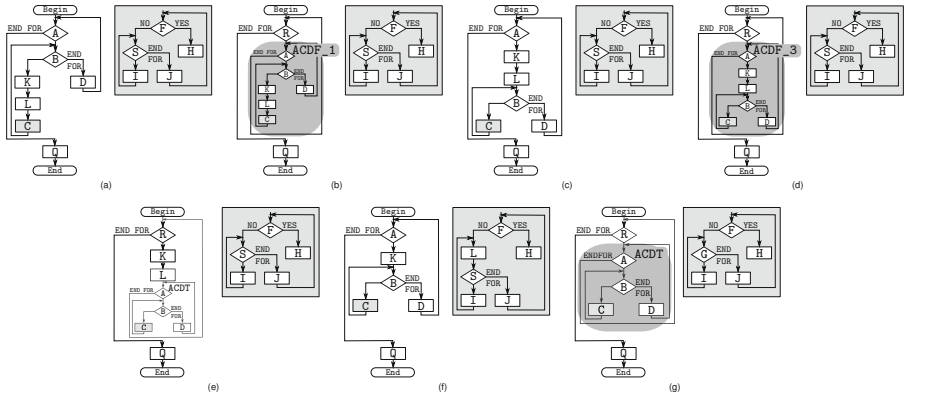


Fig. 3. Diagram of the construction ACDF forest

6 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 Tab. 2) performed for the following adjustments. We have performed 30 experiments for each data set. Each experiment included 1250 generations with the population size of ant colony equal to 50. In each case, the decision forest consists of 25 trees. Comparative study of ACDT algorithm (described in [1,2]) with seven different versions described in section 5 have been performed for examination this new approach.

Evaluation of the performance behavior of ACDF was performed using 12 public–domain data sets from the UCI (University of California at Irvine) data set repository available from: `http://archive.ics.uci.edu/ml/`. The data sets larger than 1000 samples (balance-scale, breast-cancer, breast-tissue, cleveland, heart) are divided into two groups in a random way: training and testing sets, appropriately. The data sets less than 1000 samples (the other) are estimated by 10-fold cross-validation.

In most experimental studies, the algorithms: ACDF_1 – ACDF_7 give the better performance on the vast majority of data sets, when comparing this approach with ACDT algorithm (see Tab. 2 and Fig. 4). The exact reasons for ACDF's success are not fully understood. One line of explanation is based on the many independently performed searchings and the process of learning which take place among agent-ants via pheromone values (especially the ACDF_5 algorithm, where a single decision trees are weak in the context of classification, but the forest is very good in the case of accuracy of classification).

Analyzing the performance of the ACDF_6 approach, we observe that the obtained results are interesting – we noted high accuracy of classification in

comparison with the rest of analyzed versions (ACDF_1 –ACDF_5). The first observation that can be made in the case of ACDF_7, is that these results show that it is better to use independent colonies and separately analyzed pheromone tables. This approach usually gives better results in all cases (in the context of accuracy of classification). It could be analyzed in the Fig. 5, where the number of nodes is presented. Concerning the two methods: ACDT (with ACDT-forest) and ACDF_7, these techniques are characterized by the same number of nodes in created decision trees. When compared ACDT with the rest versions of ACDF, the number of nodes have been significantly diminished. Analyzing the results in term of accuracy of classification - prediction about new samples, the approaches ACDF_1 to _6 are promising but not very specific.

The results in Tab. 2 confirm that the performance of the presented new approach is significantly interested and unaffected by local minima in hypotheses space. Nevertheless these results also show that the use of meta-ensemble is able to mitigate the problem of loosing diversity in the tree structure occurring in the arisen forest.

Table 2. Comparative study – accuracy rate

Data set	ACDT – tree		ACDT – forest		ACDF_1		ACDF_2	
	acc	#n	acc	#n	acc	#n	acc	#n
heart	0.7744	13.0	0.7628	253.4	0.8269	121.8	0.8311	133.5
breast-cancer	0.7165	6.5	0.7284	82.1	0.7363	68.0	0.7308	83.6
balance-scale	0.7821	51.6	0.8003	1198.8	0.7877	440.5	0.8343	440.7
dermatology	0.9339	7.5	0.9314	113.7	0.9290	232.6	0.9122	225.0
hepatitis	0.7989	5.0	0.8005	43.4	0.8190	48.6	0.8085	68.4
breast-tissue	0.4702	12.3	0.4611	205.1	0.4810	138.3	0.4782	124.1
cleveland	0.5401	15.7	0.5456	338.7	0.5660	133.1	0.5578	136.5
b-c-w	0.9301	9.3	0.9306	103.2	0.9363	245.4	0.9441	131.8
lymphography	0.7828	8.3	0.7821	161.6	0.8524	183.6	**0.8857**	116.0
shuttle	0.9971	62	0.9975	1508	0.9969	3464	0.9965	5557
mushroom	0.6309	71	0.6313	1604	**0.6506**	654	0.6426	633
optdigits	0.8021	151	0.8751	4273	0.9111	4222	0.8999	4112

Data set	ACDF_3		ACDF_4		ACDF_5		ACDF_6		ACDF_7	
	acc	#n	acc	#n	acc	#n	acc	#n	acc	#n
heart	0.8147	120.2	0.8194	129.4	0.8191	155.1	**0.8391**	194.8	0.7781	257.8
breast-cancer	0.7367	74.2	0.7303	85.9	0.7313	92.7	**0.7414**	96.0	0.7308	122.9
balance-scale	0.8209	416.5	0.8180	428.4	0.8299	426.9	**0.8559**	884.3	0.8202	1135.7
dermatology	0.9114	231.2	0.9033	225.0	0.9135	242.8	0.9690	297.7	**0.9765**	272.1
hepatitis	0.8089	60.1	0.8045	70.7	0.8152	77.8	**0.8210**	58.4	0.8145	79.9
breast-tissue	0.4769	135.4	0.4710	121.7	0.5029	151.7	0.4919	146.0	**0.5167**	198.5
cleveland	0.5549	128.4	0.5575	133.7	0.5590	165.7	0.5737	202.9	**0.5795**	291.4
b-c-w	**0.9544**	428.4	0.9514	207.4	0.9519	146.1	0.9280	275.1	0.9505	127.1
lymphography	0.8068	200.2	0.7857	102.2	0.7782	115.5	0.8265	318.5	0.8122	167.5
shuttle	0.9960	7890	0.9957	7875	0.9934	7992	0.9972	2458	**0.9976**	1510
mushroom	0.6383	604	0.6398	592	0.6420	579	0.6475	1007	0.6331	1617
optdigits	0.8843	4236	0.8882	4157	0.8862	4725	**0.9442**	3942	0.9429	4235

Abbrev.: acc – accuracy rate; #n – number of nodes.

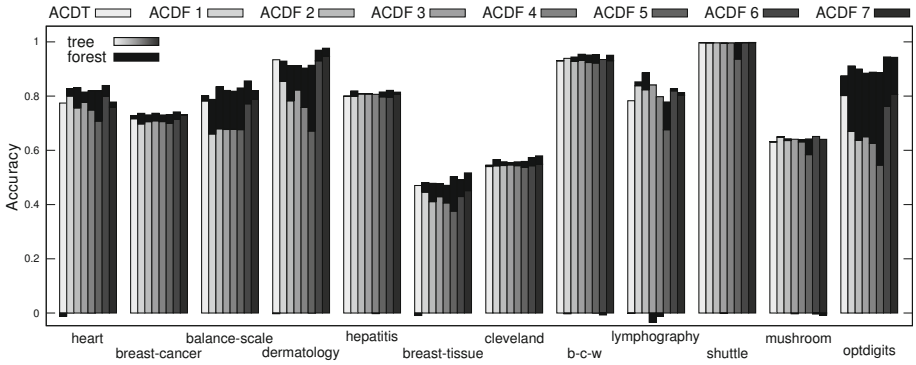


Fig. 4. Accuracy rate of decision tree and decision forest

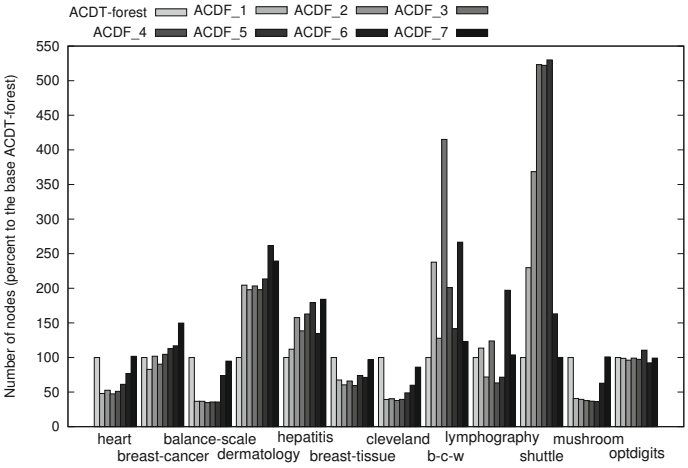


Fig. 5. Number of nodes (percent to the base ACDT-forest)

7 Conclusions

In this work we addressed the problem of classification in data mining tasks. A meta-ensemble approach combined with ACO is followed. We carry out an extensive experimental study to analyze the performance of different variants of our proposition, including seven, that were proposed here. The new approach for data classification as well as decision making – ACDF is clearly a promising method for this task. Additionally, although the method is general, it remains to be investigated if it also achieves competitive results in other data sets. Nevertheless the results indicate the motivation which led its development is relevant, so we need to investigate the reasons for its learning mechanisms via pheromone and improve this by new augmentations.

Other conclusions can be made concerning the behavior of meta-ensemble, particularly the selection or choosing pseudo-samples from learning-samples.

The size of forest – the number of trees in presented results is the next issue, we want to focus on. Secondly, the concept of weighting: constant, non constant and dynamically or adaptively weighting techniques during the pseudo-samples creation is particularly relevant in order to increase the robustness of the ensemble predictions.

These results provided by the experimental study are satisfying and interesting. The cost of this situation is the complexity of the system, so we decide to concentrate on the simplification the ACDF algorithm, constituting our future work. Statistical (The Wilcoxon Two Sample Test) evidence confirms the results.

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.T. (eds.) ICCCI 2010, Part I. LNCS, vol. 6421, pp. 373–382. Springer, Heidelberg (2010)
2. Boryczka, U., Kozak, J.: An Adaptive Discretization in the ACDT Algorithm for Continuous Attributes. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 475–484. Springer, Heidelberg (2011)
3. Boryczka, U., Kozak, J., Skinderowicz, R.: Parellel Ant–Miner. Parellel implementation of an ACO techniques to discover classification rules with OpenMP. In: 15th International Conference on Soft Computing - MENDEL 2009, pp. 197–205. University of Technology, Brno (2009)
4. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
5. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (2001)
6. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman & Hall, New York (1984)
7. Bühlmann, P., Hothorn, T.: Boosting algorithms: Regularization, prediction and model fitting. Statistical Science 22(4), 477–505 (2007)
8. Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.): ANTS 2008. LNCS, vol. 5217. Springer, Heidelberg (2008)
9. Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7(1), 1–26 (1979)
10. Hyafil, L., Rivest, R.: Constructing optimal binary decision trees is NP–complete. Inf. Process. Lett. 5(1), 15–17 (1976)
11. Murphy, O., McCraw, R.: Designing Storage Efficient Decision Trees. IEEE Transactions on Computers 40, 315–320 (1991)
12. Otero, F.E.B., Freitas, A.A., Johnson, C.G.: *c*Ant-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)
13. Otero, F.E.B., Freitas, A.A., Johnson, C.G.: Handling continuous attributes in ant colony classification algorithms. In: CIDM, pp. 225–231 (2009)
14. Rokach, L., Maimon, O.: Data Mining With Decision Trees: Theory and Applications. World Scientific Publishing (2008)

Ant Colony System with Selective Pheromone Memory for TSP

Rafał Skinderowicz

Institute of Computer Science, Silesia University, Sosnowiec, Poland
rafal.skinderowicz@us.edu.pl

Abstract. Ant Colony System (ACS) is a well known metaheuristic algorithm for solving difficult optimization problems inspired by the foraging behaviour of social insects (ants). Artificial ants in the ACS cooperate indirectly through deposition of pheromone trails on the edges of the problem representation graph. All trails are stored in a pheromone memory, which in the case of the Travelling Salesman Problem (TSP) requires $O(n^2)$ memory storage, where n is the size of the problem instance. In this work we propose a novel *selective pheromone memory model* for the ACS in which pheromone values are stored only for the selected *subset* of trails. Results of the experiments conducted on several TSP instances show that it is possible to significantly reduce ACS memory requirements (by a constant factor) without impairing the quality of the solutions obtained.

Keywords: ant colony system, selective pheromone memory.

1 Introduction

Ant colony system (ACS) belongs to a wide range of metaheuristic search algorithms used to solve a variety of difficult combinatorial optimization problems. The algorithm is an improved version of the ant colony optimization algorithm (ACO) which was inspired by the behavior of some species of ants. The ants deposit pheromone trails to guide other ants when searching for a food source [5]. In the ACO artificial ants incrementally construct candidate solutions. Selection of the solution components is based on the heuristic information about the problem and the artificial pheromone trails. The artificial pheromone trails form a parametric probabilistic model that is updated based on previously constructed solutions [10]. ACO algorithms were tested on a large number of academic and real-world problems and proved to obtain very good performance on many of them [1].

The artificial pheromone trails represent the "common" memory of ants and are used to guide the process of constructing candidate solutions, therefore they play essential role for the performance of ACO. A lot of research was focused on improving the pheromone trails update rules and finding better minimum and maximum pheromone levels in order to prevent the premature stagnation of the search process. The ACS is a modified version of the ACO, in which a few

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 483–492, 2012.

important changes were introduced, among them addition of local pheromone update rule [4]. Stüzle and Hoos developed the MAX-MIN Ant System (MMAS) which includes explicit upper and lower limits on the pheromone trail values to avoid stagnation [9]. In the work of Matthews [6] improved lower limits for pheromone trails were proposed for the MMAS and ACS. A summary of the most important advances may be found in [3].

Obviously, storing the artificial pheromone trails requires a certain amount of the computer's memory, depending on the complexity and size of the tackled problem. For example, in the case of TSP the ACS requires a pheromone trails memory of size proportional to n^2, where n is the size of the problem instance, equal to the number of cities. If one tries to solve large problem instances, the computer's RAM memory capacity may quickly become an obstacle. For example, in case of *pla85900* instance from the TSPLIB [8] the number of cities equals to $n = 85900$, which results in memory storage requirement of $\approx 5.9 \cdot 10^{10}$ bytes (nearly 55 GB), if double precision (64 bits) numbers are used. In this work we investigate a simple idea of allowing only a limited number of pheromone trails to be stored at the same time in the computer's memory, in order to reduce the memory storage requirements of the ACS algorithm.

The rest of the paper is organized as follows. In Section 2, we briefly describe the artificial pheromone memory model used in the ACO and ACS. Section 3 is devoted to the proposed pheromone memory model in which (separate) pheromone values are stored only for a selected subset of trails. Section 4 contains the discussion of the experimental results. Section 5 concludes the work.

2 Pheromone Memory Model

Prior to defining the pheromone memory model, we define a *model* of the combinatorial optimization (CO) problem being solved, following the notation presented in [3]. A model $\mathcal{P} = (\mathcal{S}, \Omega, f)$ of the CO problem consist of:

- a solution space $\mathcal{S}$ defined over a set of discrete decision variables and a set Ω of constraints among the variables;
- an objective function $f : \mathcal{S} \to \mathbb{R}^+$ to be minimized.

A set of n *discrete decision variables* X_i with values $v_i^j \in D_i = \{v_i^1, v_i^2, \ldots, v_i^{|D_i|}\}$, $i = 1, \ldots, n$, is given, where n denotes the size of the problem. A feasible solution $s \in \mathcal{S}$ to the problem is a complete assignment of the variables to values taken from their domain, that satisfies the constraints. A feasible solution $s^* \in \mathcal{S}$ is called a *global optimum*, if $f(s^*) \leq f(s)\ \forall s \in \mathcal{S}$. The set of all s^* is denoted by $\mathcal{S}^* \in \mathcal{S}$. Solving the given CO problem requires finding a solution $s^* \in \mathcal{S}^*$.

A combination of a decision variable X_i and one of its domain values v_i^j is a *solution component* denoted by c_i^j. The pheromone model consists of a *pheromone trail parameter* $\mathcal{T}_i^j$ for each solution component c_i^j. The set of all solution components is denoted by $\mathcal{C}$. The value of a pheromone trail parameter $\mathcal{T}_i^j$ is denoted by τ_i^j. The vector of values for all pheromone trail parameters is denoted by $\mathcal{T}$. Size of the vector $\mathcal{T}$ is equal to $|\mathcal{T}| = \sum_{i=1}^{n} |D_i|$.

Pheromone model for the TSP is defined as follows. A fully connected graph $G(V, E)$ is given, where $V = \{1, 2, \ldots, n\}$ is the set of vertices representing cities and $E = \{e_{ij} : i, j \in V \wedge i \neq j\}$ is the set of edges which represent connections (roads) between the cities. With each edge e_{ij} there is a positive weight d_{ij} associated equal to the distance between the pair of cities (i, j). The goal consists in finding a Hamiltonian cycle in G with the minimum sum of edge weights. Model for the TSP problem consists of n decision variables X_i with domains $D_i = \{v_i^j : e_{ij} \in E\}$. A variable assignment $X_i = v_i^j$ means that edge e_{ij} is a part of the corresponding solution. The set of constraints Ω is defined, such that only Hamiltonian cycles are valid solutions. The set of solution components $\mathcal{C}$ consists of a solution component c_i^j for each combination of variable X_i and domain value v_i^j. The pheromone model $\mathcal{T}$ is a vector of pheromone trail parameters $\mathcal{T}_i^j$, in which value τ_i^j corresponds to solution component c_i^j. In other words, for each city i there is one decision variable X_i with value from the set D_i. The size of the pheromone vector is equal to $|\mathcal{T}| = \sum_{i=1}^{n} |D_i| = \sum_{i=1}^{n} n - 1 = n(n-1) = O(n^2)$.

2.1 ACS for TSP

In a single step of the ACS for TSP each of the ants constructs a solution as follows. Firstly, it is placed in a randomly selected node (city). Next, the ant constructs a tour in the TSP graph by moving in each construction step from its current node to one of the unvisited nodes. At each step the traversed edge is added to the partial solution s^p. The next solution component c_i^j for the ant k placed in the node i is selected according to the formula:

$$j = \begin{cases} \arg\max_{l \in J_k^i} [\tau_i^l] \cdot [\eta(c_i^l)]^\beta, & \text{if } q \leq q_0 \\ J, & \text{if } q > q_0, \end{cases} \tag{1}$$

where $\eta(c_i^l)$ is a heuristic knowledge associated with component c_i^l, J_k^i is a list of unvisited (candidate) cities of the ant k, q_0 is a parameter, $q \in [0, 1]$ is a random number and J is a city selected with probability:

$$P(J|i) = \frac{[\tau_i^J] \cdot [\eta(c_i^J)]^\beta}{\sum_{l \in J_k^i} [\tau_i^l] \cdot [\eta(c_i^l)]^\beta}. \tag{2}$$

Parameter q is an enhancement introduced in the ACS in order to drive algorithm's search process towards areas of the problem solution space containing the solutions of high quality. It is based on the implicit assumption that solutions of high quality have similar structure, i.e. share many components, which appears to be true for many of the difficult optimization problems [3]. If the value of q is high, the choice given by the (1) is mostly deterministic and focused on the exploitation driven by the possessed knowledge – heuristic and accumulated in the pheromone trails memory.

There are two kinds of pheromone trails updates performed in the ACS. The first is called *local pheromone update* and is performed after an ant has selected a new solution component, represented by an edge in the graph $G(V, E)$.

The second is called *global pheromone update* and is performed after all ants have completed construction of their solutions. It consists in deposition of additional pheromone on the trails associated with the components of the highest quality solution found to date.

3 Limiting the Size of the Pheromone Memory

Dorigo el al. [4] noticed in the study of the ACS for TSP that the search process is focused on a subset of edges (components) which belong to the last best solution and those which do not belong to it, but which did in one of the last few iterations. On those edges the pheromone concentration is high compared to the rest of the edges, which have low pheromone values and therefore very small probability of being selected, according to (1).

Our idea is to limit the pheromone trails memory to a subset of m selected *important* pheromone parameters. Only for the selected parameters (each associated with an edge in TSP graph) the pheromone values are stored as described in Section 2. The rest of the pheromone parameters receive a value of τ_0 (i.e., initial pheromone value).

More formally, the pheromone model vector $\mathcal{T}$ is replaced with the combination of a vector $\hat{\mathcal{T}}$ and a set U. The set U is used to store the indices of the *selected* pheromone parameters in the vector $\hat{\mathcal{T}}$ and is defined as:

$$U = \{u_i | i \in \{1, \ldots, m\} \wedge u_i \in W\}, \tag{3}$$

where m is a positive natural number, $W = \{(j,k) | j,k \in \{1,\ldots,n\}\}$ and n is the number of decision variables in the model for the tackled problem, as described in Section 2. For each of the pheromone parameters, $\hat{\tau}_i^j$, such that $\exists u_k \in U \; u_k = (i,j)$, the values are set like in the *unmodified* (original) pheromone memory model (see Section 2). The rest of the parameters, for which there are no corresponding elements in U, are considered *non-important* and have the values equal to τ_0. Notice, that the vector $\hat{\mathcal{T}}$ has exactly the same size as the original vector $\mathcal{T}$, therefore there is no need to change the equations (1) or (2). Memory saving comes from the fact that in the actual implementation only the entries $\hat{\mathcal{T}}_i^j$ whose indices are in the set U have to be stored in the computer's memory.

A few questions arise:

- how to set the value of m, i.e. size of the selective pheromone memory;
- how to select the edges for which the pheromone parameters values should be remembered, i.e. how to set the elements in U;
- should the selection (equal to the set U) remain static or should it change as computations are performed (dynamic selection).

It is difficult to answer the first question because the ACS algorithm has many parameters which affect its performance. In our work we set the size, m, of the selective pheromone memory as the function of the problem size. Giving answers

to the second and third questions is even more difficult without the prior knowledge of the solution search space. Obviously, if only a subset of the pheromone trail parameters can be stored at the same time in the memory, it should contain those which are *important*, i.e. the parameters associated with the edges which are components of the high quality solutions. Of course, such knowledge is not available prior to solving the problem, therefore in our approach the *initial* pheromone memory starts with an empty set U, hence all the pheromone parameters values $\hat{\tau}_i^j$ are equal to τ_0.

In the course of the computations, as the local or global pheromone updates are performed, assignment of a new (or updated) value to the pheromone trail parameter $\hat{\mathcal{T}}_i^j$ is made according to the algorithm presented in Fig. 1. If the parameter is already in the memory, it simply receives an updated value (line 5). If it is not in the memory (i.e., no suitable entry in U exists), but the memory is not full it is simply added to it. Otherwise, *the least important* pheromone parameter is selected (line 8) and its value is reset to τ_0. In the actual implementation, it means that entry for the selected parameter can be removed from the computer's memory making place for the new entry, i.e. the updated pheromone parameter $\hat{\mathcal{T}}_i^j$. Selection of the least important pheromone trail parameter is made according to the given *selection criterion* (SC).

```
1  Input: (T̂, U) – selective pheromone memory
2  Input: τ̂_i^j – a new value for the pheromone parameter T̂_i^j
3  Input: SC – a criterion for selecting the least important parameter from (T̂, U)
4  if (i, j) ∈ U then {Entry for T̂_i^j exists in the memory}
5      T̂_i^j := τ̂_i^j {Update pheromone parameter's value}
6  else {No entry for T̂_i^j exists in the memory, create one}
7      if |U| ≥ m then {Is the memory full?}
8          T̂_k^l := select_pheromone_parameter(SC)
9          T̂_k^l := τ_0 {In the actual implementation an entry for T̂_k^l may be
               removed from the computer's memory}
10         U := U − {(k, l)}
11     end
12     U := U ∪ {(i, j)}
13     T̂_i^j := τ̂_i^j {In the actual implementation an entry for T̂_i^j is created
           in the computer's memory}
14 end
```

Fig. 1. Procedure for updating the *selective pheromone memory* model $(\hat{\mathcal{T}}, U)$ using the given selection criterion SC

We investigated two selection criteria. The first is the *minimum pheromone value* criterion (MPV), which is based on the intuition that the least important pheromone trail parameters are the ones with the lowest pheromone values and the most important are those with high pheromone values. The second criterion was inspired by the *least recently used* (LRU) algorithm, which is widely used in

the cache memory implementations [7]. It is based on the assumption that the oldest entry is the least important. Its main drawback is that it requires storing the *age* of each entry, i.e. time of its last usage. In our work, the age simply equals to the time of the last change of a parameter's value, resulting from the local or global pheromone updates.

Both criteria try to preserve the most important pheromone trail values, so that the ant colony may still find high quality solutions despite having only "partial" memory. Obviously, both criteria result in additional computational overhead compared to the original ACS pheromone memory model, which can be implemented simply as an array. The MPV criterion requires a data structure providing efficient methods for finding the element (pheromone trail parameter) with the lowest value and updating the value of any element. It may be implemented using the binary heap and hash table combination, resulting in time complexity for both operations equal to $O(\log |U|) = O(\log m)$, where m is the maximum size of the selective pheromone memory. The LRU criterion may be implemented using hash table and linked list combination, resulting in the amortized $O(1)$ time complexity [7].

4 Experiments

In order to investigate how the proposed limitation of the pheromone memory size affects the ACS performance a number of experiments were conducted using a few tests from the well known TSPLIB repository [8]. Namely, tests *kroA100, tsp225, lin318, u574, rat783* and *pcb1173* were selected. As mentioned in Section 1, limiting the size of the pheromone memory enables to run ACS algorithm on much larger instances, but the goal at this stage of the research was to compare the quality of solutions obtained using different pheromone memory variants and different values of parameters.

ACS algorithm has several parameters which affect its performance. All the computations were performed with the following parameters values: 2500 algorithm iterations, $\beta = 2$, $\rho_g = 0.2$ (global evaporation rate), $\rho_l = 0.01$ (local evaporation rate), $q_0 = 0.9$. Most of the values were set following the work presented in [4]. Also the so-called nearest neighbor candidate list was used (of size 20) to further reduce the algorithm runtime as described in [5]. For each problem instance and a set of parameters values algorithms were run 30 times.

First part of the experiments concerned the question how many of the pheromone trail parameters are replaced (lost) during each algorithm iteration (line 9 in Fig. 1) as the result of the limited pheromone memory capacity. The algorithm was run for both selection criteria (MPV and LRU) with memory size limit linearly dependent on the size of the problem: $m = r_l n^2$, with $r_l \in \{0.02, 0.04, \ldots, 0.1\}$. The calculations were repeated with the two ant colony sizes: $\lceil 0.1n \rceil$ and $\lceil 0.2n \rceil$. As can be seen in Fig. 2 the number of replaced pheromone parameters strongly depends on the number of ants because the more ants are used, the higher probability of traversing an edge in the TSP graph for which there is no corresponding pheromone trail parameter in the memory.

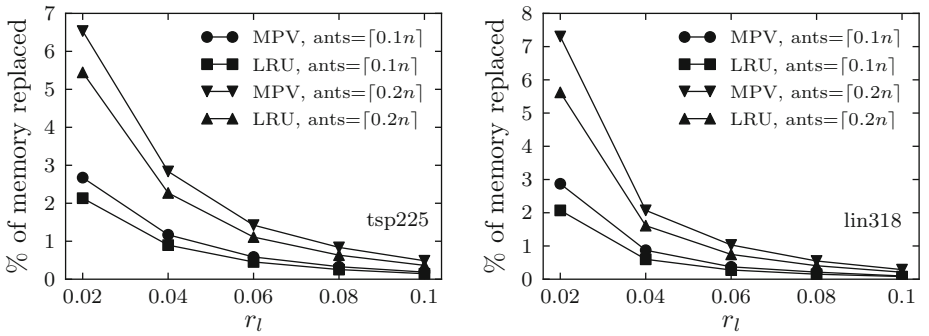


Fig. 2. Avg. percent of pheromone memory replaced during a single iteration of the ACS with selective pheromone memory for both replacement criteria (MPV and LRU) vs. the memory size limit $m = \lceil r_l n^2 \rceil$ for tests *tsp225* and *lin318* (n is the size of the problem, r_l is memory size limit coefficient, e.g., $r_l = 0.1$ means that the size of the limited memory is equal to 10% of the size of the original pheromone memory in the ACS).

When the ant performs local pheromone update for such an edge, it results in the removal of the least important pheromone trail parameter from the memory to make place for the one associated with the traversed edge. With the increasing memory capacity the size of the memory replaced falls quickly. When it equals to only 10% of the original memory size, the number of parameters replaced in course of a single iteration falls below 1%. It suggests that memory capacity may be significantly reduced with little effect on the knowledge gathered by the ants, and thus on the algorithm performance. This observation is further confirmed by the Fig. 3, which shows how the ratio of the average pheromone trail value to the initial value τ_0 changes with the increasing memory capacity. The ratio seems to fall not linearly, but exponentially, what confirms that most of the pheromone accumulates on a small number of the pheromone trails, corresponding to edges in the TSP graph which belong to high quality solutions.

The second part of the experiments concerned how the limitation of the pheromone memory size affects the quality of the solutions. The computations were conducted for the six TSP instances mentioned and with the number of ants set to 25. The selective pheromone memory with the MPV and LRU selection criteria was used, with the memory size limit set to 2%, 10% and 50% of the original size, respectively. The algorithm with the selective pheromone memory was compared with the unmodified ACS algorithm (STD) and with the ACS in which all pheromone parameters remained constant during all the iterations (CONST). The latter comparison was made to see how important is the ant colony ability to "learn" through pheromone trails deposition. All the algorithms were run without a local search (No LS) and with the local search method (2-Opt LS) applied. From the Fig. 4 it can be seen that, surprisingly, the quality of the solutions found by all the algorithms except for CONST was similar. Significantly worse performance was observed for the CONST version (without ability

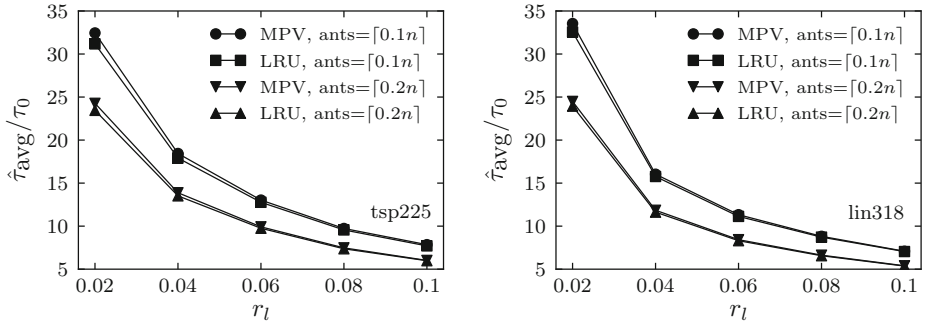


Fig. 3. Ratio of the avg. pheromone trail parameters value $\hat{\tau}_{\text{avg}}$ to the initial pheromone value τ_0 vs. the memory size limit $m = \lceil r_l n \rceil$ for tests *tsp225* and *lin318* (meaning of the symbols is the same as before)

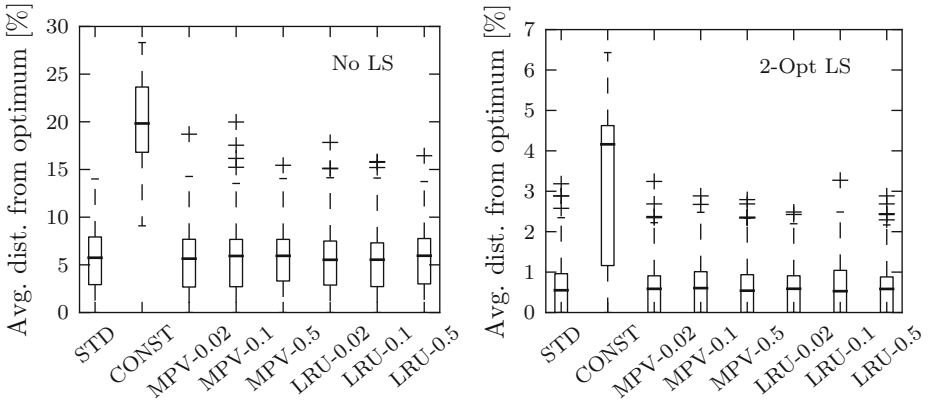


Fig. 4. Boxplots of the normalized average distance of solutions (for the 6 TSP tests investigated) from the optimum for the ACS with different pheromone memory strategies: CONST – constant pheromone values, STD – unmodified pheromone memory, MPV-*num* and LRU-*num* – selective pheromone memory with respective replacement criteria, *num* indicates the size of the memory, e.g., MPV-0.02 indicates that the size of the memory was set to 2% of the STD size

to gather knowledge), what confirms the fact that the pheromone memory plays essential role in the ACS.

It is worth noting that, when the local search was applied the quality of solutions was much higher, with the median distance to optimum below 1%. There was no difference between the standard ACS and the other versions (except CONST), what shows that limiting the size of the pheromone memory does not interfere with the local search performance. It is a very important characteristic because it is common to use the ACS in combination with the local search [5].

As can be seen from Fig. 5, the average solution distance from the optimum increased with the increasing size of the problem (number of cities). It is an

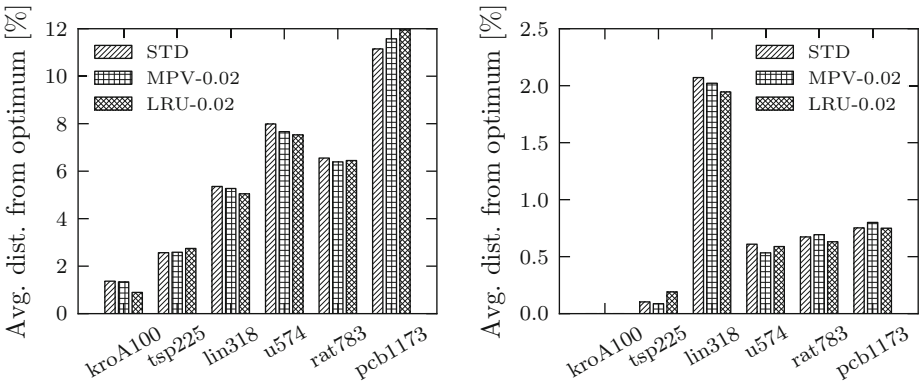


Fig. 5. Comparison of the average solution quality for the TSP instances investigated (meaning of the symbols as before). Left chart shows results for the algorithm without local search and the right chart shows results for the algorithm with 2-Opt local search method.

expected result because the size of the solution search space grows exponentially with the size of the problem. Application of the local search visibly improved the quality of solutions. The relatively bad quality of solutions was obtained for the *lin318* instance which consists of many dispersed clusters of cities. This is probably due to too small size of the nearest neighbor candidate list used.

Statistical analysis using the *non-parametric two-sample Wilcoxon rank-sum test* was conducted to test hypothesis H_0 that the quality of the results for the ACS with selective pheromone memory does not differ from the quality of the results for the standard ACS (STD). In all cases, the null hypothesis, H_0, was not rejected, with the confidence level of 99%, what confirms that there was no statistically significant difference between the results for the ACS with selective memory compared to the ACS with standard pheromone memory model.

5 Conclusions

Pheromone memory is essential to the performance of the ACS algorithm, but not all of the pheromone trail parameters are equally important. This follows from the fact that, the search process in the ACS is strongly focused on the areas of the problem search space which contain solutions of high quality. It poses a chance to reduce the size of the pheromone memory without impairing the quality of the results. We proposed a novel selective pheromone memory model, which stores only the selected subset of pheromone trail parameters, but does not require any modifications of the key ACS aspects. When a change to the value of the pheromone trail parameter which is not in the memory is made, it replaces the *least important* parameter currently stored in the memory. Two criteria for the selecting such parameters were proposed: MPV and LRU. For both criteria the quality of the results obtained was similar, but the latter is preferred because of the lower, constant time, computational overhead.

The proposed selective pheromone memory model allowed to achieve the same quality of the results as the ACS with the standard pheromone memory model, but with only 2% of its size.

5.1 Directions for Further Research

Our current implementation was not targeted for the maximum performance because various statistics were gathered during computations, therefore further experiments should focus on the performance. Obviously, selective pheromone memory model requires additional computations when accessing or updating pheromone trail values. Fortunately, the LRU version has only amortized constant time, $O(1)$, overhead, hence only slight increase in computation time should be expected.

Another important issue concerns how the proposed selective pheromone memory will affect the ACS performance for other difficult optimization problems. TSP, despite being a $\mathcal{NP}$-complete problem, has relatively "smooth" fitness landscape, therefore the algorithm with the proposed changes should be examined on problems with more constraints and more rugged fitness landscapes, such as Vehicle Routing Problem or Quadratic Assignment Problem [2,9].

References

1. Blum, C.: Ant colony optimization: Introduction and recent trends. Physics of Life Reviews 2(4), 353–373 (2005)
2. Czech, Z.J.: Statistical measures of a fitness landscape for the vehicle routing problem. In: IPDPS, pp. 1–8. IEEE (2008)
3. Dorigo, M., Blumb, C.: Ant colony optimization theory: A survey. Theoretical Computer Science 344, 243–278 (2005)
4. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
5. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Books, MIT Press (2004)
6. Matthews, D.C.: Improved Lower Limits for Pheromone Trails in Ant Colony Optimization. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN X. LNCS, vol. 5199, pp. 508–517. Springer, Heidelberg (2008)
7. Megiddo, N.: Outperforming lru with an adaptive replacement cache algorithm. IEEE Computer 37(4), 58–65 (2004)
8. Reinelt, G.: Tsplib95, http://www.iwr.uni-heidelberg.de/groups/comopt/-software/tsplib95/index.html
9. Stützle, T., Hoos, H.H.: Max–min ant system. Future Generation Computer Systems 16, 889–914 (2000)
10. Zlochin, M., Birattari, M., Meuleau, N., Dorigo, M.: Model-based search for combinatorial optimization: A critical survey. Annals of Operations Research 131(1), 373–395 (2004)

Ant Colony Optimization for the Pareto Front Approximation in Vehicle Navigation

Wojciech Bura and Mariusz Boryczka

Institute of Computer Science, University of Silesia
Wojciech.Bura@asseco.pl, mariusz.boryczka@us.edu.pl

Abstract. This paper presents an example of a multi-criteria optimization problem for vehicle navigation in the presence of multiple criteria and method of employing Ant Colony Optimization metaheuristic to solve it. The paper presents an approach based on the concept of Pareto optimality and approximation of set of non dominated solutions forming the so-called Pareto front.

Keywords: multi-agent system, ant system, multi-criteria problem, vehicle navigation.

1 Introduction

The process of solving complex combinatorial problems has practical applications in many fields of human activity. Most of actual optimization problems are multi-criteria optimization ones. There are a lot of methods of solving this kind of problems. One of them is searching for a set of non dominated solutions (Pareto efficient) that make up so-called Pareto front. For many real-life optimization problems the computational complexity of algorithms which provide the full set of non-dominated solutions is very high. This is the main reason for the popularity of methods giving an approximation of full set of such solutions. Subject of this paper is to use Ant Colony Optimization to approximate the Pareto front for the task of finding the optimal route for car navigation. This problem for the weighted objectives method was studied in works [2, 3] and in the work [4] the version using the CUDA technology was presented.

Section 2 introduces Ant Colony Optimization. In section 3 Ant Vehicle Navigation Algorithm (AVN) is described. Section 4 introduces the subject of the multi-criteria optimization with particular emphasis on the concept of Pareto efficiency and Pareto front approximation. Section 5 describes possible strategies to use ant colony optimization algorithms to solve multi-criteria optimization problems. Section 6 presents modifications to the AVN algorithm so that it is able to provide a set of solutions, approximating Pareto front as close as possible. Section 7 shows results of the experiments with proposed algorithm and finally section 8 summarizes the work.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 493–502, 2012.

2 Ant Colony Optimization

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 [5] and for the first time ant algorithms were presented in [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 for minimum cost feasible solutions for the problem being solved. It has a memory storing information about the path followed to date. 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 Ant Vehicle Navigation Algorithm

Communication networks and road maps are usually represented in computer systems as directed graphs. Thus many real-world optimization problems related to communication can be solved as optimization problems on graphs.

An example of such a problem is finding the shortest path between two points on a map, which boils down to finding the shortest path in a graph with weights. There are fast deterministic algorithms that solve this problem (e.g. Dijkstra's algorithm), with the computational complexity of the order of $O(n^2)$.

In fact, the shortest route is not necessarily regarded as the optimal solution. During the process of optimization solutions found are evaluated for several criteria of quality, such as travel time, number of intersections, traffic, etc. The objective is to determine a solution or set of solutions for which the individual objective functions reach the optimal value.

Finding the optimal path between two points on a map is a multi-criteria problem similar to finding the shortest path in the graph, known in literature as a MOSPP (Multiobjective Shortest Path Problem). The formal description of this problem can be found among others in the work [11].

This problem belongs to the class of NP-complete problems [8, 9] which encourages to use a heuristic methods that work quickly, but do not guarantee that the optimal solution is found. The Ant Colony Optimization algorithms are the example of such methods.

The original AVN algorithm [13–15] and its modified version NAVN [2, 3] search for the optimal path, which corresponds to the preferences set by the

user of the system. The parameter set consists of coefficients which control the importance level of distance, width of the route, number of intersections, traffic on the road, safety and quality of the proposed route.

The individual elements of the NAVN algorithm (Algorithm 1) are as follows.

Prepare Normalization. In this step for each element of the cost function (distance, width, traffic, risk, quality and intersections) normalization coefficient is calculated. Therefore user preference parameters have values in range $\langle 0, 1 \rangle$ and there is no need to adjust them to particular data on the map.

Initialize. Here, setting the algorithm parameters and pheromone trail initialization with the initial value is made.

ConstructProbability. Calculating the components of an array of probabilities used to select edges.

```
PrepareNormalization;
Initialize;
foreach loop do
    LocateAnts;
    foreach iteration do
        foreach ant do
            if ant is active then
                ConstructProbability;
                if ant has no move
                then MoveBack;
                else
                    SelectRoute;
                    UpdateTABUList;
    ValueAnts;
    AwardBestSolution;
    PunishLoserAnts;
    Modify Q;
ReturnBestOptimizedDirection;
```

Algorithm 1. NAVN

MoveBack. If ant is locked (there is no edge it can travel), it makes one step back. The edge used to go back is added to the list, which contains forbidden no more chosen by ant in current loop.

SelectRoute. Choosing the edge is performed the same way as in original algorithms but after selecting the edge an ant updates a pheromone trail according to the formula: $\tau_{ij}(new) = (1 - \rho) \cdot \tau_{ij}(old) + \rho \cdot \tau_0$, where: ρ is the trail evaporation coefficient ($0 \leq \rho \leq 1$).

ValueAnts. The solutions found by the ants are evaluated. If the solution is better than the best found so far then it is saved as the new best.

AwardBestSolution. The best solution found so far is rewarded through a global pheromone trail update rule.

PunishLoserAnts. In this step, the ants which failed to find a solution in a given number of iterations of the algorithm are punished. The pheromone trail on the edges belonging to their routes is decreased with the punishment coefficient.

Later in this paper in section 6 a new, Pareto version of AVN algorithm is proposed.

4 Multi-criteria Optimization

Optimization task can be commonly understood as the pursuit of the ideal state that meets certain criteria of evaluation. Solving optimization problems with the single evaluation criterion is single objective optimization. In real life optimization problems with the substantial amount of evaluation criteria appear more

often. For the substantial amount of the criteria very often there can exist a conflict between them and it can be very hard to satisfy them all which means that the desired solution is a kind of compromise between them. Formally, it is possible to formulate multicriteria optimization as follows:

Let $X = \{x_l\}, l = 1, 2, ..., N$ be a vector of independent decision variables. Let $F = \{f_i\}, i = 1, 2, ..., M$ be the set of criteria (functions) which are evaluated in the search for compromise solutions. Let there be restrictions on the solutions:

- inequality $G = \{g_k\}, k = 1, 2, ..., K$, where: $g_k(X) \leq 0$;
- equality $H = \{h_j\}, j = 1, 2, ..., J$, where: $h_j(X) = 0$

Multi-objective optimization is to achieve a solution, for which the condition is met (while maintaining restrictions):

$$\min F(X) = \{f_1(X), f_2(X), ..., f_l(X)\} \tag{1}$$

but if it is required to maximize a function, then you can enter a secondary criterion, according to the formula:

$$\min f_l(X) = -\max -f_l(X) \tag{2}$$

Two main approaches to solving multi-criteria problems are:

- reducing the problem to one criterion (substitute criterion);
- determine a set of nondominated solutions, so-called Pareto efficient, and seeking final solution among the elements of this collection by the method specified before the optimization process.

4.1 Substitute Criterion

Frequently used method for solving multicriteria optimization problems is the weighted objectives method. It consists in bringing to the single objective optimization of multi-criterion by introducing a replacement, which is a weighted sum of the individual criteria:

$$Z = \sum_{i=0}^{M} (w_i \cdot f_i(X)) \tag{3}$$

where: w_i is a coefficient of importance of i-th criterion which meets the following conditions:

$$0 \leq w_i \leq 1 \wedge \sum_{i=0}^{M} w_i = 1 \tag{4}$$

The objective function, obtained this way, is optimized using standard optimization methods with a single objective function. The main disadvantage of this method is the problem with the selection of appropriate values of weights for each criterion, and that the method usually returns only one solution.

4.2 Pareto Optimality

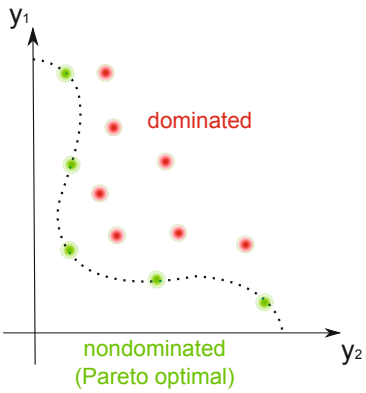


Fig. 1. Example of dominated and non dominated solutions

The French-Italian economist Vilfredo Pareto (1848-1923) in 1906 formulated the principle of multi-criteria optimization for economic issues, which later became known as Pareto optimization [12]. According to the principles of this method possible solutions of the optimization task are classified as dominated and non dominated solutions (Pareto optimal).

Definition 1 (Nondominated solution) . *Solution $x' \in X$, is nondominated (Pareto optimal) if there is no other solution $x \in X$, such that $F(x) \leq F(x')$ and at least one criterion is satisfied $F_l(x) < F_l(x')$.*

One Pareto optimal solution occurs only if all sub-optima can be found at the same point, it is also an optimal solution for the whole problem. In general, there are many Pareto optimal solutions (fig. 1), in the extreme case, every solution can be such a solution. A collection of non dominated solutions in the space of decision variables (space controls) corresponds to a set of criteria variables which is called the Pareto front (fig. 2).

After determining the set of non dominated solutions it is nesessery to specify the method of selection of the final solution. Examples of such methods are: dialogue method, metacriterion and hierarchy of objectives.

The techniques used in determining the set of non dominated solutions can be divided into exact and heuristic ones. Exact methods allow to find a set of optimal solutions in the Pareto sense for a given optimization problem, but are not suitable for most practical applications due to the very high computational cost algorithms. Heuristic methods do not guarantee to find a complete set of non dominated solutions, but they are able to significantly reduce the execution time of the algorithm.

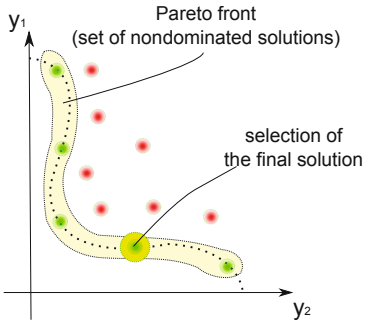


Fig. 2. Example of Pareto front

The examples of heuristic methods are: genetic algorithms, simulated annealing, tabu-search and ant colony algorithms.

4.3 The Pareto Front Aproximation

The task of the algorithm approximating the Pareto front is to determine the set of non dominated solutions that approximates a set of Pareto optimal solutions as

precisely as possible. Exact methods, determining the full set of Pareto optimal solutions, tend to have high computational complexity, of the order of $O(n^2)$. The high computational complexity of conventional algorithms encourages the use of heuristic methods, such as genetic algorithms (VEGA, SPEA). An example of such a method can also be ant colony optimization algorithms.

If it is possible to determine the complete set of Pareto optimal solutions with an exact method, this set can be used to evaluate the quality of approximation (fig. 3) as the distance between the two sets (Hausdorff metric). This volume should be minimized. The desired solution has a uniform distribution (a dispersed set and not concentrated within a small range of values of the criteria) and as wide as possible Pareto front, so that set of possible solutions for each criterion is as large as possible.

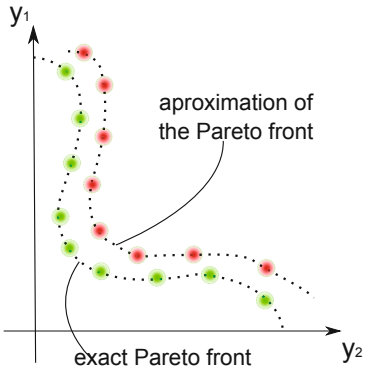


Fig. 3. Pareto front aproximation

5 Ant Colony Optimization and Multi-criteria Problems

Ant Colony Optimization algorithms have been successfully used to solve many optimization problems. The overall concept of the method is presented among others in [6]. Examples of the use of ACO for solving specific problems may include: Ant algorithm to optimize the supply chain [10], the algorithm to solve the bi-criteria shortest path problem [7], and algorithm to optimize production and maintenance scheduling [1]. Examples from the literature encouraged us to develop our own Ant Colony Optimization algorithms to solve complex multi-criteria optimization problems.

5.1 Multi-criteria Ant Colony Optimization in Pareto Sense

The general form of the algorithm is somewhat similar to the single objective optimization, but one should clearly distinguish new elements related to the methods of handling multiple criteria, including:

- one common pheromone table or separate tables for each criterion;
- separate heuristic information (the visibility) for each criterion or substitute criterion (e.g. weighted objectives method);
- global update of pheromone for non dominated solutions or for the best one in the series;
- one or more castes of ants - each caste responsible for the selected criterion.

In the case of one caste and separate heuristic information the ant when calculating the probability of edge selection randomly chooses a criterion. Algorithm 2) presents a skeleton of the ant algorithm which takes into account several criteria, which seeks a set of solutions approximating the Pareto front.

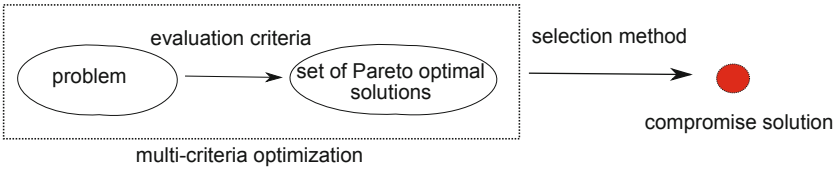


Fig. 4. The process of selecting the final solution

5.2 Selection of the Final Solution

Determining a set of Pareto optimal solutions is a first step of multi-criteria optimization. The aim is to appoint one, a compromise solution (fig. 4).

Selection of the final solution of the Pareto front set can be implemented in different ways, among others using: dialogue method, metacriterion or hierarchy of objectives.

```
InitializePheromone;
InitializeHeuristic;
InitializeParetoSet;
foreach loop do
    BuildSolutionsForAnts;
    UpdateParetoSet;
    UpdatePheromone;
ReturnParetoSet;
```

Algorithm 2. MultiACO

6 Application of the Pareto Optimality for Ant Vehicle Navigation Algorithm

A general Ant Colony Optimization algorithm, seeking non dominated solutions, which was presented in section 5.1, can be applied to solving many multi-criteria combinatorial problems. The following is proposed a modification of the original algorithm NAVN, so that it can be used to determine the set of nondominated solutions approximating the Pareto front in the way presented in section 4.3.

The proposed algorithm returns the set of solutions. In comparison with the original version of algorithm elements associated with the set of Pareto optimal solutions were introduced. There are different variants of realization of the *ConstructProbability* procedure depending on the structure of the pheromone array. Procedure *AwardBestSolution* may take into consideration solutions from the set of Pareto optimal solutions or from the best solution in the cycle.

A pseudocode of modified NAVN algorithm is presented as Algorithm 3. It is able to determine a set of nondominated solutions.

```
PrepareNormalization;
Initialize;
InitializeParetoSet;
foreach loop do
    LocateAnts;
    foreach iteration do
        foreach ant do
            if ant is active then
                ConstructProbability;
                if ant has no move
                then MoveBack;
                else
                    SelectRoute;
                    UpdateTABUList;
    ValueAnts;
    UpdateParetoSet;
    AwardBestSolution;
    PunishLoserAnts;
    Modify Q;
ReturnParetoSet;
```

Algorithm 3. MultiNAVN

The key change is the introduction of elements related to search of the set of Pareto optimal solutions:

UpdateParetoSet. The procedure is responsible for comparing the solutions created by the ant with the solutions found so far and recognized as Pareto optimal. If the new solution is nondominated by any solution found so far it is added to the list of nondominated solutions. Solutions dominated by the new solution are removed from this list.

ReturnParetoSet. The algorithm returns a list of solutions identified as Pareto optimal.

It is planned to test the different versions of a modified NAVN algorithm which is able to find a set of non dominated solutions — the Pareto front approximation:

- single caste with randomly selected criteria and a separate array of pheromones;
- muli-caste system with constant and variable population;
- hybrid version (Dijkstra ⇒ AVN ⇒ local search);
- various options for the selection of the final solution from a designated set of Pareto optimal solutions.

7 Computational Experiments

In order to verify the described approach, computational experiments were performed with the modified NAVN algorithm. As in the previous experiments [2–4], data were collected from the Open Street Map (OSM) in the area of the city of Katowice.

In terms of size the datasets are comparable to those which were used in the experiments with the algorithm NAVN [2, 3]. Cartographic data downloaded from the OSM has been supplemented with information on accidents and collisions resulting from Sewik '99 system operated by the Polish Police.

Table 1. Algorithm's parameters

Parameter	Value
α	1
β	3
pv	0.95
bv	0.15
ρ	0.2
Q	0.95
φ	0.9
τ_0	0.000025
ant count	16
loop count	50

The map consisted of more than 25,000 nodes and more than 50,000 edges. Criteria taken into account were: distance, width (number of lanes) and travel safety.

The detailed data for the experiments were (other parameters of the NAVN 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.

Experiments were performed with the algorithm using internally substitue criterion (section 4.1), and therefore it was necessary to determine the user's preferences before running the experiments.

NAVN algorithm parameter values are presented in tab. 1. These values were chosen empirically during preliminary trials, to provide optimum performance in solving a problem that is the subject of the work.

In one experiment, the algorithm has found five different non dominated solutions, which are presented in tab. 2. In the subsequent rows of the table are included the cost function values for each criterion: distance, width (number of lanes), and safety. The purpose of the NAVN algorithm is to determine solutions with the lowest cost for each criterion (minimization).

Subsequently, from a designated set of non dominated solutions a compromise solution should be selected, in accordance with the principles described in subsection 5.2.

If we decide to use the method of selection of the final solution based on a hierarchy of criteria and adopt the following order of criteria: distance, width, safety then the solution number 4 is chosen, because it has the lowest cost of the criterion of distance. If we feel that we care most about the safety then solution number 3 is selected with the lowest value of the cost for the safety criterion, although it is a solution with the greatest cost associated with the distance.

Table 2. Example of the set of non dominated solutions found by the NAVN

Solution number	Distance	Width	Safety
1	21040	1746	217
2	19353	1396	227
3	27025	1857	175
4	19219	1331	230
5	25738	2144	212

8 Conclusion

In the paper, we presented different approaches to solving multi-criteria optimization problems using Ant Colony Optimization algorithms. Ant Colony Optimization algorithms are a promising method of solving this type of complex optimization problems.

It is planned to examine the different versions of the AVN algorithm modified so that it determine the set of solutions forming the Pareto front approximation, including multi-cast system.

References

1. Berrichi, A., et al.: Bi-Objective Ant Colony Optimization approach to optimize production and maintenance scheduling. Computers & Operations Research 37, 1584–1596 (2010)
2. Bura, W., Boryczka, M.: Ant colony system in ambulance navigation. Journal of Medical Informatics & Technologies 15, 115–124 (2010)

3. Bura, W., Boryczka, M.: Parallel version of the NAVN system. In: Systemy Wspomagania Decyzji. Uniwersytet Śląski w Katowicach (2010) (in Polish)
4. Bura, W., Boryczka, M.: The Parallel Ant Vehicle Navigation System with CUDA Technology. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS (LNAI), vol. 6923, pp. 505–514. Springer, Heidelberg (2011)
5. 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)
6. Dorigo, M., Maniezzo, V., Colorni, A.: Positive Feedback as a Search Strategy. Technical Report No. 91-016. Milano, Politechnico di Milano (1991)
7. Ghoseiria, K., Nadjaria, B.: An ant colony optimization algorithm for the bi-objective shortest path problem. Applied Soft Computing 10, 1237–1246 (2010)
8. Garey, J.: Computers and Intractability: A Guide to the Theory of NPcompleteness. Freeman, San Francisco (1979)
9. Hansen, P.: Bicriterion path problems. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making: Theory and Application. Lectures Notes in Economics and Mathematical Systems, vol. 177, pp. 109–127. Springer, Heidelberg (1980)
10. Moncayo-Martinez, L., Zhang, D.: Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design. Int. J. Production Economics 131, 407–420 (2011)
11. Pangilinan, J.M.A., Janssens, G.K.: Evolutionary Algorithms for the Multiobjective Shortest Path Problem. World Academy of Science, Engineering and Technology 25 (2007)
12. Pareto, V.: Course d'Economie Politique. F. Rouge, Lausanne (1896)
13. 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)
14. Salehinejad, H., Pouladi, F., Talebi, S.: A new route selection system: Multi-parameter ant algorithm based vehicle navigation approach. In: CIMCA 2008, IAWTIC 2008 and ISE 2008 (2008)
15. 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)

A Hybrid Discrete Particle Swarm Optimization with Pheromone for Dynamic Traveling Salesman Problem

Urszula Boryczka[1] and Łukasz Strąk[2]

[1] University of Silesia, Intitute of Computer Science,
Będzińska 39, 41-205 Sosnowiec, Poland
urszula.boryczka@us.edu.pl
[2] University of Silesia, Intitute of Computer Science,
Będzińska 39, 41-205 Sosnowiec, Poland
lukasz.strak@gmail.com

Abstract. This paper introduces a new Discrete Particle Swarm Optimization algorithm for solving Dynamic Traveling Salesman Problem (DTSP). An experimental environment is stochastic and dynamic, based on Benchmark Generator was prepared for testing DTSP solvers. Changeability requires quick adaptation ability from the algorithm. The introduced technique presents a set of advantages that fulfill this requirement. The proposed solution is based on the virtual pheromone first applied in Ant Colony Optimization. The pheromone serves as a communication topology and information about the landscape of global discrete space. Experimental results demonstrate the effectiveness and efficiency of the algorithm.

Keywords: dynamic traveling salesman problem, pheromone, particle swarm optimization.

1 Introduction

There has been a growing interest in studying evolutionary algorithms in dynamic environments in recent years due to its importance in real applications [4]. A problem where input data are changeable depending on time is called Dynamic Optimization Problem (DOP). The purpose of the optimization for DOPs is to continuously track and adapt the changes through time and to find quickly the currently best solution [15]. Metaheuristics that proved their effectiveness for static problems are being modified by different adaptation strategies for use in dynamic environments. Especially Evolutionary Algorithms with Gene Pool [14] and Ant Colony Optimization [6,22] provide good results.

Particle Swarm Optimization is a technique based on swarm population created by Russell Eberhart and James Kennedy in 1995 [12]. This technique is inspired by the social behavior of a bird flocking or fish schooling. The algorithm was created primarily to optimize the function of a continuous field of space exploration. The PSO algorithms quickly became popular due to the fact

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 503–512, 2012.

that it has small number of parameters and it is easy to implement [12]. The growing interest has caused that this technique was adapted to solve different problems: static and dynamic [25,19].

In contrast with the classic Traveling Salesman Problem in the DTSP distances between the cities which are subjected to changes and cause that it increases the computational complexity of the algorithm. The PSO algorithm in the problem of dynamic TSP is unexplored, but there are many publications of the PSO in a dynamic environment. The PSO in the dynamic environment was presented by [3,24]. First it was adapted for the Moving Peaks Benchmark, secondly for automatically tracks various changes in a dynamic system. The PSO with the virtual pheromone had first appeared in [11]. Kalivarapu [11] modified the parameters so that the pheromone was treated as additional information on the global landscape of optimized functions. Senthilkumar et al. [20] combined PSO with ACO for combinatorial problem. The PSO in this algorithm generates the initial solution which optimizes the ACO.

In this paper we introduce the algorithm for solving Dynamic Traveling Salesman Problem based on digital pheromone trail known in the ACO. The main idea is to use various proven techniques rather than creating a new one from scratch. Our research focus also on it as a new topology for communication between particles. We also propose optimal values for the parameters of the most successful tests.

This paper is structured as follows: in section 2 we present the basic concepts: Dynamic Traveling Salesman Problem and miscellaneous implementations of Discrete Particle Swarm Optimization. Section 3 presents ours DPSO algorithm proposals. Research results are shown in Section 4. Section 5 contains summary and conclusions.

2 Background

The Traveling Salesman Problem is a classic discrete combinatorial optimization problem. The objective of TSP is to find a shortest Hamilton cycles in an undirected graph $G = (V, E)$, where V is a vertex set (cities) and E is the edge set (routes). This problem plays an important role in the discrete optimization. This follows from the fact that many problems can be transformed to the problem of TSP by adjusting the encoding problem [1]. An interesting example is the DNA Computing [18] where cities represent gene sequences. It has been proved that the TSP is the $NP - complete$ problem [7]. For this reason, exact algorithms can not be used practically.

There are several approaches to discrete optimization for the TSP with the PSO. The first attempt was coded the TSP city as ROV (Ranked Order Value) [2,17] by converting the continuous position values into a tour. Kennedy and Eberhart [13,25] defined first discrete binary version of PSO. All particles were coded as a binary string. The predefined velocity was interpreted as the probability of a bit state transition from zero to one and from one to zero. In a velocity formula we can use a sigmoid function restricting value to 0 or 1. Hu,

Shi and Eberhart [10,25], introduced a modified PSO to deal with permutation problems. Particles were defined as permutations of a group of unique values. The velocity is a vector of probability of exchange two elements of permutation. Pairwise exchanging operations are also called 2-swap or 2-exchange.

Most of studies were carried out for the small size of the TSP problem (14 cities) [9,23]. Shi [21] tried to expand problem size, presenting a new PSO with two different local search operators for generalized TSP. Both algorithms defined subtraction in terms of sequences of 2-swap operations as defined in the path-relinking velocity operator. He based his research on an algorithm proposed by Wang [23]. Zhong, Zhang and Chen [16,8] proposed a new algorithm for TSP in which the particle was not a permutation of numbers but a set of edges. It used the parameter c_3 named by the authors a mutation factor, which allowed control (balance) between exploration and exploration in discrete space. Most current PSO adaptations to the problem of static TSP were developed and compared in [8]. The study [16,8] has shown that this is the best adaptation of the PSO to the problem of static TSP.

The dynamic TSP is expressed through changes in both the number of vertices and a cost matrix. Younes et al. proposed a tool that unifies these changes - the Benchmark Generator (BGM) [26].The test began with the optimization of the static data from the TSP library. Then began the first phase - vertex or matrix modification cost depending on the BMG mode; which is described later in this article. Then, the modifications were withdrawn, so that at the end of the second phase, test data were exactly the same as in the beginning. Then it compares a priori result (optimal) with the last found by the algorithm.

3 DPSO with Pheromone

The common feature of the PSO implementations is the general symbolic equation describing steps of the algorithm, which is inspired by the social behavior of bird flocking or fish schooling (equation 1) [12].

$$v_i^{k+1} = w \cdot v_i^k + c_1 rand() \cdot (pBest - x_i^k) + c_2 rand() \cdot (gBest - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

where i denote the number of particles, k - iteration number, $rand()$ is a random value from $[0, 1]$. Variables $pBest$ and $gBest$ denote particle personal best position and global best position respectively. Variable w is called an inertia weight and decide how much the pre-velocity will affected the new position. c_1 and c_2 are cognitive and social parameters for scaling $pBest$ and $gBest$ respectively. At the beginning, the algorithm initializes the distributed population and randomly distributed particles. At the second step this algorithm iteratively assigns velocity (which means assign the search direction) using equation 1 and it assigns position from previously calculated velocity using equation 2. After each iteration, variables $pbest$ and $gbest$ are updated. Depending on the problem being solved particle, position and velocity can be a vector, a set or a number.

Most PSO algorithms for the TSP problem are based on permutations of numbers [8]. The algorithm proposed by Zhong et al. [16] is based on the concept of a set of edges. Its requires the introduction of many new concepts to adapt PSO and to operate with the edges. Most important concepts are described later in this article. The concepts are also described in [16].

Edge can be represented by $A(x, y)$, where A denotes the probability of choosing edge, x and y are the endpoints. A is constrained to a number between 0 and 1 and $A(x, y) = A(y, x)$. Edge (2,3) with probability 0.4 takes the form $0.4(2, 3)$. When the algorithm is on velocity updating phase, a random number R between 0 and 1 is given, and if $R \leq X$ this edge is chosen. Velocity v is a set of elements as $A(x, y)$ like $\{A(a, b), B(a, c), C(b, d)\}$. To reduce the size of the edges set, vertex may be found at most 4 times. Therefore, the velocity can create sub tours (1-2-3-1) and some vertices can be omitted. The same edge but with a different probability is also acceptable. A substation between two positions is defined as a set of edges which exist in x_1 but not in x_2. To adapt this result to edge representation form, the probability 1 is added to these edges and makes its uniform as velocity. A multiplication between a real number and a set of edges is defined like multiplication number and probability. Sum of two velocities v_1,v_2 is a set of $v_1 \cup v_2$, but vertex can't exist more than 4 times in edges. The new parameter c_3 is introduced. Equation 1 and 2 take a new form given by equations 3 and 4. New position is calculated in three steps. There are also more restrictive rules that obtained edge must fulfill like added edge can not create sub-cycle, or vertex can not occur more than 2 times.

$$v_i^{k+1} = c_2 rand() \cdot (gBest - x_i^k) + c_1 rand() \cdot (pBest - x_i^k) + w \cdot v_i^k \tag{3}$$

$$x_i^{k+1} = v_i^{k+1} \oplus c_3 rand() \cdot x_i^k \tag{4}$$

In the first step we choose edges depending on their corresponding probability in v_i to construct temporary position x_i'. If edge is chosen but construct an incorrect tour it will discard this edge. In second step, the algorithm selects edges from x_i according to the probability to complete the x_i'. Like before, an illegal edge is discarded. Third, if the first two steps not make a whole Hamiltonian cycle, it will add the absent with nearest principle heuristic. First and second steps have more restricted rules because third step must create feasible tour. Changing the order of performed operations in equation 3 is caused due to increasing importance of $gBest$ rather than $pBest$.

If parameters c_1, c_2, c_3 are more than 1, the probability of some edges after multiplied a random number may be larger than 1, then probability would be limited to 1. The full description of the algorithm may be found in [16].

The above algorithm is designed for searching of the solution space from the random position. In dynamic environment the DPSO algorithm starts from some partial solutions, which are the solutions obtained from previous calculations. To adapt this algorithm and to exploit the partial solutions, virtual pheromone was used as a form of attraction to the edges, which are frequently visited in best positions. So the algorithm starts running with some information about searching space. This use of the virtual pheromone works like backbones concept.

The backbone of a TSP instance consists of all edges, which are contained in each optimum tour of the instance [5]. However in our case, the edges of the backbones are not constant but depend on the frequency of visits. In addition, the global pheromone stores information about the best solutions better than $gBest$ because the information is more complete. Edge diversity depends on pheromone distribution. When just some edges will accumulate much pheromone value, the diversity is decreasing. The pheromone value depends on a frequency visit matrix and function that calculates value stored in matrix to the real amount of pheromone. The matrix is presented in the formula 5. If the edge (a, b) is chosen to be part of the current position, the matrix is modified (cell located by pair a, b is increased by 1). In the following example the edge (a, c) is visited 2 times, edge (a, d) only once. The matrix is also symmetric, so the edge $(a, b) = (b, a)$.

$$fvm = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} - & 0 & 2 & 1 & 0 \\ 0 & - & 1 & 1 & 0 \\ 2 & 1 & - & 1 & 1 \\ 1 & 1 & 1 & - & 0 \\ 0 & 0 & 1 & 0 & - \end{pmatrix} \end{array} \tag{5}$$

The pheromone amount is calculated according to the formula 6 that control an importance of pheromone. The amount of pheromone controls diversity and a number of allowed edges. The value range and formula 6 are the subjects of the research presented in section 4.

$$amount(a, b) = \begin{cases} 0 & fvm(a, b) = 0 \\ 4^{-1} \cdot log(1 + x) & fvm(a, b) > 0 \end{cases} \tag{6}$$

Set-based algorithm with pheromone is combined by changing all edges probability by the parameter calculated according to the formula 6. Before selection in line 6 (Algorithm 1) the pheromone amount is added to edge probability. If the pheromone amount is equal 0, then no change to original will be done. The pseudo code presents algorithm 1, N denotes number of cities.

4 Experimental Results

The set-based algorithm proposed by Zhong et al. originally has been adapted to the static environment. The ability of the algorithm to quickly find the shortest path requires large number of viewed edges. In the DTSP such large variety is not desirable because it increases the algorithm working time and makes the transition to exploration. Our first experiments focused on reducing the diversity of the edges and force the algorithm to focus mainly on attractors - edges with large amounts of pheromone. In other words, it was necessary to force the algorithm to exploitation instead of exploration. However, the algorithm shouldn't lose the ability to find the shorter path. The results were also used to develop the formula 6. At this stage of research the pheromone was applied on edges obtained

Algorithm 1. DPSO with pheromone outline

```
 1: procedure DPSO
 2:     calibrate pheromone using formula 6
 3:     for k = 1 → N · 10 do
 4:         for i = 1 → 30 do                                  ▷ DPSO phase
 5:             calculate velocity
 6:             do selection from v_i and make result as x'_i
 7:             select edges from x_i to complete x'_i with probability c3 rand()
 8:             add missing edges using nearest principle
 9:             exchange pBest, gBest by new x_i if necessary
10:         end for
11:     end for
12: end procedure
13: procedure DPSO Solver
14:     random initial population for n                        ▷ Initial phase
15:     do full initial running of DPSO
16:     overlap pheromone, set pBest and gBest
17:     repeat                                                 ▷ 1st phase
18:         modify one edge in the best tour
19:         do DPSO with 0.25 iterations
20:         promote pheromone for best positions
21:     until half the required instances
22:     repeat                                                 ▷ 2nd phase
23:         remove latest modification
24:         modify one edge in the best tour
25:         make DPSO with 0.25 iterations
26:         promote pheromone for best positions
27:     until half the required instances
28:     return best tours for all instances
29: end procedure
```

from $gBest - x_i$, $pBest - x_i$ and $w \cdot v_i$ edges with the same pheromone amount. Figure 1 presents diversity of edges with different values of the pheromone. Configuration was set to $c_1 = 1.5$, $c_2 = 2$, $c_3 = 2$, $w = 0.6$, i - iterations $= 1000$ (n cities $\cdot 10$), n_s - swarm size $= 30$, data set is kroA100. This configuration was originally proposed by Zhong et al. in [16].

Our observations has shown that the algorithm is very dynamic and difficult to control. As expected, the greatest impact on diversity have the parameters c_1, c_2, c_3, and w. The higher value means more edges selected from $gBest$, $pBest$ and v_i. The diversity is decreased only for the probability nearest 1. Due to the large algorithm's dynamic, is not necessary to use the spread of the pheromone. The algorithm itself exploit the nearest edges. Experimental results have shown that pheromone should be value between 0 and 1. This follows from the fact that the new edges are only obtained in the last phase of the algorithm (Algorithm 1 line 8). If edges from previous phase creates Hamiltonian cycles, new edges will be omitted. Pheromone value near 1 is large enough to fix a small random value. Even if random value is equal 0.1 edge will be selected to the next position.

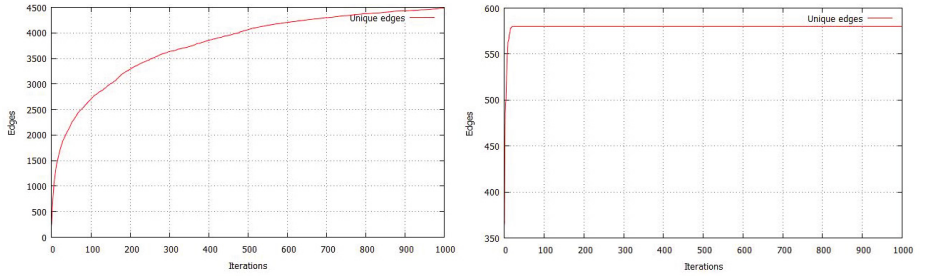


Fig. 1. Uniques edges with different amount of pheromone from left 0 - neutral, 0.9

However, the value of pheromone close to 1 reduces the ability to browse the new potential solutions. Algorithm shouldn't lost the ability to find short paths, which means value near 1 must be selected only in extreme cases. Figure 2 presents the performance of original algorithm and our algorithm with pheromone. The configuration is the same as in previous experiments.

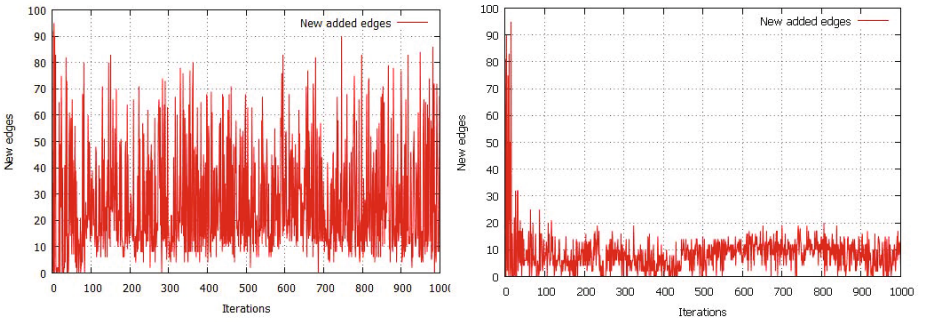


Fig. 2. New edges obtained from nearest principle. Left picture is from original algorithm and right from our modification.

Up to 30 iterations of the algorithm has a similar number of new edges. This is due to a large difference between the current position and $gBest$. The new position is medley with edges from $gBest$, $pBest$ and previous velocity v_i. Previous position is not able to cover the set of the new position. To complete the tour new edges are obtained from the nearest principle algorithm.

Table 1 presents results of Dynamic Traveling Salesman Problem benchmark. The benchmark is generated by BGM with Edge Change Mode described in section 2. The pseudo code, which was used for testing is presents in Algorithm 1. After changes of city length, set-based DPSO with pheromone iterations are reduced to full algorithm iterations divided by 4. Changes in city distances are equal to 10. It means in five instances distance matrix is modified and in five instances this modifications have been withdrawn. Table 1 presents result repeated

Table 1. Experiment results for DTSP with pheromone

Changes		Problem				
		Eil51	Berlin52	Eil76	KroA100	KroA200
Before	Avg. time [s]	1,34	1,39	5,05	15,1	102,67
	Avg. distance	482	8894	655	27205	53341
	Best	447	8023	585	23713	47266
1	Avg. time [s]	0,27	0,27	0,8	1,98	7,11
	Avg. distance	468	8664	637	26130	48334
	Best	443	7766	578	23334	41193
2	Avg. time [s]	0,22	0,23	0,62	1,59	6,46
	Avg. distance	463	8548	627	25604	45923
	Best	427	7762	578	21951	41193
3	Avg. time [s]	0,22	0,22	0,63	1,44	5,57
	Avg. distance	457	8454	621	25421	45339
	Best	426	7626	571	21774	41193
4	Avg. time [s]	0,2	0,21	0,63	1,47	5,53
	Avg. distance	453	8417	615	25300	45076
	Best	426	7626	568	21774	40806
5	Avg. time [s]	0,2	0,2	0,6	1,43	5,3
	Avg. distance	451	8387	611	25184	44821
	Best	426	7626	564	21774	39682
6	Avg. time [s]	0,19	0,2	0,58	1,35	5,1
	Avg. distance	450	8383	610	25082	44757
	Best	426	7626	560	21774	39682
7	Avg. time [s]	0,18	0,2	0,54	1,34	5,2
	Avg. distance	447	8369	609	25056	44757
	Best	426	7626	560	21597	39682
8	Avg. time [s]	0,18	0,19	0,54	1,44	5,81
	Avg. distance	446	8358	609	24916	44161
	Best	426	7626	560	21597	39682
9	Avg. time [s]	0,18	0,2	0,62	1,45	5,41
	Avg. distance	447	8328	604	24611	44061
	Best	426	7626	560	21597	39682
10	Avg. time [s]	0,18	0,2	0,61	1,42	5,19
	Avg. distance	446	8266	598	24453	43428
	Best	426	7626	560	21597	39682
Optimal		426	7542	538	21282	29368

50 times. The exception to this rule is KroA200 where it was 10 repeats. This is due to a very long execution. The DPSO configuration remains unchanged. Experiments were run on computer with Intel i7 processor 3.2 GHz and 12 GB of RAM memory. Operating systems is Microsoft Windows Server 2008 R2. All tests were run on single core.

The most important results can be occur in the initial (before changes) and the last iteration of the algorithm. All benchmark iterations between them may contains Hamiltonian cycle, shorter than optimal. It is due to randomly edge

length changed by the BGM. From the standpoint of TSPLIB correct cycles are received before the changes and in the last iteration. Each iteration is independent of the others. This means that the best position obtained in one iteration will be reflected with changes in the distance matrix in the context of the second iteration. As can be seen in Table 1 the running time is decreasing when pheromone amount is increased. It was one of the objectives. Due to the fact that the pheromone increases the probability of choosing the edge from the previous position to the new position, the addition of new edges will be blocked. At the same time the algorithm through formula 6 not lost the ability to search for short edges because the latter result is better than the initial score.

5 Conclusions

In this paper, a set-based DPSO with pheromone to the DTSP was proposed. Although the algorithm is very dynamic, we were able to adapt it for its intended purpose. It should also be noted that the algorithm is fast enough to solve Dynamic Traveling Salesman Problem which are demonstrated in experiments. The aim was to limit the search to those solutions which are attractors - edges with large amount of pheromone. As demonstrated by the study the objective has been completed. The future work will focus on improving the nearest principle heuristic. In our solution this is the only place where we do not control the algorithm by using pheromone.

References

1. Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The traveling salesman problem: A computational study. Princeton University (2006)
2. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA Journal of Computing 6, 154–160 (1994)
3. Blackwell, T.: Particle Swarm Optimization in Dynamic Environments. In: Yang, S., Ong, Y.-S., Jin, Y. (eds.) Evolutionary Computation in Dynamic and Uncertain Environments. SCI, vol. 51, pp. 29–49. Springer, Heidelberg (2007)
4. Branke, J.: Evolutionary approaches to dynamic environments. In: GECCO Workshop on Evolutionary Algorithms for Dynamics Optimization Problems (2001)
5. Dong, C., Ernst, C., Jäger, G., Richter, D., Molitor, P.: Effective heuristics for large euclidean tsp instances based on pseudo backbones (2009)
6. Eyckelhof, C.J., Snoek, M.: Ant Systems for a Dynamic TSP. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) ANTS 2002. LNCS, vol. 2463, p. 88. Springer, Heidelberg (2002)
7. Garey, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman (1979)
8. Goldbarg, E.F.G., Goldbarg, M.C., de Souza, G.R.: Particle swarm optimization algorithm for the traveling salesman problem (2008)
9. Hendtlass, T.: Preserving Diversity in Particle Swarm Optimisation. In: Chung, P.W.H., Hinde, C.J., Ali, M. (eds.) IEA/AIE 2003. LNCS, vol. 2718, pp. 31–40. Springer, Heidelberg (2003)

10. Hu, X., Eberhart, R.C., Shi, Y.: Swarm intelligence for permutation optimization: Case study of n-queens problem (2003)
11. Kalivarapu, V., Foo, J.-L., Winer, E.: Improving solution characteristics of particle swarm optimization using digital pheromones. In: Structural and Multidisciplinary Optimization (2009)
12. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
13. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, Computational Cybernetics and Simulation. IEEE (1997)
14. Li, C., Yang, M., Kang, L.: A New Approach to Solving Dynamic Traveling Salesman Problems. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 236–243. Springer, Heidelberg (2006)
15. Li, W.: A Parallel Multi-start Search Algorithm for Dynamic Traveling Salesman Problem. In: Pardalos, P.M., Rebennack, S. (eds.) SEA 2011. LNCS, vol. 6630, pp. 65–75. Springer, Heidelberg (2011)
16. Zhong, W.L., Zhang, J., Chen, W.N.: A novel set-based particle swarm optimization method for discrete optimization problems. In: Evolutionary Computation, CEC 2007, vol. 14, pp. 3283–3287. IEEE (1997)
17. Liu, B., Wang, L., Jin, Y.H., Huang, D.X.: An Effective PSO-Based Memetic Algorithm for TSP. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCIS, vol. 345, pp. 1151–1156. Springer, Heidelberg (2006)
18. Azimi, P., Daneshvar, P.: An Efficient Heuristic Algorithm for the Traveling Salesman Problem. In: Dangelmaier, W., Blecken, A., Delius, R., Klöpfer, S. (eds.) IHNS 2010. LNBIP, vol. 46, pp. 384–395. Springer, Heidelberg (2010)
19. Schutte, J.F., Groenwold, A.A.: A study of global optimization using particle swarms. Journal of Global Optimization 31, 93–108 (2005)
20. Senthilkumar, K.M., Selladurai, V., Raja, K., Thirunavukkarasu, V.: A hybrid algorithm based on pso and aco approach for solving combinatorial fuzzy unrelated parallel machine scheduling problem. European Journal of Scientific Research (2011)
21. Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C.L., Wang, Q.X.: Particle swarm optimization-based algorithms for tsp and generalized tsp. Inf. Process. Lett. (2007)
22. Silva, C.A., Runkler, T.A.: Ant colony optimization for dynamic traveling salesman problems. In: ARCS Workshops (2004)
23. Wang, K.-P., Huang, L., Zhou, C.-G., Pang, W.: Particle swarm optimization for traveling salesman problem. In: International Conference on Machine Learning and Cybernetics, vol. 3. IEEE (2003)
24. Xiaohui, H., Eberhart, R.C.: Adaptive particle swarm optimisation: detection and response to dynamic systems. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002 (2002)
25. Xiaohui, H., Shi, Y., Eberhart, R.C.: Recent advances in particle swarm (2004)
26. Younes, A., Basir, O., Calamai, P.: A benchmark generator for dynamic optimization. In: Digest of the Proceedings of the Wseas Conferences (2003)

A Modified Shuffled Frog Leaping Algorithm with Genetic Mutation for Combinatorial Optimization

Kaushik Kumar Bhattacharjee and Sarada Prasad Sarmah

Deptt. of Industrial Engineering and Management
Indian Institute of Technology Kharagpur
India, Kharagpur 721302
bhattacharjee.kaushik@gmail.com,
sp_sarmah@yahoo.com

Abstract. In this work, we propose modified versions of shuffled frog leaping algorithm (SFLA) to solve multiple knapsack problems (MKP). The proposed algorithm includes two important operations: repair operator and genetic mutation with a small probability. The former is utilizing the pseudo-utility to repair infeasible solutions, and the later can effectively prevent the algorithm from trapping into the local optimal solution. Computational experiments with a large set of instances show that the proposed algorithm can be an efficient alternative for solving 0/1 multidimensional knapsack problem.

Keywords: Genetic mutation, metaheuristics, multidimensional knapsack problem, repair operator, shuffled frog leaping algorithm.

1 Introduction

The 0/1 multidimensional (multiple constrained) knapsack problem (01MKP) is a well-studied, strongly NP-hard combinatorial optimization problem occurring in many different applications, such as the capital budgeting problem, allocating processors and databases in a distributed computer system, project selection, cargo loading, and cutting stock problems. The most common formulation of 01MKP is as follows:

$$\begin{aligned} &Maximize \;\; f(x_1, x_2, ..., x_n) = \sum_{j=1}^{n} c_j x_j \\ &Subject \;\; to \;\; \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, 2, ..., m \\ &\qquad x_j \in \{0, 1\}, \quad j = 1, 2, .., n \\ &\qquad c_j > 0, \;\; a_{ij} \geq 0, \;\; b_i > 0. \end{aligned} \tag{1}$$

The objective function $f(x_1, x_2, ..., x_n)$ should be maximized subject to the constraints. For 01MKP problems the variable x_j can take only two values 0 and

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 513–522, 2012.

1. Here in a 01MKP, it is necessary that a_{ij} is non negative. This necessary condition paves a way for better heuristics to obtain near optimal solutions.

Exact and heuristic algorithms have been developed for the 01MKP , like many NP-hard combinatorial optimization problems. Existing exact algorithms are essentially based on branch and bound method [1], dynamic programming [2], systematic approach [3] and 01MKP relaxation techniques [4] such as Lagrangian, surrogate and composite relaxations. Due to their exponential time complexity, exact algorithms are limited to small size instances. On the other hand heuristic and metaheuristic algorithms are designed to produce near-optimal solutions for larger problem instances. The greedy method is used as the first heuristic approach to solve 01MKP. Metaheuristics are also used to solve 01MKP, like tabu search [5], simulated annealing [6], genetic algorithm [7], ant colony optimization [8], and particle swarm optimization (PSO) [9].

SFLA is one of the most recent developed metaheuristic which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food [10]. SFLA, originally developed by Eusuff and Lansey in 2003, can be used to solve many complex optimization problems, which are nonlinear, non-differentiable, and multi-modal [11]. The most distinguished benefit of SFLA is its fast convergence speed [12]. The SFLA combines the benefits of both the genetic-based memetic algorithm (MA) and the social behavior-based PSO algorithm [13].

In this present study we test the convergence property of SFLA on different types of 01MKP instances. Two quantitative measures are considered in order to assess the performance of the algorithms concerning the deviation from the optimal value and number of iterations to reach the best solution, respectively. Further, computational experiments with a set of large-scale instances are also tested. The work is organized as follows. Section 2 introduces key concepts of shuffled frog leaping algorithm. Different modified versions of SFLA is the subject of Section 3. Experiments and computational results are presented in Section 4. The last section offers concluding remarks and direction of future work.

2 Shuffled Frog Leaping Algorithm

The SFLA is a combination of deterministic and random approaches. The deterministic strategy allows the algorithm to use response surface information effectively to guide the heuristic search as in PSO. The random elements ensure the flexibility and robustness of the search pattern.The SFLA, in essence, combines the benefits of the genetic-based memetic algorithms and the social behavior-based PSO algorithms. An initial population of P frogs is created randomly. For S-dimensional problems (S variables), a frog i is represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iS})$. Afterwards, the frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m memeplexes, each containing n frogs (i.e. $P = m \times n$). In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the m-th memeplex, and frog $m + 1$ goes back to the first memeplex,

etc. Within each memeplex, the frogs with the best and the worst fitness are identified as X_b and X_w, respectively. Also, the frog with the global best fitness is identified as X_g. Then, a process similar to PSO is applied to improve only the frog with the worst fitness in each cycle. Accordingly, the position of the frog with the worst fitness is adjusted as follows:

$$D_i = Rand() \times (X_b - X_w), \tag{2}$$

where D_i is the change in i-th frog position and new position is given by:

$$\begin{aligned} X_w(new) &= X_w + D_i, \\ -D_{max} &\leq D_i \leq D_{max}; \end{aligned} \tag{3}$$

where $Rand()$ is a random number($Rand() \sim U(0,1)$); and D_{max} is the maximum allowed change in a frog's position. If this process produces a better solution, it replaces the worst frog. Otherwise, the calculations in Eqs. 2 and 3 are repeated but with respect to the global best frog (i.e. X_g replaces X_b). If no improvement becomes possible in this case, then a new solution is randomly generated to replace that frog. The calculations then continue for a specific number of iterations [14].

3 Modified Shuffled Frog Leaping Algorithms for 01MKP

01MKP has a special structure as shown by Eq. 1, which can not be handled by SFLA. For this reason original SFLA is modified and the modified binary shuffled frog leaping algorithm (MBSFLA) is discussed in this section in full details.

3.1 Process for Binary Variables

01MKP problem is an integer programming problem. There are only two possible values for the decision variable x_j $(j = 1, 2, ..., n)$. For this reason, in this work we have used three different kinds of discretization techniques to solve it.

1. Method 1: the worst frog X_w of each memeplex is replaced according to

$$\begin{aligned} t &= X_w + D; \\ X_w(new) &= \begin{cases} 0 & \text{if } t \leq 0, \\ round(t) & \text{if } 0 < t < 1, \\ 1 & \text{if } t \geq 1. \end{cases} \end{aligned} \tag{4}$$

2. Method 2: D is transformed to the interval $[0, 1]$ by using sigmoid function. The worst frog is replaced according to Eq. 5.
3. Method 3: the updating formula for the worst frog is given by Eq. 6. The parameter α is called static probability.

$$t = 1/(1+\exp(-D)); \quad u \sim U(0,1) \qquad X_w(new) = \begin{cases} 0 & \text{if } t \le u, \\ 1 & \text{if } t > u. \end{cases} \tag{5}$$

$$t = 1/(1+\exp(-D)); \qquad X_w(new) = \begin{cases} 0 & \text{if } t \le \alpha, \\ X_w & \text{if } \alpha < t \le \frac{1}{2}(1+\alpha), \\ 1 & \text{if } t \ge \frac{1}{2}(1+\alpha). \end{cases} \tag{6}$$

3.2 Pseudo-utility and Repair Operator

When the binary string violates the constraint of the given problem, then repair algorithm is employed to make infeasible solutions to feasible one. Infeasible solutions will be repaired by using a repair operator, which is a kind of greedy heuristic based on the pseudo-utility.

At the initialization step, MBSFLA sorts and renumbers variables according to the decreasing order of their pseudo-utility, which were calculated by the surrogate duality approach introduced by Pirkul [15].

The surrogate relaxation problem of the 01MKP can be defined as:

$$\begin{aligned} &Maximize \sum_{j=1}^{n} c_j x_j \\ &Subject\ \ to \sum_{j=1}^{n} (\sum_{i=1}^{m} \omega_i a_{ij}) x_j \le \sum_{i=1}^{m} \omega_i b_i, \quad i = 1, ..., m \\ &\qquad x_j \in \{0,1\}, \quad j = 1, ..., n \end{aligned} \tag{7}$$

where $\omega = \{\omega_1, \omega_2, ..., \omega_m\}$ is a set of surrogate multipliers (or weights) of some positive real numbers. The pseudo-utility ratio for each variable, based on the surrogate constraint coefficient, is defined as

$$u_j = \frac{c_j}{\sum_{i=1}^{m} \omega_i a_{ij}}. \tag{8}$$

The repair operator is inspired from the idea of Chu and Beasley [16], which consists of two phases. The pseudo-code of the repair operator is given in Algorithm a of Figure 1.

3.3 Genetic Mutation

Discrete shuffled frog leaping algorithm sometimes trapped in the local optimal point. To avoid this situation we utilize the genetic mutation to avoid premature convergence. After shuffling the memeplexes we use the genetic mutation operator to modify the population as $\bar{x}_{ij} = |x_{ij} - 1|$ with a small mutation probability p_m.

The pseudocode for MBSFLA is given in Algorithm b of Figure 1. Accordingly, the main parameters of MBSFLA are: number of frogs P; number of memeplexes m; number of generation for each memeplex before shuffling n; number of shuffling iterations it; maximum number of iterations $iMax$; static probability α (for Method 2); and genetic mutation probability p_m.

(a) a) Pseudocode for Repair Operator

```
Let: R_i = accumulated resources of constraint i in x
Initialize R_i = Σ_{j=1}^{n} a_ij x_j, ∀i ∈ I
{DROP Phase}
for j = n to 1 do
    if (x_j = 1) and (R_i > b_i, for any i ∈ I) then
        x_j := 0
        R_i := R_i − a_ij, ∀i ∈ I
    end if
end for
{ADD Phase}
for j = 1 to n do
    if (x_j = 0) and (R_i + a_ij ≤ b_i, ∀i ∈ I) then
        x_j := 1
        R_i := R_i + a_ij, ∀i ∈ I
    end if
end for
```

(b) b) Pseudocode for Main Procedure

```
Generate random population of P solutions (frogs)
for each individual i ∈ P do
    Calculate fitness(i)
end for
Sort the population P in descending order of their fitness
Divide P into m memeplexes
for each memeplex do
    Determine the best and worst frogs
    Improve the worst frog position using Eq. 4, 5 or 6
    Repeat for a specific number of iterations
end for
Combine the evolved memeplexes
Mutate
Repair the population P
Sort the population P in descending order of their fitness
if termination = true then
    Return best solution
end if
```

Fig. 1. Pseudocode for MBSFLA Procedure

4 Experiments and Computational Results

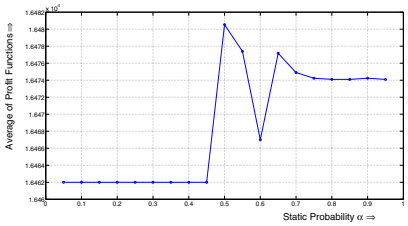
In this section, the performance of shuffled frog leaping algorithm is extensively investigated by a large number of experimental studies. For performance analysis we select total 56 benchmark problem instances of 01MKP from OR-Library ([17]) corresponding to small and medium class; items are ranging from 6 to 105; and number of knapsack is 2 to 30. All computational experiments are conducted with MATLAB 7.6.0 in Intel(R) Core(TM)2 Duo CPU E7400 @2.80 GHz with 4GB of RAM.

4.1 Effect of Static Probability

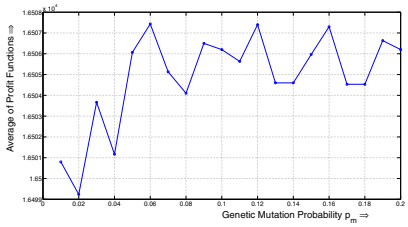
Among first seven test problems of small class, f_7 is much more difficult to solve, because the capacity constraints are of the type of average knapsack capacity. So the initial parameter setting is done with this problem instance. The population size is $P = 200$ along with $m = 10$ memeplexes, number of iterations within each memeplex $it = 10$ and maximum number of iterations is considered as $iMax = 100$. The performance criteria of binary shuffled frog leaping algorithm (BSFLA) for different static probabilities (α values) for the function f_7 is given in Fig. a of Figure 2. Average profit for 30 individual runs corresponding to different α values is plotted corresponding to the range of α values ($[0.05, 0.95]$) in the figure. The objective function value deteriorates as one moves far from the mid point. Therefore we may choose $\alpha = 0.5$.

4.2 Comparison among Three Binary Shuffled Frog Leaping Algorithms

For first seven test problems best solution and worst solution among 30 independent runs are reported in Table 1 for three binary shuffled frog leaping algorithms



(a) Effect of Static Probability



(b) Effect of Genetic Mutation Probability

Fig. 2. Effect of Static Probability (α) and Genetic Mutation Probability (p_m)

Table 1. Comparison between Three Binary Shuffled Frog Leaping Algorithms

f	op	Criteria					f	op	Criteria				
		best	*worst*	*average*	*std*	*ATT*			*best*	*worst*	*average*	*std*	*ATT*
		3800	3800	3800	0	0.01			12400	12400	12400	0	0.25
f_1	3800	3800	3800	3800	0	0	f_5	12400	12400	12165	12357.84	57.48	1.39
		3800	3800	3800	0	0			12400	12360	12383	16.65	1
		8706.1	8706.1	8706.1	0	0.03			10605	10570	10585.84	8.7	1.73
f_2	8706.1	8706.1	8706.1	8706.1	0	0.03	f_6	10618	10604	10547	10565.74	19.7	1.79
		8706.1	8706.1	8706.1	0	0.06			10604	10547	10553.14	14.35	1.77
		4015	4005	4013.67	3.46	0.3			16537	16421	16485.47	27.22	2.18
f_3	4015	4015	4005	4014.34	2.54	0.18	f_7	16537	16501	16358	16442	42.46	2.27
		4015	4005	4011.67	4.8	0.44			16511	16436	16460.07	30.28	2.24
		6120	6110	6119	3.06	0.26							
f_4	6120	6120	6110	6119.67	1.83	0.11							
		6120	6100	6118	4.85	0.28							

as discussed in Section 3.1. Also average, median and standard deviation (*std*) for all the solutions are given here along with average total time (ATT) to solve the problem. Maximum number of iterations is considered as $iMax = 100$, and population size is $P = 200$ along with $m = 10$ memeplexes.

From Table 1, it is clear that, Method 1 is much more effective to find out best solution with respect to others. In most of the cases Method 1 performs better with respect to average, median, standard deviation and ATT (except for the function f_3 and f_4, Method 2 performs better with respect to average and standard deviation (*std*)).

4.3 Effect of Genetic Mutation

For the standard test problem f_7 of small size, population size of MBSFLA is set to $P = 200$ along with $m = 10$ memeplexes, and the number of iterations in each memeplex is set to $it = 10$. Total 30 independent experiments are carried out in each case. The performance of MBSFLA with respect to different mutation

Table 2. Comparison between BSFLA and MBSFLA

f	$iMax = 50$					$iMax = 100$					$iMax = 150$				
	best	*worst*	*average*	*std*	*ATT*	*best*	*worst*	*average*	*std*	*ATT*	*best*	*worst*	*average*	*std*	*ATT*
f_1	3800	3800	3800	0	0.01	3800	3800	3800	0	0.01	3800	3800	3800	0	0.01
	3800	3800	3800	0	0.01	3800	3800	3800	0	0.01	3800	3800	3800	0	0.01
f_2	8706.1	8706.1	8706.1	0	0.02	8706.1	8706.1	8706.1	0	0.03	8706.1	8706.1	8706.1	0	0.03
	8706.1	8706.1	8706.1	0	0.03	8706.1	8706.1	8706.1	0	0.02	8706.1	8706.1	8706.1	0	0.02
f_3	4015	4005	4014	3.06	0.11	4015	4005	4014	3.06	0.16	4015	4005	4014	3.06	0.25
	4015	4015	4015	0	0.03	4015	4015	4015	0	0.03	4015	4015	4015	0	0.03
f_4	6120	6110	6119	3.06	0.11	6120	6110	6119.67	1.83	0.16	6120	6110	6119.67	1.83	0.18
	6120	6120	6120	0	0.03	6120	6120	6120	0	0.03	6120	6120	6120	0	0.03
f_5	12400	12370	12398.34	6.48	0.31	12400	12400	12400	0	0.3	12400	12400	12400	0	0.3
	12400	12400	12400	0	0.06	12400	12400	12400	0	0.06	12400	12400	12400	0	0.07
f_6	10618	10547	10580.37	17.92	0.79	10618	10552	10586.8	11.84	1.55	10618	10570	10585.74	11.06	2.33
	10618	10584	10592.5	9.55	0.82	10618	10584	10596.57	8.61	1.64	10618	10584	10604.17	7.65	2.26
f_7	16520	16371	16473.14	35.61	0.99	16537	16356	16485.24	35.87	1.92	16537	16436	16489.87	20.35	2.88
	16518	16499	16500.47	3.9	1.01	16537	16499	16506.27	9.61	1.96	16537	16499	16505.37	8.83	2.97

probabilities is shown in Fig. b of Figure 2. Range of p_m is taken as $[0.01, 0.2]$, as beyond this range the performance of MBSFLA degraded gradually.

The best results were obtained at more than one point ($p_m = 0.06$, 0.12 and 0.16). As the complexity of the multiple knapsack problem increases with its size and the adaptivity of p_m to problems with higher dimension sizes may decrease more or less within this region. So for finding dynamic balance between problem size and p_m value, we fixed the value of $p_m = 2/n$ for large problem instances, where n is the number of items. And for small and medium size problems we fixed p_m value at 0.06.

4.4 Comparison among BSFLA and MBSFLA

We consider the same seven standard test problems to compare the performance of BSFLA and MBSFLA. In the first case, we present the comparison between these two with respect to objective function value, and in the second case we only consider the maximum number of iterations. The parameter setting for these two algorithms is given below.

We consider population size $P = 200$, number of memeplexes $m = 10$ and number of iterations within each memeplex $it = 10$ for both the cases. For MBSFLA mutation probability is $p_m = 0.06$. Three values of maximum number

Table 3. Comparison between BSFLA and MBSFLA with respect to Iteration Number

f	BSFLA					MBSFLA				
	best	*worst*	*average*	*std*	*ATT*	*best*	*worst*	*average*	*std*	*ATT*
f_1	1	1	1	0	0.01	1	1	1	0	0.01
f_2	1	11	2.44	2.78	0.03	1	23	2.17	4.2	0.02
f_3	1	500	74.8	152.26	0.84	1	7	2.1	1.25	0.03
f_4	1	500	24.97	91.99	0.34	1	30	2.97	5.21	0.05
f_5	1	74	19.17	19.19	0.3	1	6	3.74	1.29	0.06
f_6	500	500	500	0	7.77	22	500	438.77	159.04	7.06
f_7	500	500	500	0	9.4	74	500	440.47	119.04	8.62

Table 4. Solutions of Medium Size 01MKP Instances

f	n	m	op	Solution quality					No of iterations				ATT
				best	*worst*	*average*	*std*	*dev*	*best*	*worst*	*average*	*std*	
f_1	28	2	141278	141278	141278	141278	0	0	1	4	2.04	0.97	0.1
f_2	60	30	7772	7772	7772	7772	0	0	2	112	35.34	24.64	3.58
f_3	60	30	8722	8722	8722	8722	0	0	3	15	7.64	2.52	0.7
f_4	28	2	141278	141278	141278	141278	0	0	1	8	2.24	1.39	0.11
f_5	28	2	130883	130883	130883	130883	0	0	1	22	6.47	4.51	0.32
f_6	28	2	95677	95677	95677	95677	0	0	2	58	12.1	12.53	0.69
f_7	28	2	119337	119337	119337	119337	0	0	1	25	7.3	8.38	0.34
f_8	28	2	98796	98796	98796	98796	0	0	1	2	1.77	0.44	0.1
f_9	28	2	130623	130623	130623	130623	0	0	1	56	14.6	14.17	0.73
f_{10}	105	2	1095445	1095445	1095382	1095384.1	11.51	0	38	150	146.27	20.45	11.79
f_{11}	105	2	624319	624319	623612	624295.44	129.08	0	1	150	44.47	40.19	6.61
f_{12}	30	5	4554	4554	4554	4554	0	0	2	6	3.2	1.13	0.18
f_{13}	30	5	4536	4536	4536	4536	0	0	2	16	7.3	4.19	0.41
f_{14}	30	5	4115	4115	4115	4115	0	0	1	8	3.07	1.51	0.18
f_{15}	30	5	4561	4561	4561	4561	0	0	1	2	1.9	0.31	0.11
f_{16}	30	5	4514	4514	4514	4514	0	0	1	2	1.9	0.31	0.11
f_{17}	40	5	5557	5557	5557	5557	0	0	2	7	3.64	1.83	0.25
f_{18}	40	5	5567	5567	5567	5567	0	0	2	6	3.64	1.33	0.25
f_{19}	40	5	5605	5605	5605	5605	0	0	2	32	14.17	8.98	0.89
f_{20}	40	5	5246	5246	5246	5246	0	0	2	5	2.37	0.72	0.17
f_{21}	50	5	6339	6339	6339	6339	0	0	4	38	15.97	8.21	1.28
f_{22}	50	5	5643	5643	5643	5643	0	0	2	4	2.2	0.49	0.2
f_{23}	50	5	6339	6339	6339	6339	0	0	2	29	11.57	6.79	0.95
f_{24}	50	5	6159	6159	6159	6159	0	0	2	7	3.37	1.52	0.29
f_{25}	60	5	6954	6954	6954	6954	0	0	2	8	4.54	1.95	0.41
f_{26}	60	5	7486	7486	7486	7486	0	0	2	20	11.37	4.62	0.99
f_{27}	60	5	7289	7289	7288	7288.84	0.38	0	5	150	70.5	50.16	5.92
f_{28}	60	5	8633	8633	8633	8633	0	0	3	12	7.44	2.2	0.54
f_{29}	70	5	9580	9580	9580	9580	0	0	15	117	41.64	23.51	3.63
f_{30}	70	5	7698	7698	7698	7698	0	0	2	19	6.54	4.3	0.68
f_{31}	70	5	9450	9450	9450	9450	0	0	8	96	41.04	25.04	3.76
f_{32}	70	5	9074	9074	9074	9074	0	0	5	56	21.6	12.51	2.01
f_{33}	80	5	8947	8947	8929	8939.8	8.97	0	18	150	99.1	53.49	10.58
f_{34}	80	5	8344	8344	8344	8344	0	0	7	108	32.44	24.59	3.58
f_{35}	80	5	10220	10220	10198	10209.57	6.8	0	52	150	145.5	18.91	13.61
f_{36}	80	5	9939	9939	9923	9932.4	6.33	0	38	150	143.87	22.54	14.32
f_{37}	90	5	9584	9584	9552	9578.94	8.38	0	7	150	109.17	47.68	13.11
f_{38}	90	5	9819	9819	9819	9819	0	0	8	144	33.67	34.15	4.02
f_{39}	90	5	9492	9492	9492	9492	0	0	4	99	33.6	27.03	4.05
f_{40}	90	5	9410	9410	9410	9410	0	0	4	82	22.67	15.12	2.8
f_{41}	90	5	11191	11191	11191	11191	0	0	12	142	41.84	27.12	4.39
f_{42}	27	4	3090	3090	3076	3082.2	6.95	0	2	150	97.07	65.05	4.15
f_{43}	34	4	3186	3186	3156	3174.1	9.93	0	12	150	123.7	44.47	5.92
f_{44}	29	2	95168	95168	95168	95168	0	0	1	1	1	0	0.05
f_{45}	20	10	2139	2139	2139	2139	0	0	1	4	2.17	0.75	0.09
f_{46}	40	30	776	776	776	776	0	0	2	17	6.07	4.08	0.52
f_{47}	37	30	1035	1035	1035	1035	0	0	1	2	1.97	0.19	0.15
f_{48}	28	4	3418	3418	3404	3408.3	6.47	0	4	150	123.3	47.65	5.3
f_{49}	35	4	3186	3186	3153	3170.94	10.35	0	38	150	134.57	32.08	6.88

of iterations ($iMax$) 50, 100 and 150 respectively, are considered. Total 30 independent runs are made and corresponding results are given in Table 2.

In Table 2, for each problem instance the first row represents the solution correspond to BSFLA, and the second row produced by MBSFLA. As we can see in Table 2, MBSFLA performs better than BSFLA, and it can easily find the optimal solution for all the cases (except for f_7 when $iMax = 50$). MBSFLA is also performing well with respect to other performance criteria like average and standard deviation (std).

In the second case, we consider the maximum number of iterations $iMax = 500$. The parameter settings of the two algorithms are same as the previous case. From Table 3 we find out that BSFLA fails to find best solutions for f_6 and f_7, and it successfully solve function f_3 only for 26 cases out of 30 and 29 out of 30 cases for function f_4. Whereas MBSFLA needs $iMax$ on an average 3 (for function f_3)and 4 (for function f_4). The fewer number of iterations shows that MBSFLA has higher efficiency than BSFLA on finding best solutions for 01 multidimensional knapsack problems.

4.5 01MKP Instances with Medium Dimension Size

In this case the parameter settings of the MBSFLA is as follows: population size $P = 400$, number of memeplexes $m = 20$, number of iterations within each memeplex $it = 10$, mutation probability $p_m = 0.06$ and maximum number of iterations $iMax = 500$. Total 30 independent runs are considered and the corresponding results for all test problems are reported in Table 4.

Table 4 presents the best solution, worst solution, average solution and standard deviation with respect to solution quality and number of iterations. Also time of execution (ATT) and deviation from the best known solution are given for 49 test instances. Deviation from the optimum values are calculated as $D = \frac{(OptimumValue - BestSolution)}{OptimumValue} \times 100\%$. MBSFLA finds optimum solutions for all the test cases. On an average only 35 iterations required to solve this type of problem instances. MBSFLA finds best solutions for every independent runs in 38 cases and the average execution time is much lower (within 3 seconds).

5 Conclusion

In this work a relatively new member of memetic based meta-heuristics called shuffled frog leaping algorithm is explained. Most of the studies on SFLA is carried out in last five years and researchers mainly concentrated on continuous optimization and TSP problems in the literature. In this study, the performance of MBSFLA has been extensively investigated by using a large number of problem instances. The experimental results show that MBSFLA has demonstrated strong convergence and stability for 01MKP and it has strong ability to prevent premature convergence by utilizing genetic mutation. The proposed algorithm thus provides a new method for 01MKP, and it may find the required optima in cases when the problem is too complicated and complex. It may also be used for

solving multi-objective knapsack problems and other combinatorial optimization problems like generalized assignment problems, set covering problems etc. and is scheduled as the future work.

References

1. Osrio, M., Glover, F., Hammer, P.: Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. Technical Report HCES-08-00, Hearing Center for Enterprise Science (2000)
2. Nemhauser, G., Ullmann, Z.: Discrete dynamic programming and capital allocation. Management Science 15(9), 494–505 (1969)
3. Balas, E.: An additive algorithm for solving linear programs with zero-one variables. Operations Research 13(4), 517–546 (1965)
4. Crama, Y., Mazzola, J.: On the strength of relaxations of multidimensional knapsack problems. INFOR 32(4), 219–225 (1994)
5. Arntzen, H., Hvattum, L.M., Lokketangen, A.: Adaptive memory search for multidemand multidimensional knapsack problems. Computers and Operations Research 33(9), 2508–2525 (2006)
6. Egeblad, J., Pisinger, D.: Heuristic approaches for the two- and three-dimensional knapsack packing problem. Computers and Operations Research 36(4), 1026–1049 (2009)
7. Hua, Z., Huang, F.: A variable-grouping based genetic algorithm for large-scale integer programming. Information Sciences 176(19), 2869–2885 (2006)
8. Ke, L., Feng, Z., Ren, Z., Wei, X.: An ant colony optimization approach for the multidimensional knapsack problem. Journal of Heuristics 16(1), 65–83 (2010)
9. Zhao, Q., Zhang, X., Xiao, R.: Particle swarm optimization algorithm for partner selection in virtual enterprise. Progress in Natural Science 18(11), 1445–1452 (2008)
10. Eusuff, M.M., Lansey, K., Pasha, F.: Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. Engineering Optimization 38(2), 129–154 (2006)
11. Zhang, X., Hu, X., Cui, G., Wang, Y., Niu, Y.: An improved shuffled frog leaping algorithm with cognitive behavior. In: Proc. 7th World Congr. Intelligent Control and Automation 2008 (2008)
12. Elbeltagi, E., Hegazy, T., Grierson, D.: Comparison among five evolutionary based optimization algorithms. Adv. Eng. Informat. 19(1), 43–53 (2005)
13. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE Conf. Neural Networks, vol. 4, pp. 1942–1948 (1995)
14. Eusuff, M., Lansey, K.: Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resource Plan Management 129, 210–225 (2003)
15. Pirkul, H.: A heuristic solution procedure for the multiconstraint zero-one knapsack problem. Naval Research Logistics 34, 161–172 (1987)
16. Chu, P.C., Beasley, J.E.: A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics 4(1), 63–86 (1998)
17. Beasley, J.E.: Or-library: Distributing test problems by electronic mail. Journal of Operational Research Society 41(11), 1069–1072 (1990)

Integrating Curriculum and Instruction System Based on Objective Weak Tie Approach

Chia-Ling Hsu[1], Hsuan-Pu Chang[2], Ren-Her Wang[3], and Shiu-huang Su Hsu[4]

[1] Center for Teacher Education, Tamkang University
151, Ying-chuan Road, Tamsui, New Taipei City, 25137, Taiwan
[2] Department of Information and Library Science, Tamkang University
151, Ying-chuan Road, Tamsui, New Taipei City, 25137, Taiwan
[3] Department of Banking and Finance, Tamkang University
151, Ying-chuan Road, Tamsui, New Taipei City, 25137, Taiwan
[4] Academic affairs, Tamkang University
151, Ying-chuan Road, Tamsui, New Taipei City, 25137, Taiwan
{clhsu,musicbubu,138230,0115582}@mail.tku.edu.tw

Abstract. In order to improving the quality of higher education, sever strategies or models are proposed from universities. Evaluation system is one of these strategies for checking the performance and keeping the high quality. However, it is easy to describe the evaluation system but it will be hard to implement in higher education environment. When there is a goal to be achieved, evaluation system is used. By the time past, more paper works and more tables are required to fill in for different evaluation systems. Due to these problems, the purpose of this paper is to propose an objective weak tie system to integrate four different curriculum and instruction systems. The benefits of this weak tie system will not increase the loading for teachers and will increase the efficiency for administration by remaining the original system and using the objective to make linking with each other as an integrating curriculum and instruction system.

Keywords: curriculum and instruction, weak tie, integrating system, higher education, knowledge management.

1 Introduction

Due to the low birth rate, a dramatic change in higher education causes a big competition in Taiwan. Every university is making efforts on improving the quality of higher education. In order to improving the quality of higher education, sever strategies or models are proposed from universities. Evaluation system is one of these strategies for checking the performance and keeping the high quality. As a result, the higher education become more business-oriented than liberal art oriented [1][2]. The performance and outcome turn out to be the main leadership. In addition, it is easy to describe the usage of evaluation system

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 523–531, 2012.

but it will be hard to implement in higher education environment. When there is a goal to be achieved, evaluation system is used. By the time past, more paper works and more tables are required to fill in for different evaluation systems. This phenomenon leads to a lot of working loads for teachers and administrators because of many paper works need to be done. The performance evaluation system causes faculties over work in higher education.

The technology will help solve the problem mention above. Therefore, how to build up a system for higher education keeping high quality is the main issue. For one purpose, one computer system is developed. For another purpose, another computer system is established. It raises a question which is how to the use these computer system efficiently for user among these computer systems. Whether to integrate these computer systems or to build up a new computer system will be considered. If building up a computer system for higher education, It will be a complicated to design for many purposes since the curriculum and instruction emphasizes in many different aspects. Hence, integrating different computer will be another choice. However, it will be harder than to build a new system for the program system point of view. How to link these different purposes computer system will be a big problem.

This paper will base on evaluating the university existing system and then find the way to link these systems. The purpose of this study will propose an objective weak tie approach to integrate the existing computer system. Since Mark Granovetter (1973) brought up an idea of strong tie and weak tie, many scholars paid attention in weak tie for innovation, opportunity, chance or improving[3][4][5]. In this study, the link will be defined as "weak tie" since the integration system needs these linkage but every existing computer system may work independently without the linkages. So, this paper will find the links as weak tie to integrate systems for higher education quality purpose.

Many decision making researches applied Keeney's (1992) value focused Thanking [6]. Although the weak tie approach was used to integrate the existing computer system in this study, the decision making skill was also important in differentiating between the noises and chances.

1.1 Research Purpose and Significance

The purpose of this study is to integrate the curriculum and instruction system using objective weak tie approach. In detail, the purposes below:

- Find the elements in each existing computer system in order to possible weak ties.
- Use value focus think technology to investigate the alternative objective in curriculum and instruction system.
- Integrate a curriculum and instruction system

This study proposes an objective weak tie approach to integrate the existing computer systems into a new and virtual curriculum and instruction system. In practical point of view, the significance of this study is to save the money, human power and time. The benefit of the integrating curriculum and instruction system

will provide the linkages among the existing computing system; however, the existing computer systems will remain independently by themselves without the linkage. Therefore, it will be easy to maintain the individual computer system. In theoretical point of view, the significance of the weak tie theory will extend with value focus thinking and will be apply to higher education environment.

2 Relative Literature

This study is based on the objective weak tie approach which was come up from different theories. The innovation and weak tie theories were the foundation. Then, in order to finding the weak ties among the existing, the value focus thinking model was applied. A trial was implemented in higher education for curriculum and instruction. So, the literature review related to innovation, weak tie, value focus thinking, curriculum and instruction.

2.1 Innovation and Weak Tie

The technologies of finding weak tie are variety. Wang, Hong, Sung, and Hsu applied the KeyGraph technology to find the rare and important element [7]. The results indicated that although the statistics data showed no significant difference, the KeyGraph technology provided more information. Hsu and other educators also applied the KeyGraph technology in education setting. The results pointed out that the learners' scenario map would tell more information than the traditional statistics results. Huang, Tsai, & Hsu also applied the KeyGraph technology to exploring the learners' thinking [8]. Tsai, Huang, Hong, Wang, Sung, and Hsu [9] used KeyGraph technology and tried to find the chances in instructional activity.

The studies above were concerned in finding weak tie in text data. Some weak tie research was related to knowledge creation and transfer. Levin and Cross (2004) found that weak ties provide access to no redundant information [10].

2.2 Value Focus Thinking

Thinking about value was to decide what you want and then to figure out how you can get it. So, it was a nature way for alternative-focus thinking. Hsu, Hong, Wang, Chiu, and Chang (2009) used the VFT model in instruction design to improving the teaching quality [11].

2.3 Curriculum and Instruction

In the domain of curriculum and instruction are emphasizes not only designing an effective curriculum or instruction but also evaluation the students learning outcomes. No matter what approach the research used would lead to a reflection of instruction [12][13][14][15][16]. Marsh and Willis (1999) indicated that different school implemented the curriculum and instruction differently but it only a few models in

total, the objective model, Countenance model, illuminative model and educational connoisseurship model [17].

3 Research Method

This study was to analyze the existing computer systems and to fine the weak ties among them. Then, an objective weak tie approach was applied to integrate a curriculum and instruction system. Finally, the integrated and innovated virtual curriculum and instruction system would be appeared. Research method of this study was followed by the process of Ground Theory. Although the ground theory was usually used in qualitative research, the method was universally for any research as long as the research purpose was to understand phenomenon, cause effects or relationships. Using the process of ground theory, this study would provide an integrated and innovated virtual curriculum and instruction system by the objective weak tie approach. Therefore, the research procedure bellows in figure 1.

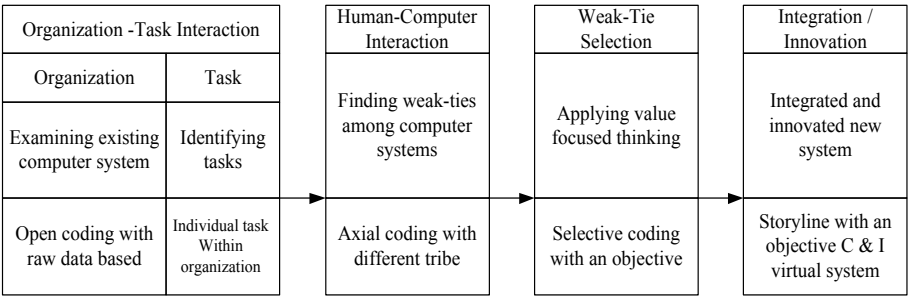


Fig. 1. Research Procedure

3.1 Objective Weak Tie Model

The objective weak tie model is begun with some existing computers which were established by different purposes. Each computer system contains its own attributes. Thus, it is important to find all the elements in each computer. In addition, the strong ties are easily to define by the administration department. The visualization will be in figure 2.

Each of the computer system will belong to one administration department with one special aim. Therefore, in practically, there are some strong ties among these systems. The strong ties usually will link to one administration department. Different department develops its own computer system when there is one goal to achieve once at a time. Then, there will be more than one computer systems under one administration department. However, in the functional aspect, the linkage may be mission or on the surface. This situation causes the computer systems independently although all the computer systems are managed by one administration department.

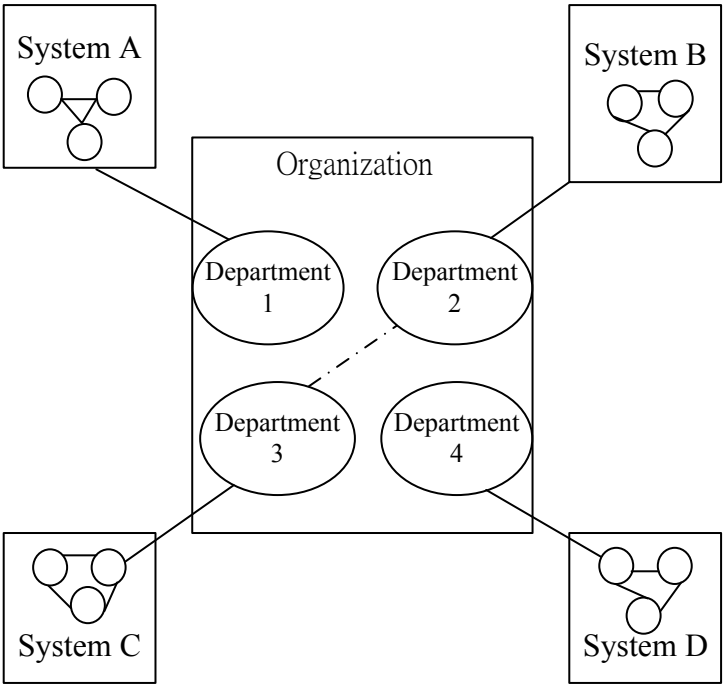


Fig. 2. The strong ties in one organization

In order to achieving the particular goal, the weak ties are needed to be established. Nevertheless, it will be hard to look for the relationships among the individual physical existing system. In other words, these computer systems are independently. Usually, when the particular aim to achieve, staffs would make up forms to the persons or departments which will relate to this particular aim and would ask these people to check the individual computer system to fill out the form and then to hand in to the organization. As a result, it costs a lot of time consume and paper works. Therefore, this study would provide an objective weak ties approach to look for weak ties beneath the particular goal to link the individual computer system as a virtual connected system in order to achieving the particular goal of tasks within organization. The figure 3 will demonstrate the objective weak tie model.

When the special main goal is defined, the essential elements or attributes will be determined. Using these elements or attributes as weak ties in different existing computer systems, an integrated and innovated virtual system will then be formed. The objective weak tie model will be established without rebuilding a new computer system.

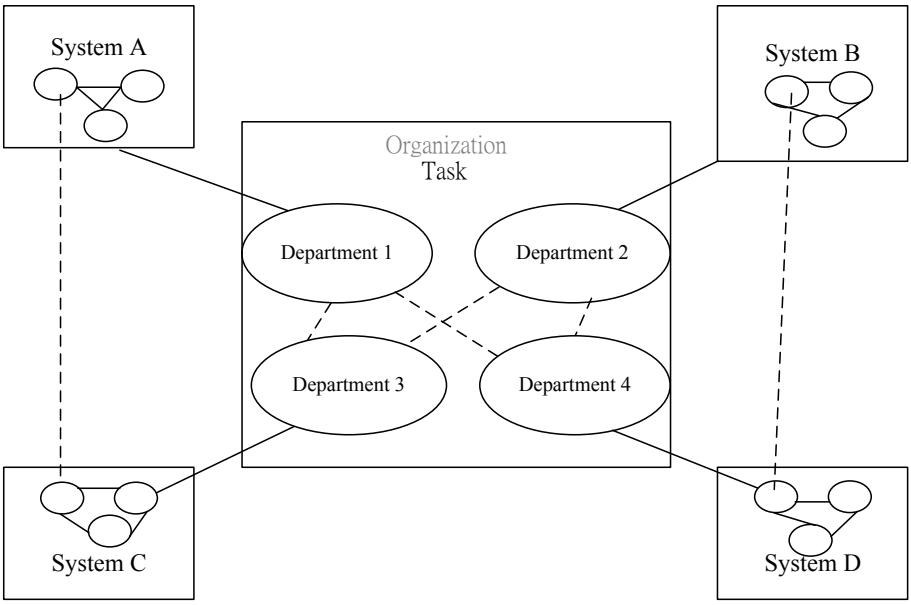


Fig. 3. The Objective Weak tie Model

4 Case Illustration

Now, a real case in higher education is used to illustrate the objective weak tie model. The administration department named Office of Academic Affairs in a university established four computer systems independently for four aims. The four computer systems were the syllabus system, class selection system, test system, and score system. The syllabus system asked teachers to provide their syllabuses on line before the class begin. The class selection system asked students to choose all they would take courses in a semester. The test system asked teachers to give a paper and pencil test for students. Finally, the score system will asked teachers to grade the score for whole semester.

So, the first step for this model was to indicate the important attributes in each computer system. In syllabus system, the attributes were the instructional goals, objectives, core competence indicators, and some other attributes. Figure 4 was part of the syllabus system shown the core competence indicators. In addition to these attributes, there are the basic attribute such as the course number, course name and publish their syllabus on line.

The other computer system will be the test system. In test system would ask instructors to produce tests for midterm and final exam. Then, another computer system called score system would demand faculties to submit and upload the detail score and final score for whole semester. Figure 5 shown part of the score system. The attributes in score system were the midterm score, final exam score, and term score, homework or quiz score; moreover, the course id, course name, instructor name student name were also included.

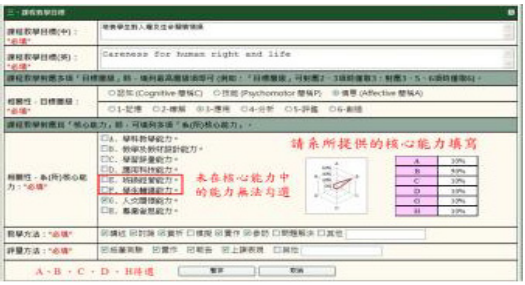


Fig. 4. Part of the syllabus system

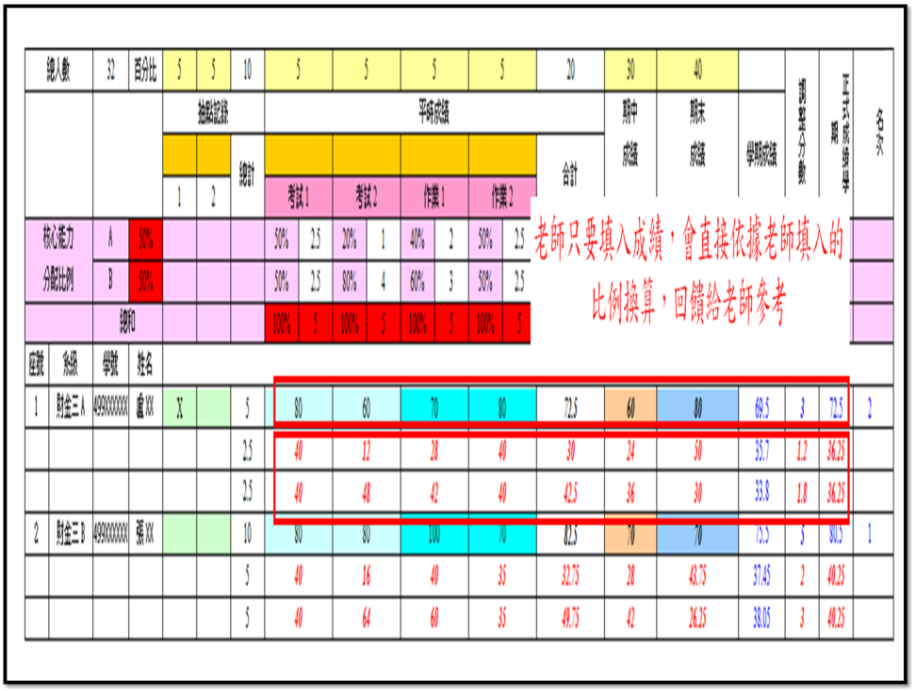


Fig. 5. Part of the score system

Examining these three existing computer system, there were some attribute related to each other; however, each system had its own different purposes. These attributes would connect to each other for relationship. Nevertheless, if there was no particular goal to achieve, these computer systems would be stand along by themselves. So, the reason why used these relationship as weak ties to link these individual system would be to accomplish a particular value. In this case the value is to reflect the student learning. The thinking would be focus on this value only. With this value, only one task needed to be done, defining the student core competence indicators for each course as table 1.

Table 1. The student core competence indicators for each course

Course Name	Goal A	Goal B	Goal C	100%
Education Theory	30	40	30	100
Education Technology	50	50	---	100

Using these values in the table, each individual computer system would be an integrated and innovated virtual student core competence system without making a new system or a heaver working load for staffs and faculties.

5 Conclusion

This study is to provide a frame work for using objective weak ties approach for integrated individual computer as a virtual curriculum and instruction system. The new virtual system combines the value focus thinking technology and curriculum developing model to enhance the strength of the weak ties. The result, namely the final outcomes indicated that the attributes such as course id, course name, student id, score, would be the weak ties within the whole system by the one goal, the student core competence indicators.

5.1 Results and Discussion

The result indicated that the student core competence indicators would be the main goal for curriculum and instruction in higher education. Based on these indicators the weak ties would link each individual computer system. This virtual system would be one system as well as each physical computer system alone. This frame work would save time and human power by virtual computer system. The strength of weak ties is proof in this study the same as the point of Granovetter. In order to making the organization more efficiency, the knowledge management would be taking care within the organization. The student core competence indicator value would be spread in each staff and faculties as the organization value. The knowledge management and innovation diffusion would be the future studies.

6 Conclusion and Suggestion

The objective weak ties approach for integrating or innovating systems is one of the simple ways to make a reform in curriculum and instruction in higher education. The reason is that many individual computer systems already exist in university usually. So, using the value and the strengths of weak ties saves a lot of work. In addition, it is also important that by using this model would not increase the working loading. Therefore, only making slice differences with weak ties will make a big successful in higher education quality. However, the quality of the student competence indicators will be the main value for people in higher education to concern. As the results, the main value and the strength of the weak ties are the two essential factors for improving higher education quality.

More research is needed for future study.

Acknowledgments. We would like to thank many staff members in Tamkang University.

References

1. Birnbaum, R.: Management Fads in Higher Education: Where They Come From, what They Do, Why They Fail. Jossey-Bass, San Francisco (2000)
2. Barnett, R.: Improving Higher Education: Today Quality Care. Open University Press, Buckingham (1992)
3. Granovetter, M.S.: The Strength of Weak Ties. American Journal of Sociology 78, 1360–1380 (1973)
4. Granovetter, M.S.: The strength of Weak Ties: A Network Theory Revisited. In: Marsden, P.V., Lin, N. (eds.) Social Structure and Network Analysis. Sage, Beverly Hills (1982)
5. Krackhardt, D.: The Strength of Strong Ties: The Importance of Philos in Organizations. In: Nohria, N., Eccles, R.G. (eds.) Networks and Organizations: Structure, Form, and Action. Harvard Business School Press, Boston (1992)
6. Keeney, R.L.: Value Focused Thinking – A Path to Creative Decision making. Harvard University Press, Cambridge (1992)
7. Wang, L.-H., Hong, C.-F., Hsu, C.-L.: Closed-Ended Questionnaire Data Analysis. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 1–7. Springer, Heidelberg (2006)
8. Huang, C.J., Tsai, P.H., Hsu, C.L.: Exploring Cognitive Difference in Instructional Outcomes Using Text Mining Technology. In: Proc. of 2006 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2116–2120. IEEE Press, New York (2006)
9. Tsai, P.H., Huang, C.J., Hong, C.F., Wang, L.H., Sung, M.Y., Hsu, C.L.: Discover Learner Comprehension and Potential Chances from Documents. In: The 11th IPMU International Conference, Paris, France (2006)
10. Levin, D.Z., Cross, B.: The Strength of Weak Ties You Can Trust: The Mediating Role of Trust in effective Knowledge Transfer. Management Science 30(1), 1477–1490 (2004)
11. Hsu, C.L., Hong, C.F., Wang, A.L., Chiu, T.F., Chang, Y.F.: Value Focused Association Map (VFAM) - An Alternative Learning Outcomes Presenting. In: Word Conference on Educational Multimedia, Hypermedia & Telecommunications, Hawaii, USA, June 22-26, pp. 3270–3275 (2009)
12. Hsu, C.L., Chang, Y.F.: Study of the Relationship with the Media Material and the Students' Learning motivation. J. Educational Study 116, 64–76 (2003)
13. Hsu, C.L.: E-CAI Case Study. Educational Technology and Media 33, 28–35 (1997)
14. Hsu, C.L., Kuo, C.H.: Study of e-Learning Material Technology. In: 2000 e-Learning Theory and Practice Conference, pp. 61–65. National Chiao Tung University, Shin-Chu (2000)
15. Hsu, C.C., Wang, L.H., Hong, C.F., Sung, M.Y., Tasi, P.H.: The KeyGraph Perspective in ARCS motivation model. In: The 6th IEEE International Conference on Advanced Learning Technologies, pp. 970–974. IEEE Press, New York (2006)
16. Hsu, C.C., Wang, L.H., Hong, C.F.: Understanding students' Conceptions and providing Scaffold Teaching Activities. In: International Conference of Teaching and Learning for Excellence, Tamsui, pp. 166–175 (2007)
17. Marsh, C.J., Willis, G.: Curriculum alternative approaches, ongoing Issues. Prentice-Hall Inc., New Jersey (1999)

Business Opportunity: The Weak-Tie Roaming among Tribes

Chao-Fu Hong[1], Mu-Hua Lin[2], and Hsiao-Fang Yang[2]

[1] Department of Information Management, Aletheia University
au4076@au.edu.tw
[2] Management Information Systems, National Chengchi University
95356503@nccu.edu.tw, hfyang.wang@gmail.com

Abstract. In this study we assume that the opportunity is already exists in the world. Furthermore, the weak-tie strategy is used to recognize two or more useful tribes and compare what differences between tribes to find out useful infor-mation, such as business opportunity for creating innovative service. At last, this model is used to design new Bali service to evidences our model is useful.

Keywords: business opportunity, consuming tribe, weak-tie, value-focused thinking.

1 Introduction

Because of people preferences particular products or brands, or has similar life expe-riences or ideas, which could help them to build their social blocks, as tribes or neo-tribes (Cova, 1997). Furthermore, Granovetter [4] illustrates social networks how to work: friends from other groups (tribes) came to see him and bring information about new jobs for him is more important than he and his close friends (strong tie) have.

This social phenomenon means that early adopters roam among the tribes, to get a chance for integrating differential use innovations of tribes to create an innovative use. In that time, they should be "values first" to iterate between articulating values and creating alternative for identifying objectives, and helps researcher to discover hidden objectives [7], such as early adopters, they may accept new products and may create innovative ways of using the products (business opportunity) and that innovative use can convince the early majority to accept the new product [12].

Summary above discussion, this study includes two contributions: the first is to emerge the features of all tribes and find out tribal weak-tie as the business opportunity. Second is based on weak-tie to extend the linkages and to connect with other tribes, that differential successful experiences will be brought into tribe, and help us to generate innovative alternative for building new market.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 532–539, 2012.

2 Literature Review

When new products are introduced to the market, according to Roger's IDM [12] only early adopters will purchase new products and create innovative ways of using them. Furthermore, if more tribes are merged, we could build a larger market. In this section, we will review relevant literature to demonstrate our research is executable.

2.1 The Innovative Use, Social Influence and Business Opportunity

In this section we are going to do detail discussing with the early adopters how to influence innovation diffusion. Firstly, [10] defined innovative use as "the degree to which an individual is relatively earlier in adopting an innovation than other members of his social system." This definition of innovation is limited to a purchasing context. In addition, creative consumers may possess special skills and abilities required for using the product in a wide variety of ways [9]. [5] extends the concept of innovation to two other categories - use innovation and vicarious innovation. But for innovative product, consumer always cares when he/she need to use innovative product to solve the novel problem.

From other view spot, social influence is the interactions within individuals of a group, or the fundamental role as a medium for spreading information, ideas, and influence among its members [1]. Besides, [13] tell us that individuals must possess prior knowledge to perceive the value of innovative product and to identify an opportunity. These actions state that the process is start from human, who compares ideas of new product, as "business opportunity" processing to build new market [14]. This implies that recognizing key early adopters (neo-tribes) to understand their innovative ways of using new product are very important for triggering social influence.

As mention above, the uses of innovative products are conceptualized as a consumer's receptivity/attraction to and creativity with using innovative products in new ways [9], and innovative ways of using products are created by early adopters. To return to Rogers' IDM [12], if the early majority could not obtain or accept the uses of the innovative products of the early adopters, a chasm will exist between them [Moore]. This means that the uses of innovative products are the key factors of social influence (SI) for influencing the majority to cross the chasm. Then, innovative use is the opportunity of crossing chasm, and help decision maker to format the new business. In the next section we try to discuss the technologies how to extract opportunity.

2.2 Grounded Theory and Text Mining

Grounded theory (GT) is used to analyze the data to find new uses of innovative products. Nevertheless, the validity in its traditional sense is consequently not an issue in GT, which instead judges by fit, relevance, workability, and modifiability. Therefore, how to develop workable information technology to reduce human power on GT analysis will become an important research issue. Furthermore, Qualitative Chance Discovery model (QCD) [6] and Human-Centered Computing System (HCCS) [8],

they try to reduce human power in GT analysis, there combine GT with text mining to extract new ideas.

As previously discussing, we may have two rough ideas to let innovative idea diffuse on the social network. First is that the social network has to be constructed. Second is innovative idea, such as weak-tie, must be discovered and is used to bridge tribes. Therefore, we attempt to develop a new method to find out innovative idea.

3 Methodology

The innovative use is the opportunity to cross the chasm, which can help decision maker to construct new business and to influence other consumers. In addition, if he also is the member of different tribes, he may serve as a weak tie, bridging and affecting members of the entire tribes to create an innovative market. Therefore, this study aims to propose a two cycle model: first cycle, he has to identify first tribal innovative use and it is used to discover other tribes, second cycle, in that tribe the innovative use is not special, there are various innovative uses richen their daily life. The researcher can based on difference between two tribes finds out useful information to design an influential alternative, such as Medici effect. The research flowchart is shown as follows.

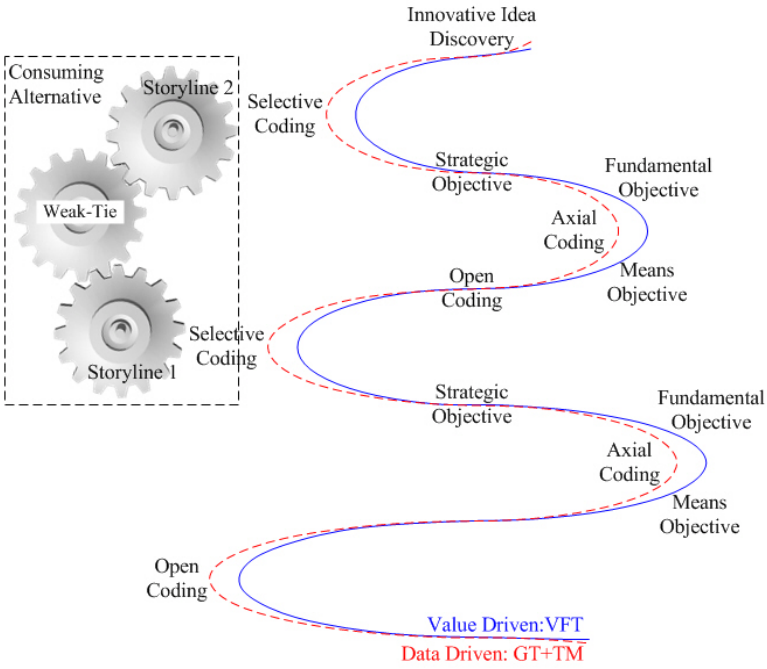


Fig. 1. A research flowchart for extracting key early adopter

3.1 Two Cycle Human-Centered Computing System

In Rogers' IDM [12], the uses of innovative products also are the business opportunity, or the key factors of Social Influence (SI) for influencing the majority to accept the innovative products and to format new product market. Therefore, this study wants using two concepts to develop a process model for discovering business opportunity. The first concept is value-focused thinking (VFT) [7], because VFT is not only includes a process for identifying objectives, but also involves discussions with relevant decision makers and stakeholders to move quickly away from the ill-defined to the well-defined, from constraint- free thinking to constrained thinking. And then, they focus on the useful values for guiding the decision situation remove the anchor on narrowly defined alternatives and make the search for new alternatives a creative and productive exercise. The other concept is that we used grounded theory (GT) [15] and employed text-mining method to extract new use or innovative idea. After first cycle analysis, the innovative idea is, as weak-tie, which bridges into other tribes for finding out their key successful factors, which will be brought back for designing innovative service, in second cycle analysis. The detailed process is listed as follows:

First Cycle Analysis.

Phase 1: Preparation for data and labeling process.

Start from value driven: based on researching interesting, researcher defines the domain and relevant keywords he/she intends to study. Then entering data driven process: system sifts out the data which correspond to keywords from the Internet. Based on his/her domain knowledge, the researcher interprets the texts, and at the same time, segments texts into words, and removes useless words. System calculates the co-occurrence of words in all sentences, to analyze the associative relationship between all words and visualize the analysis result. The researcher identifies keywords as concepts and the clusters as categories derived from the co-occurrence association diagram, and gives the clusters' label, such as topic1, topic2, and so on, which helps the researcher to preliminarily realize the various theme values presented in the data.

Phase 2: Construct the tribe (social network) by template of consumers.

In this step, customers create many uses and freely share their innovations to others [11]. So, the researcher uses "the technical capabilities and use" as the template of tribe to extracts the documents and search out the useful sentences data to create a use clusters. Then based on the analysis done in phase 1, which the researcher discovers various types of clusters/tribes, the different types of use key factors also are used to assign tribes, and clusters/tribes are derived from consuming data. Therefore, in the same cluster/tribe, they have similar techniques and use, such as tribal values.

Phase 3: Extract key innovative idea from consumers.

After phase 2, researcher discovers various type tribes, the decision maker adjusts values, such as integrate relative values, then he/she integrates and analysis data. Some tribes are linked by terms that terms are called innovative idea.

Second Cycle Analysis.
Phase 4: Weak-tie strategy to extract another tribal use as key successful factors.
The researcher uses innovative idea to find out the relative Weblogs, and follows Phase 1 to Phase 3 analysis. Then various key uses are found in another type tribe.

Phase 5: Construct the innovative alternative.
The researcher compares two tribal terms (values), to discover relative innovative terms (values), and to help him/her generating an innovative alternative.

4 A Case Study

In 2011, according to bulletin of Department of budget, accounting and statistics of New Taipei city government, number of consumers visited Bali district is third big number of traveler to visit all districts of New Taipei city, but the number of visiting is still less than the number of first place about 350,000. In addition, Bali has many nature sciences, such as Shihsanhang Museum of Archaeology, left bank (rive Gauche). That Bali has many chances to improve traveling service or to create innovative traveling service, for attracting more travelers to visit it. For this reason, we try to discover new service for Bali.

4.1 Data Resources

The researchers collected data posted on blogs relevant to traveling Bali. These data ranged from January 1, 2011 to December 31, 2011. Using Google blogs (http://blogsearch.google.com/ blogsearch) and the keywords, Bali traveling, to search for the data, the researchers obtained 63 related data from blog articles. After we carefully read the collected data and removed the articles that did not contain innovative uses. Then researcher executed a value from these 60 articles.

4.2 Experimental Results

Based on novel HCCS, the researchers used their knowledge to extract the traveling thought from weblogs to investigate the travel saturation. The results are shown in Fig. 2. From Fig. 2, the researchers identified some consuming characteristics in traveling Bali, e.g. Shihsanhang Museum of Archaeology, left bank (rive Gauche), Paris and Café. Additionally, the researchers also found that consumers have planned when they visit Bali, cycling in left bank (rive Gauche), visiting museum, and have a coffee. Following phase 2 and phase 3, basically, there are many axial clusters, such as customers walking or riding bicycle in left bank (rive Gauche) to support consumers leisure life etc., are emerged. These consuming lifestyles are famous but not innovative in Taiwan. From rare information, the innovative idea is discovered from few consumers, that they are not only enjoy leisure time in left bank (rive Gauche) to

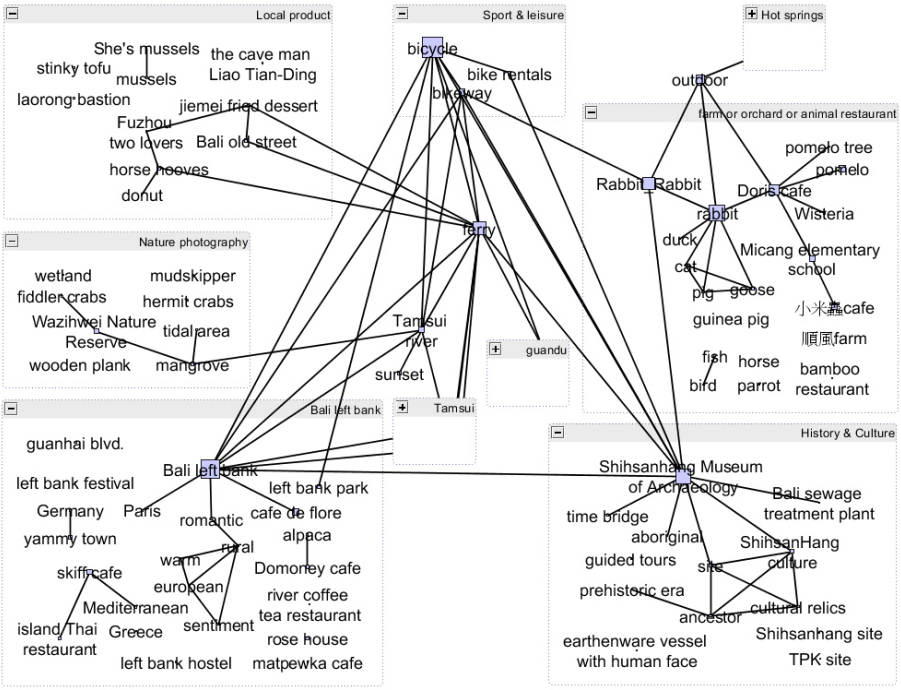


Fig. 2. The consuming characteristics of Bail

drink coffee, ride bicycle, and enjoy nature science, but also thought the mood of Bali is like romantic Paris. This scenario helps the researchers perceive the innovative service is the mood of left bank (rive Gauche) in Bali may support consumers to enjoy the romantic mood like in Paris. But the experimental results do not clearly pointed out how to do.

The romantic Paris, as Granovetter's story illustrates, friends who were not so close served as the weak tie (bridge), are connected with other clusters (tribes) and gave them a different piece of information. Therefore, in phase 4, romantic Paris is used to collect another data, and follows phase 1-3 to analyze what romantic exits in Paris. Compare these two tribes (Bali and Paris); the researchers identified some consuming characteristics in traveling Paris shown in Fig. 3. From Fig. 3, we could understand that on Paris there have many palaces romantic stories are happen in right bank (rive Droite). Therefore, litterateurs or novelists always drink coffee, see right bank (rive Droite), and write down the constructed story in left bank (rive Gauche). The environment of Bali is similar with Paris. Tamsui district stands in right bank (rive Droite). Tamsui had been the Spain's colony, Holland's colony and English's colony, European style architectures are still stood on right bank (rive Droite). In addition, Mackey taught Christian religion in Tamsui also is a famous story in Taiwan. In order to implement key successful factor of left bank (rive Gauche) of Paris

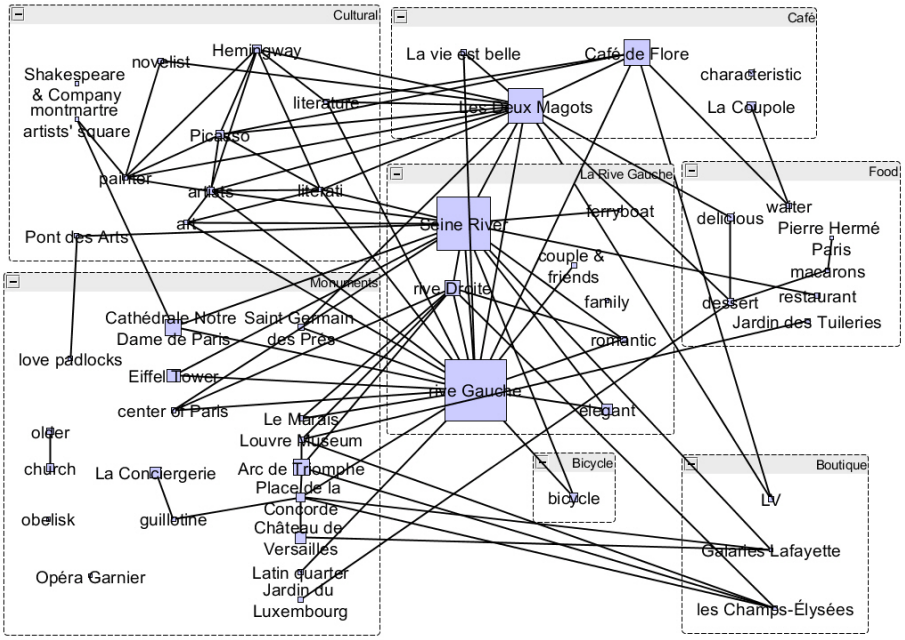


Fig. 3. The consuming characteristics of Paris

into Bali, they must not only sell the coffee, but also have to serve consumers seeing the architectures and understanding the stories in right bank (rive Droite) for creating innovative service.

5 Conclusion

To recognize business opportunity, it not only needs prior knowledge, but also has a good framework to discover opportunity. In this paper we propose a two cycles HCCS: in first cycle, VFT is used to guide value driven for recognizing useful values, and then the useful values are used to guide data driven for finding weak-tie, as innovative idea. In second cycle, a weak-tie is a bridge to start value driven, to discover other tribe. After second data driven, the key factor of managing coffee shop is discovered in left bank (rive Gauche) on Paris. This key successful factor of managing coffee shop in left bank (rive Gauche) on Paris maybe is a good strategy for us manages coffee shop in left bank (rive Gauche) on Bali. This experimental result explains that our two cycle novel HCCS is good for discovering innovative service.

Acknowledgment. This research is supported by the National Science Council (NSC), Taiwan, R.O.C. (NSC 99-2632-H-156 -001 -MY3).

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: SIGMOD 1993 Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM, New York (1993)
2. Bickart, B., Schindler, R.M.: Internet forums as influential sources of consumer information. Journal of Interactive Marketing 15(3), 31–40 (2001)
3. Cova, B.: Community and consumption: Towards a definition of the "linking value" of product or services. European Journal of Marketing 31(3/4), 297–316 (1997)
4. Granovetter, M.: The strength of weak ties: a network theory revisited. Sociological Theory 1, 201–233 (1983)
5. Hirschman, E.C.: Innovativeness, novelty seeking and consumer creativity. The Journal of Consumer Research 7(3), 283–295 (1980)
6. Hong, C.-F.: Qualitative chance discovery: Extracting competitive advantages. Information Sciences 179(11), 1570–1583 (2009)
7. Keeney, R.L.: Value-Focused Thinking: A path to creative decisionmaking. Harvard University Press (1996)
8. Lin, M.-H.: Opportunities for Crossing the Chasm between Early Adopters and the Early Majority through New Uses of Innovative Products. The Review of Socionetwork Strategies 5(2), 27–42 (2011)
9. Price, L.L., Ridgway, N.M.: Use innovativeness, vicarious exploration and purchase exploration: Three facets of consumer varied behavior. In: Proceedings of the 48th Educator's Conference, pp. 56–60. American Marketing Association, Chicago (1982)
10. Robertson, T.S.: Innovative behavior and communication. Holt, Rinehart and Winston, Inc., New York (1971)
11. Rogers, E.M., Shoemaker, F.F.: Communication of innovations: A cross-cultural approach, 2nd edn. The Free Press, New York (1971)
12. Rogers, E.M.: Diffusion of Innovations. Free Press, New York (2003)
13. Shane, S., Venkataraman, S.: The promise of entrepreneurship as a field of research. Academy of Management Review 25(1), 217–226 (2000)
14. Shane, S.: A general theory of entrepreneurship: The individual-opportunity nexus. Edward Elgar, UK (2003)
15. Strauss, A.C., Corbin, J.M.: Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd edn. Sage Publications, Inc. (1998)
16. Venkatesh, A., Vitalari, N.P.: Computing technology for the home: Product strategies for the next generation. Journal of Product Innovation Management 3, 171–186 (1986)

Emerging Technology Exploration Using Rare Information Retrieval and Link Analysis

Tzu-Fu Chiu[1], Chao-Fu Hong[2], and Yu-Ting Chiu[3]

[1] Department of Industrial Management and Enterprise Information, Aletheia University, Taiwan, R.O.C.
[2] Department of Information Management, Aletheia University, Taiwan, R.O.C.
[3] Department of Information Management, National Central University, Taiwan, R.O.C.
{chiu,cfhong}@mail.au.edu.tw, gloria@mgt.ncu.edu.tw

Abstract. To explore the clues of an emerging technology is essential for a company or an industry so that the company can consider the feasibility of resource allocation to the technology and the industry can observe the developing directions of the technology. Patent data contains plentiful technological information from which it is worthwhile to extract further knowledge. Therefore, a research framework for emerging technology exploration has been formed where rare information retrieval is designed to sift out the rare patents, cluster analysis is employed to generate the clusters, and link analysis is adopted to measure the link strength between the rare patents and clusters. Consequently, the rare patents were found, the clusters were generated and named, the notable rare patents were recognized, and the potentiality of emerging technology was discussed. Finally, the notable rare patents and the potentiality of emerging technology would be provided to the decision makers of companies and industries.

Keywords: emerging technology, rare information retrieval, link analysis, patent data, thin-film solar cell.

1 Introduction

Emerging technology is often viewed as a complementary or substitute solution and the opportunity to create entirely new business which is neglected [1]. It is also essential for a company or an industry to realize the starting point of a possible technology so that the company can consider the feasibility of resource allocation to the technology and the industry can monitor the developing directions of the technology. In addition, a rare and notable information appears in a technical area could be a non-ignorable idea or clue of an emerging technology. As up to 80% of the disclosures in patents are never published in any other form [2], it would be worthwhile for researchers and practitioners to explore the potentiality of an emerging technology upon the patent database. Therefore, a rare information retrieval technique will be proposed so as to find out the rare patents from the patent database; a link

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 540–549, 2012.

strength measure method will be designed so as to identify the notable rare patents from the possible ones. Afterward, the notable rare patents and its linked clusters will be utilized to describe the emerging technology.

2 Related Work

As this study is aimed to explore the emerging technology of thin-film solar cell, a research framework needs to be constructed via a consideration of rare information retrieval, link analysis, and cluster analysis. Therefore, the related areas of this study would be emerging technology exploration, patent data, thin-film solar cell, rare information retrieval, link analysis, and cluster analysis.

2.1 Emerging Technology Exploration

Emerging technology is a science-based innovation that has the potential to create a new industry or transform an existing one [3]. It is often viewed as a complementary or substitute solution and the opportunity to create entirely new business which is neglected [1]. The commonly used methods for identifying emerging technology are: citation-based or text-based analysis for indexing categories and vocabularies, data mining (mainly clustering and factor analysis), and scientometric analysis (including co-authorship analysis, co-word analysis and citation analysis) [4]. In this study, a research framework, formed by rare information retrieval, link analysis, and cluster analysis, will be used for conducting an emerging technology exploration upon thin-film solar cell via patent data.

2.2 Patent Data and Thin-Film Solar Cell

A patent document is similar to a general document, but includes rich and varied technical information as well as important research results [5]. Patents can be gathered from a variety of sources, such as the Intellectual Property Office in Taiwan (TIPO), the United States Patent and Trademark Office (USPTO), the European Patent Office (EPO), and so on. A patent document contains numerous fields, such as: patent number, title, abstract, issue date, application date, application type, assignee name, international classification (IPC), US references, claims, description, etc.

Solar cell, a sort of green energy, is clean, renewable, and good for protecting our environment. It can be mainly divided into two categories (according to the light absorbing material): crystalline silicon (in a wafer form) and thin films (of other materials) [6]. A thin-film solar cell (TFSC), also called a thin-film photovoltaic cell (TFPV), is made by depositing one or more thin layers (i.e., thin film) of photovoltaic material on a substrate [7]. The most common materials of TFSC are amorphous silicon or polycrystalline materials (such as: CdTe, CIS, and CIGS) [6]. In recent years (2003-2007), total PV production grew in average by almost 50% worldwide, whereas the thin film segment grew in average by over 80% and reached 400 MW or 10% of total PV production in 2007 [8]. Thin film is the most potential segment with

its highest production growth rate in the solar cell industry, and it would be appropriate for academic and practical researchers to contribute efforts to this technology.

2.3 Rare Information Retrieval

Rare information sometimes grows into a prevalent concept, if they satisfy the desire of people for information [9]. A rare and notable information appears in a technical area could be a non-ignorable idea or clue of an emerging technology. As up to 80% of the disclosures in patents are never published in any other form [2], this study tries to find out the rare information from the patent database so as to explore the clues of an emerging technology. Meanwhile, an IPC (International Patent Classification) is a classification derived from the International Patent Classification System (supported by WIPO) which provides a hierarchical system of symbols for the classification of patents according to the different areas of technology to which they pertain [10]. The IPC of patents are assigned by the examiners of the national patent office and contain the professional knowledge of the experienced examiners [11]. Therefore, it would be reasonable for a research to base on the IPC to extract out the rare information [12]. Here, an IPC-based rare patent retrieval technique will be proposed and described as follows:

(1) To generate a dataset of patents, *D*, with regard to a technical area for a certain period of time.

(2) To calculate the number of IPC codes contained in a patent, *Num-of-IPC-in-Patent*, according to *D*.

(3) To calculate the number of patents connecting to the same IPC code, *Num-of-Patent-in-IPC*, according to *D*.

(4) To identify a patent as a rare information if the *Num-of-IPC-in-Patent* = 1 and *Num-of-Patent-in-IPC* = 1 simultaneously.

2.4 Link Analysis

Link analysis is a collection of techniques that operate on data that can be represented as nodes and links [13]. A node represents an entity such as a person, a document, or a bank account. A link represents a relationship between two entities such as a reference relationship between two documents, or a transaction between two bank accounts. The focus of link analysis is to analyze the relationships between entities. The areas related to link analysis are: social network analysis, search engines, viral marketing, law enforcement, and fraud detection [13]. In search engines, the page rank of page A, $PR(A)$, can be calculated as in Equation (1), where T_j is a page pointing to A; $C(T_j)$ is the number of going out links from page T; and d is a minimum value assigned to any page [14]. In social network analysis, the degree centrality of a node can be measured as in Equation (2), where $a(P_i, P_k) = 1$ if and only if P_i and P_k are connected by a link (0 otherwise) and n is the number of all nodes [15].

$$PA(A) = d + (1-d) * \sum_{j=1}^{n} (PR(T_j) / C(T_j)) \quad (1)$$

$$C_D(P_k) = \sum_{i=1}^{n} a(P_i, P_k) / (n-1) \quad (2)$$

In this study, the link strength of node H in a cluster is measured by Equation (3), where V_j is a node linking to the node H; $Fr(H)$ is the frequency of node H (which is already divided by the maximum frequency of that cluster); $Ja(H,V_j)$ is the Jaccard Coefficient between nodes H and V_j; n is the number of nodes linking to node H; and w is a weight assigned to the node H. The link strength of node R in a rare patent is measured by Equation (4), where H_i is a node in the clusters linked by the node R; and m is the number of nodes linked by the node R. The link strength of a rare patent is also measured by Equation (4), where m is the number of nodes in the rare patent.

$$Str(H) = w * Fr(H) + (1-w) * (\sum_{j=1}^{n} Fr(V_j) * Ja(H,V_j)) / n \quad (3)$$

$$Str(R) = 1 - \prod_{i=1}^{m} (1 - Str(H_i)) \quad (4)$$

2.5 Cluster Analysis

Cluster analysis divides data into groups (clusters) that are meaningful, useful, or both [16]. Classes, or conceptually meaningful groups of objects that share common characteristics, play an important role in how people analyze and describe the world. Clusters are potential classes and cluster analysis is the study of techniques for automatically finding classes [16]. Cluster analysis will be employed in this study for measuring the similarity nature of documents, so as to divide patent data into groups and to represent the mainstream directions of thin-film solar cell. The storylines of rare information will be organized and stated based on the related mainstream directions.

3 A Research Framework for Emerging technology Exploration

As this study is attempted to observe the potentiality of emerging directions in thin-film solar cell, a research framework for emerging technology exploration, based on rare information retrieval, link analysis, and cluster analysis, has been developed and shown in Fig. 1. It consists of five phases: data preprocessing, rare patent retrieval,

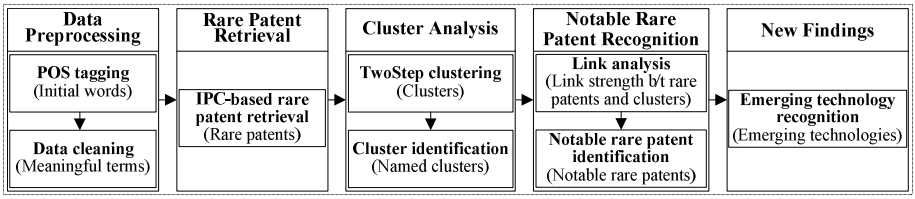


Fig. 1. A research framework for emerging technology exploration

cluster analysis, notable rare patent recognition, and new findings; and will be described in the following subsections.

3.1 Data Preprocessing

In first phase, the patent data of thin-film solar cell (during a certain period of time) will be downloaded from the USPTO [17]. For considering an essential part to represent a complex patent data, the Title, Abstract, Assignee, and Issue Date fields are selected as the objects for this study. Afterward, two processes, POS tagging and data cleaning, will be executed to clean up the source textual data.

(1) POS Tagging: An English POS tagger (i.e., a Part-Of-Speech tagger for English) from the Stanford Natural Language Processing Group [18] will be employed to perform word segmenting and labeling on the patents (i.e., the abstract field). Then, a list of proper morphological features of words needs to be decided for sifting out the initial words.

(2) Data Cleaning: Upon these initial words, files of n-grams, stop words, and synonyms will be built so as to combine relevant words into compound terms, to eliminate less meaningful words, and to aggregate synonymous words. Consequently, the meaningful terms will be obtained from this process.

3.2 Rare Patent Retrieval

Second phase is intended to perform an IPC-based rare patent retrieval function for finding out the rare patents.

IPC-Based Rare Patent Retrieval: According to the description of an IPC-based rare patent retrieval technique in Subsection 2.3, a program will be written for sifting out the rare patents based on the IPC field of patents. The rare patents will be described by its Patent Number, IPC, Issue Date, and Title fields and will be utilized for notable rare patent recognition in a later phase.

3.3 Cluster Analysis

Third phase is designed to conduct the cluster analysis via TwoStep clustering and cluster identification so as to obtain the clusters of thin-film solar cell.

(1) TwoStep Clustering: In order to carry out cluster analysis, a TwoStep clustering is adopted from SPSS Clementine for grouping patents into clusters [19]. TwoStep clustering is a scalable cluster analysis algorithm designed to handle very large data sets. It can handle both continuous and categorical variables (or attributes). It requires only one data pass. It has two steps: (a) to pre-cluster the cases (or records) into many small sub-clusters; (b) to cluster the sub-clusters resulting from pre-cluster step into the desired number of clusters [19].

(2) Cluster Identification: The above clusters will be named via summarizing the Title field of its composed patents and checking over the domain knowledge. Each named cluster is then identified as a mainstream direction and will be utilized in the following phases.

3.4 Notable Rare Patent Recognition

Fourth phase, including link analysis and notable rare patent identification, is used to measure the link strength between rare patents and clusters and then to find out the notable rare patents.

(1) Link Analysis: Referring to Equation (3) and Equation (4) in Subsection 2.4, the link strength between rare patents and clusters will be calculated and prepared for the next step.

(2) Notable Rare Patent Identification: According to the score of link strength between rare patents and clusters and considering the issue date of rare patents, the notable rare patents will be selected. As the rare patents in recent years are more likely to be the clues of emerging technology, the time frame is divided into three periods of time: earlier (1999 to 2002), middle (2003 to 2006), and later (2007 to 2010). Therefore, the rare patents with higher score of link strength and in the later period of time will be identified as the notable ones.

3.5 New Findings

In last phase, emerging technology recognition will be used to figure out the potential emerging technologies based on the notable rare patents, clusters, and link strength.

Emerging Technology Recognition: According to the notable rare patents, named clusters, and the link strength between the rare patents and clusters, the developing potentiality of notable rare patents will be explored. The possible emerging technologies will be recognized. Both the notable rare patents and possible emerging technologies will be provided to facilitate the decision-making of managers and stakeholders.

4 Experimental Results and Explanation

The experiment has been implemented according to the research framework. The experimental results would be explained in the following five subsections: result of data preprocessing, result of rare patent retrieval, result of cluster analysis, result of notable rare patent recognition, and result of emerging technology recognition.

4.1 Result of Data Preprocessing

As the aim of this study was to explore the emerging technology via patent data, the patents of thin-film solar cell were the target data for the experiment. Mainly, the

Title, Abstract, Assignee, and Issue Date fields were used in this study. 213 issued patents during year 1999 to 2010 were collected from USPTO, using key words: "'thin film' and ('solar cell' or 'solar cells' or 'photovoltaic cell' or 'photovoltaic cells' or 'PV cell' or 'PV cells')" on "title field or abstract field". The POS tagger was then triggered to do data preprocessing. Consequently, the patents were cleaned up and the meaningful terms were obtained.

4.2 Result of Rare Patent Retrieval

Using the Patent Number and IPC fields of 213 patents, the program of IPC-based rare patent retrieval was executed. 14 rare patents were obtained and depicted below in Table 1.

Table 1. Depiction of 14 rare patents

No	Patent No	IPC	Issue date	Title
1	06974976	H01L031/109	2005/12/13	Thin-film solar cells
2	07022585	H01L021/46	2006/04/04	Method for making thin film devices intended for solar cells or silicon-on-insulator (SOI) applications
3	07109517	H01L029/06	2006/09/19	Method of making an enhanced optical absorption and radiation tolerance in thin-film solar cells and photodetectors
4	07143451	A42B001/24	2006/12/05	Hat including active ventilation
5	07163179	B64G001/10	2007/01/16	Commercial service platform in space
6	07166161	C30B029/54	2007/01/23	Anisotropic film manufacturing
7	07208756	H01L029/08	2007/04/24	Organic semiconductor devices having low contact resistance
8	07252781	C08K005/01	2007/08/07	Solutions of polymer semiconductors
9	07281334	B43L013/00	2007/10/16	Mechanical scribing apparatus with controlling force of a scribing cutter
10	07554346	G01R031/302	2009/06/30	Test equipment for automated quality control of thin film solar modules
11	07671083	A61K031/381	2010/03/02	P-alkoxyphenylen-thiophene oligomers as organic semiconductors for use in electronic devices
12	07754841	C08G079/08	2010/07/13	Polymer
13	07754847	C08G075/00	2010/07/13	Soluble polythiophene derivatives
14	07824563	C03C025/68	2010/11/02	Etching media for oxidic, transparent, conductive layers

4.3 Result of Cluster Analysis

Using the meaningful terms from data preprocessing, the clusters of 199 (213 minus 14) patents were generated (via the TwoStep clustering) in Table 2. In the table, eleven clusters were identified with the number of composed patents for cluster-1 to cluster-11: 16, 32, 16, 20, 16, 16, 16, 7, 34, 11, and 15 respectively. According to the title field and domain knowledge, these eleven clusters were named as in the 'Name of cluster' field of the table.

Table 2. 11 clusters of thin-film solar cell

Id	Name of cluster	Num. of records
cluster-1	'SOI (silicon on insulator) & light-absorbing-film'	16
cluster-2	'amorphous-film & organic-light-emitting & thermal-annealing'	32
cluster-3	'CIGS-material & film-deposition'	16
cluster-4	'anti-reflection & light-trapping'	20
cluster-5	'thermal-emissive-coating & metal-organic-compound'	16
cluster-6	'porous-layer & etching-process'	16
cluster-7	'PECVD-method & RF(radio frequency)-sputtering'	16
cluster-8	'organic-chemical-vapor & OLED (organic light-emitting diode)'	7
cluster-9	'roll-to-roll-process & porous-structure & transparent-substrate'	34
cluster-10	'CdTe-film & thermal-radiator'	11
cluster-11	'aromatic-enediyne & organic-semiconductor'	15

4.4 Result of Notable Rare Patent Recognition

Using the results of rare patent retrieval and cluster analysis, the link strength of 14 rare patents were calculated and summarized as in Table 3. Subsequently, according to the selection criterion in Subsection 3.4, the notable rare patents were determined via the rank of link strength and focusing on the later period of time (2007 to 2010). They were patents: '07252781' and '07281334' in year 2007 as well as '07754841' and '07824563' in year 2010, with italic face and purple color in Table 3.

Table 3. Notable rare patents of thin-film solar cell

No	Patent No	Link strength	Rank	Issue date
1	06974976	0.939739737	1	2005/12/13
2	07022585	0.886278523	4	2006/04/04
3	07109517	0.914101627	2	2006/09/19
4	07143451	0	13	2006/12/05
5	07163179	0	13	2007/01/16
6	07166161	0.494747223	8	2007/01/23
7	07208756	0.345699571	10	2007/04/24
8	*07252781*	*0.721626569*	*7*	*2007/08/07*
9	*07281334*	*0.904466718*	*3*	*2007/10/16*
10	07554346	0.119544877	12	2009/06/30
11	07671083	0.452324995	9	2010/03/02
12	*07754841*	*0.797246441*	*5*	*2010/07/13*
13	07754847	0.29195807	11	2010/07/13
14	*07824563*	*0.765514354*	*6*	*2010/11/02*

4.5 Result of Emerging Technology Recognition

According to the above notable rare patents, named clusters, and link strength between rare patents and clusters, the developing potentiality of four notable rare patents were explored and summarized in Fig. 2 and would be stated in detail as follows.

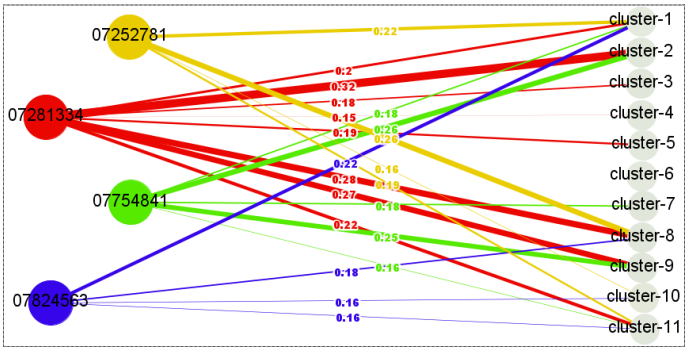


Fig. 2. Developing potentiality of four notable rare patents

(1) Potentiality of '07252781': Referring to Fig. 2, this rare patent, with a 'polymer semiconductors' feature, mainly linked to cluster 1, 8, and 11. It would be appropriate for the patent to move on to the 'light-absorbing-film', 'OLED (organic light-emitting diode)', and 'organic-semiconductor' directions.

(2) Potentiality of '07281334': According to Fig. 2, this rare patent, with a 'mechanical scribing apparatus' feature, strongly linked to cluster 2, 8, 9, and 11. It would be reasonable for the patent to move towards the 'amorphous-film', 'OLED', 'roll-to-roll-process', and 'organic-semiconductor' directions.

(3) Potentiality of '07754841': Learning from Fig. 2, this rare patent, with a 'polymer' feature, strongly linked to cluster 2 and 9; lightly linked to cluster 1, 7, and 11. It would be suitable for the patent to walk towards the 'amorphous-film', 'roll-to-roll-process', 'PECVD-method', and 'organic-semiconductor' directions.

(4) Potentiality of '07824563': Referring to Fig. 2, this rare patent, with a 'etching media' feature, significantly linked to cluster 1; lightly linked to cluster 8, 10, and 11. It would be proper for the patent to move towards the 'amorphous-film', 'OLED', 'thermal-radiator', and 'organic-semiconductor' directions.

5 Conclusions

The research framework for emerging technology exploration has been formed and applied to thin-film solar cell using patent data. The experiment was performed and the experimental results were obtained. Fourteen rare patents of thin-film solar cell during 1999 to 2010 were found via IPC-based rare patent retrieval. Eleven clusters were also generated through TwoStep clustering of SPSS Clementine. Among 14 rare patents, four notable rare patents were recognized via link strength measurement. The potentiality of emerging technology was explored and stated according to the notable rare patents, named clusters, and link strength between notable patents and clusters. The directions of emerging technology on thin-film solar cell would be helpful for managers and stakeholders to facilitate their decision-making.

In the future work, the research framework may be joined by some other methods such as co-authorship analysis or citation analysis so as to enhance the validity of

experimental results. In addition, the data source can be expanded from USPTO to WIPO or TIPO in order to explore the emerging technology on thin-film solar cell widely.

Acknowledgments. This research was supported by the National Science Council of the Republic of China under the Grants NSC 99-2410-H-156-014.

References

1. Gillier, T., Piat, G.: Exploring Over: The Presumed Identity of Emerging Technology. Creativity & Innovation Management 20(4), 238–252 (2011)
2. Blackman, M.: Provision of Patent Information: A National Patent Office Perspective. World Patent Information 17(2), 115–123 (1995)
3. Day, G.S., Schoemaker, P.J.H.: A different game. In: Day, G.S., Schoemaker, P.J.H. (eds.) Wharton on Managing Emerging Technologies, pp. 1–23. John Wiley (2000)
4. Cozzens, S., Gatchair, S., Kang, J., Kim, K.S., Lee, H.J., Ordóñez, G., Porter, A.: Emerging technologies: quantitative identification and measurement. Technology Analysis & Strategic Management 22(3), 361–376 (2010)
5. Tseng, Y., Lin, C., Lin, Y.: Text Mining Techniques for Patent Analysis. Information Processing and Management 43, 1216–1247 (2007)
6. Solarbuzz, Solar Cell Technologies (2010), http://www.solarbuzz.com/technologies.html
7. Wikipedia, Thin film solar cell (2010), http://en.wikipedia.org/wiki/Thin_film_solar_cell
8. Jager-Waldau, A.: PV Status Report 2008: Research, Solar Cell Production and Market Implementation of Photovoltaics, JRC Technical Notes (2008)
9. Ohsawa, Y., Nara, Y.: Action Proposal as Discovery of Context - An Application to Family Risk Management. In: Terano, T., Nishida, T., Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T. (eds.) JSAI 2001 Workshops. LNCS (LNAI), vol. 2253, pp. 481–485. Springer, Heidelberg (2001)
10. WIPO: Preface to the International Patent Classification (IPC) (October 30, 2010), http://www.wipo.int/classifications/ipc/en/general/preface.html
11. Kang, I.S., Na, S.H., Kim, J., Lee, J.H.: Cluster-based patent retrieval. Information Processing & Management 43(5), 1173–1182 (2007)
12. Chiu, T.-F., Hong, C.-F., Chiu, Y.-T.: A Proposed IPC-Based Clustering and Applied to Technology Strategy Formulation. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012, Part II. LNCS, vol. 7197, pp. 62–72. Springer, Heidelberg (2012)
13. Donoho, S.: Link Analysis. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 355–368. Springer (2010)
14. Weiss, S.M., Indurkhya, N., Zhang, T.: Fundamentals of Predictive Text Mining. Springer (2010)
15. Freeman, L.C.: Centrality in social networks: conceptual clarification. Social Networks 1, 215–239 (1979)
16. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison Wesley, Boston (2006)
17. USPTO: the United States Patent and Trademark Office (2010), http://www.uspto.gov/
18. Stanford Natural Language Processing Group, Stanford Log-linear Part-Of-Speech Tagger (2009), http://nlp.stanford.edu/software/tagger.shtml
19. SPSS, Clementine 10.1, Algorithms Guide. Integral Solutions Limited, USA (2006)

Introducing Fuzzy Labels to Agent-Generated Textual Descriptions of Incomplete City-Traffic States

Grzegorz Popek[1,2], Ryszard Kowalczyk[1], and Radosław P. Katarzyniak[2]

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

Abstract. An aim of this research is to create methods for a provision of textual information to users of a distributed multi-agent information system. In particular we focus on a traffic information system where agents transform the numerical data about states of the city traffic obtained using a distributed sensor network into natural language summaries. The basis for the transformation from numerical data into a linguistic domain are zadehian fuzzy-linguistic models of concepts. Unlike in typical Natural Language Generation approaches, this paper focuses on the provision of summaries in situations where data is incomplete and on conveying this incompleteness to the user using belief-based language statements. We provide an algorithm based on a theory of grounding for an agent-based evaluation of local summaries with autoepistemic operators of possibility, belief, and knowledge. We also propose a method for an aggregation of summaries generated by local agents in order to obtain a textual summary of complex structures of the road network (e.g. areas, districts, precise routes).

Keywords: intelligent agents, fuzzy labels, language summaries.

1 Introduction

The ease of use is one of the key aspects when it comes to designing systems targeted at casual users. It can be partially obtained by introducing a natural language communication between users and the system. In particular, the system can provide the user with information using natural language statements. We outline a multiagent system providing natural language-based messages to the user. Messages contain a semantic summary of numerical data acquired locally by a distributed group of independent agents. Existing solutions in an area of Natural Language Generation [3,7,15] focus mostly on a precision, short length, and lack of ambiguity of the summary in situations in which data is complete.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part II, LNAI 7654, pp. 550–561, 2012.

The focus of this research is set to handling situations in which data is partially missing. Instead of discarding such data or using imputation procedures [10], the information about the original incompleteness is conveyed to the user. It is accomplished by building textual summaries including autoepistemic operators of possibility, belief, and knowledge. In a presented setting each agent periodically observes its local environment and stores the data in a private database. Based on the proposed algorithm an agent can generate a local summary in a textual form based on collected data. To obtain a summary of a wider area (i.e. consisting of sub-areas assigned to single agents) the corresponding agent performs an aggregation of local summaries obtained from respective agents.

Enabling an agent to use semantic messages is not a straightforward task and it is strongly related to the Symbol Grounding Problem stated by Harnad [5]. We base our approach on existing partial solutions for grounding of modal statements involving binary [9] and discrete [14] properties and extend it to the case of values of properties modelled using zadehian fuzzy-linguistic concepts [16].

Section 2 contains an application scenario for a proposed system. In Section 3 we present a general structure of the system focusing on an assumed model of a cognitive agent. Section 4 describes proposed algorithm for the evaluation of local summaries. In Section 5 we describe an algorithm for an aggregation of local summaries obtained from a given region. In Section 6 we provide a discussion on the proposed solution followed by conclusions in Section 7.

2 Application Scenario

In this paper we assume a similar application scenario as in [14]. We focus on a traffic information system where agents transform the numerical data about states of city traffic into natural language summaries. The basis for the scenario is data gathered by VicRoads[1] which consists of car volumes passing through key road links in Melbourne in 15-minutes intervals.

A single agent is deployed at each main crossroad of the road network. The crossroad is treated as an environment for the respective agent. Usually crossroads consist of four links represented as objects of the environment. Each link is characterized by a numerical property (a volume of cars) changing over time.

If an agent observed the current state of traffic within the link (i.e., if the volume of cars during the last time interval has been measured), it can assign fitting linguistic concepts and in result, generate textual summary using a simple traffic description pattern "There is a ⟨traffic⟩ ⟨object_text⟩", where traffic is a linguistic concept describing a state of property traffic and object_text is a textual description assigned to the object (e.g., "*at Camberwell Junction moving east into Riversdale Road*"). Models of linguistic concepts are based on a local context. In the assumed traffic scenario the local context can be understood as properties of traffic-light cycles at a given crossroad, maximal observed volume of cars passing through the link in the past or maximal throughput of the link stated by a traffic expert.

[1] VicRoads assists the Australian Gov. to achieve its transport policy objectives.

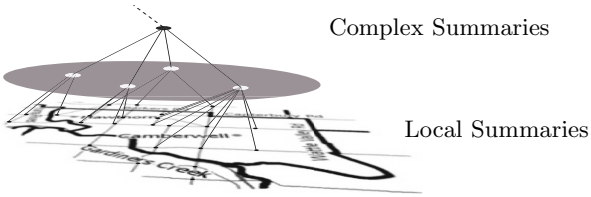


Fig. 1. Processing of summaries within a hierarchical architecture

In a situation where a respective piece of data is missing (e.g., due to a measurement failure) the agent cannot directly assign a summary. It processes the historical data in order to assign a modal summary containing one of autoepistemic modal operators of possibility, belief, and knowledge. The final textual summary is generated in an analogical way using the following pattern "⟨belief⟩ there is a ⟨traffic⟩ ⟨object_text⟩", where the belief is one out of the following three: "*It is possible that*", "*I believe that*", "*I know that*".

Summaries of complex areas (routes, districts) are constructed based on local summaries by assigned agents. In a hierarchical setting presented in Figure 1 additional agents are assigned to sub-areas and are responsible for generation of aggregated summaries. An alternative solution can be based on assigning of the aggregation task to a subset of local agents instead of deploying additional ones.

Regardless of an organisational structure used for the final implementation and for the deployment of the system, a mechanism of aggregation is needed. The mechanism which transforms multiple local summaries into an aggregated summary of a complex network structure.

Summaries of complex network structures are constructed using textual patterns which are analogical to the ones presented above for the case of generation of local summaries. There are existing predefined names for certain subset of links (e.g. districts: *Hawthorn East*, regions: *East Melbourne*, special patterns: *MCG Traffic*), others need to be constructed (e.g. routes: "*from Hawthorn Bridge to Eastern Fwy via Church St, High St, Doncaster Rd*").

As it can be seen, the presented scenario presents a need for two main mechanisms to be developed: the mechanism for a generation of local summaries and the mechanism for an aggregation of local summaries.

3 Structure of the Agent

An agent is located in a relational environment consisting of objects $\mathfrak{o} \in \mathfrak{O}$. Objects exhibit discrete properties changing over discrete time. The agent observes the environment and stores results of its observations in its private database. It is assumed that agents do not make mistakes in their observation, i.e., their observations are consistent with an objective state of the external world. However, observations can be incomplete. Generation of a message about an observed property of a given object is straightforward as there is an answer which can

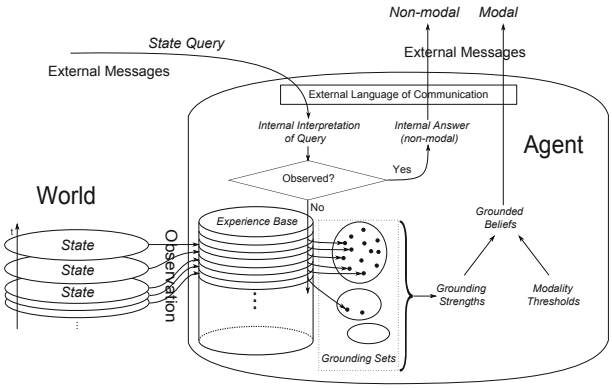


Fig. 2. A general structure of the processing performed by an agent (after [14])

be directly transformed into a textual message using patterns presented before. However, it cannot be done directly when a respective piece of data is missing.

In the assumed approach [8,9] the agent in presence of missing data reduces its lack of knowledge using its previous experiences. One of natural assumptions accepted in a theory of grounding is an ability of natural cognitive agents to fill gaps experienced in an autonomously created model for actual worlds with some mental 'patterns' extracted from previous empirical experiences.

Pieces of experience connected to concurrent states of the 'gap' are gathered in disjunctive grounding sets and lead to a raise of multiple alternative models. Each of these models can be assigned with a grounding strength with which it influences subjective convictions of the agent. Based on a distribution of these strengths and based on a system of modality thresholds (presented further) an agent grounds a certain set of statements using epistemic modal operators of possibility, belief, and knowledge.

As shown in Figure 2 an agent receives a query. In case of an incomplete observational data the agent describes its cognitive attitude toward a state of unobserved property in a given object from an environment. Epistemic operators of possibility/belief/knowledge are used to reflect an agent's internal state. Semantic messages based on such operators (constructed according to rational restrictions) are assumed to be naturally understood by a user of the system.

The approach follows works of Johnson-Laird [1] addressing a possibility of use of a model theory to deal with modal reasoning. Dennett [2] states 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*". Katarzyniak [9] uses mental models to support a choice of a modal statement related to a state of a chosen property in an object of an external environment in case where an actual observation regarding the state of the property is missing.

In the current approach a mapping between the experience base and grounding sets is not direct. Values of properties are not uniquely represented by concepts used in an external language of communication.

Each agent is equipped with a set of fuzzy mappings $\mu_1^{\mathfrak{F}}, \mu_2^{\mathfrak{F}}, \ldots$ representing meaning of respective language concepts $\mathfrak{f}_1^{\mathfrak{F}}, \mathfrak{f}_2^{\mathfrak{F}}, \ldots$ related to a particular property φ of an object $\mathfrak{o}$. Mappings may be different for each particular object. In the assumed scenario mappings for respective concepts used to describe a property Traffic are context-dependent. They depend on the maximal volume of cars for a corresponding link (provided by an expert or derived from past data).

4 Generation of Local Summaries

As it has been mentioned before, generation of fitting non-modal summaries – i.e. summaries generated when a respective piece of data is present in the agent's database – is relatively straightforward. Traffic experts together with a system designer need to provide a value of μ_{min} which will be used as a threshold of activation of concepts.

For the notational simplicity we will represent language statements in a structured form: $\left(\mathfrak{F} = \mathfrak{f}^{\mathfrak{F}}\right)(\mathfrak{o})$, $Pos\left(\mathfrak{F} = \mathfrak{f}^{\mathfrak{F}}\right)(\mathfrak{o})$, $Bel\left(\mathfrak{F} = \mathfrak{f}^{\mathfrak{F}}\right)(\mathfrak{o})$, $Know\left(\mathfrak{F} = \mathfrak{f}^{\mathfrak{F}}\right)(\mathfrak{o})$. For example, in the assumed scenario an expression $Bel\,(\mathsf{Traffic} = Heavy)$ (r4402) represents a following statement of an external language of communication (i.e. in a form in which it would be provided to the external user of the system): *I believe that there is heavy traffic going north into Glenferrie Rd at the crossroad with Riversdale Rd.* It has to be pointed out that the notion of time is completely omitted as we consider only summaries of the most recent finished 15-minutes time slot which is treated here as a present state of agents' discrete time.

When the value $\mathfrak{u} \in \mathfrak{U}_{\mathfrak{F}}$ has been observed by an agent in an object $\mathfrak{o}$, respective evaluations of membership functions for each linguistic concept $\mathfrak{f}^{\mathfrak{F}}$ related to a property $\mathfrak{F}$ are compared against the threshold μ_{min}. If the evaluation overcomes the threshold, i.e. if $\mu^{\mathfrak{F}}_{\mathfrak{f}^{\mathfrak{F}}}(\mathfrak{u}) \geq \mu_{min}$ then $\left(\mathfrak{F} = \mathfrak{f}^{\mathfrak{F}}\right)(\mathfrak{o})$ is considered to be a well-grounded non-modal summary.

4.1 Incomplete Data – Modal Summaries

When a respective piece of data is missing an agent is supposed to perform an internal reasoning to generate a modal summary. In an original approach to the problem pieces of experience were gathered in disjunctive grounding sets and lead to the raise of multiple alternative models representing potential states of the experienced 'gap'. Cardinalities of these sets were strengthening beliefs correlated with respective models. As a result, certain groups of modal statements could be grounded within agent's experience. Although the model has been extended in [14], all terms of the used external language of communication were still disjunctively matched with states of a property perceived by an agent.

In the current setting meaning of concepts represented with fuzzy mappings μ^{φ} can overlap. It has been pointed out in the literature [13] that it is a natural

situation and that it is possible to analyse actual dependencies between fuzzy-linguistic terms used by an agent to describe a certain universe (in this work concepts describe a single property) and build a personal thesaurus. It is also possible to derive new concepts from existing ones using linguistic hedges [16].

There are two basic approaches which can be used to approach the problem of an induction of mental models in an assumed setting. The first approach is to build grounding sets consisting of these pieces of experience which activate the respective concept in relation to the threshold μ_{min}. In such a situation the grounding set is defined as follows:

Definition 1. Crisp Grounding Set. *A grounding set* $A_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$ *related to a linguistic concept* $\mathfrak{f}$ *describing property* $\mathfrak{F}$ *is a set of all those past experiences in which a state* $\mathfrak{u}$ *of the property* $\mathfrak{F}$ *in a particular object* $\mathfrak{o}$ *has been observed by an agent and* $\mu_{\mathfrak{f}}^{\mathfrak{F}}(\mathfrak{u}) \geq \mu_{min}$. *The grounding strength* $G_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$ *(see [8,9]) of such a grounding set is evaluated as its cardinality.*

The second approach – rather than on a grounding set itself – focuses on how its grounding strength is evaluated. The grounding set in this case consists of all pieces of experience which activated the respective concept (the value of assigned fuzzy mapping was non-zero). Formally:

Definition 2. Fuzzy Grounding Set. *A grounding set* $A_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$ *related to a linguistic concept* $\mathfrak{f}$ *describing property* $\mathfrak{F}$ *is a set of all those past experiences in which a state* $\mathfrak{u}$ *of the property* $\mathfrak{F}$ *in a particular object* $\mathfrak{o}$ *has been observed by an agent and* $\mu_{\mathfrak{f}}^{\mathfrak{F}}(\mathfrak{u}) > 0$. *The grounding strength* $G_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$ *is evaluated as a sum of evaluations of fuzzy mappings for all respective pieces of agent's experience located in this grounding set (or – equivalently – over all pieces of experience as the ones which are not located in the grounding set yield a value of 0).*

As in original approaches grounding sets support alternative 'non-overlapping' models, it is natural to compare grounding strengths against each other to see how strongly respective models are induced. It is important to notice that it can be seen as a comparison of a numerical evaluation of the supporting experience against a numerical evaluation of the whole relevant experience. This intuition can be directly carried over to the current model resulting with a following definition of a relative grounding strength:

Definition 3. Relative Grounding Strength – Crisp Grounding Set. *For a given linguistic concept* $\mathfrak{f}$ *describing property* $\mathfrak{F}$ *a relative grounding strength* $\lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$ *of a grounding set* $A_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$ *is given as follows*

$$\lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F}) = \frac{G_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})}{\operatorname{card}\left(\bigcup_{\mathfrak{f}^{\mathfrak{F}} \text{ describing } \mathfrak{F}} A_{\mathfrak{f}^{\mathfrak{F}}}(t, \mathfrak{o}, \mathfrak{F}) \right)}.$$

For a case of fuzzy grounding set (as given by Definition 2) the relative grounding strength can be defined in an analogical way. It is advised to use in such a case in a denominator – instead of a cardinality function – a similar aggregation to the one used to evaluate the grounding strength. The most basic example would be to sum maximal membership values (evaluated over all relevant concepts) for each single piece of experience.

Please take a note that this task is strongly related to a task of aggregation of multiple fuzzy relations – and further – to a task of quantifying over them. When designing a particular method for the evaluation of the relative grounding strength in the final implementation a system designer should keep in mind interpretational properties of different types of fuzzy aggregations [4].

It is possible to limit grounding experience based on a current local context. The idea is to strengthen an influence of these pieces of agent's experience which are in some way similar to the currently perceived state. Relevant methods can be adopted from [12]. We highly encourage the usage of contextual methods for grounding experience determination as they provide mechanisms used to filter relevant experiences based on the external situation.

4.2 Relation of Epistemic Satisfaction

The so-called relation of epistemic satisfaction describes conditions which have to be fulfilled by an agent's knowledge state (summarized in a form of relative grounding strengths) in order to make a modal statement grounded. An existing theory defines a system of modality thresholds $\langle \lambda_{\text{minPos}}, \lambda_{\text{maxPos}}, \lambda_{\text{minBel}}, \lambda_{\text{maxBel}} \rangle$ and discusses in detail dependencies between thresholds which need to be fulfilled in order to guarantee a rational language behaviour of an agent.

In the assumed setting during the process of grounding each concept should be treated separately and grounded separately from the other concepts used to describe the same property. It should be possible to derive dependencies between acceptability of grounding of certain groups of modal statements and relations between concepts present in the agent's personal thesaurus [13], however we leave this part uninvestigated for now.

We propose a following definition for the relation of epistemic satisfaction:

Definition 4. Relation of Epistemic Satisfaction for Modal Statements. *For a given system of modality thresholds, a moment t, an object $\mathfrak{o}$, a property $\mathfrak{P}$ and for each of its values $\mathfrak{f}$: Know$(\mathfrak{P} = \mathfrak{f})(\mathfrak{o})$ is grounded if and only if $\lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{P}) = 1$, Bel$(\mathfrak{P} = \mathfrak{f})(\mathfrak{o})$ is grounded if and only if $\lambda_{\text{minBel}} \leq \lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{P}) < \lambda_{\text{maxBel}}$, and Pos$(\mathfrak{P} = \mathfrak{f})(\mathfrak{o})$ is grounded if and only if $\lambda_{\text{minPos}} \leq \lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{P}) < \lambda_{\text{maxPos}}$.*

To preserve a rational behaviour of the agent the system of modality thresholds needs to fulfil following inequalities [9,14]:

$$\begin{cases} 0 < \lambda_{\text{minPos}} < \lambda_{\text{maxPos}} \leq \lambda_{\text{minBel}} < \lambda_{\text{maxBel}} \leq 1 \\ \lambda_{\text{minPos}} \leq 0.5 < \lambda_{\text{maxPos}}, \lambda_{\text{minBel}} \end{cases}$$

It should be noted that even though there is an order between agent's beliefs (i.e. possibility is weaker than belief and both are weaker than certainty/knowledge), they are grounded exclusively. It means, that to describe dependencies between beliefs one cannot use basic models which often contain equivalents of axioms like **B** ($p \rightarrow \Box\Diamond p$) and **D** ($\Box p \rightarrow \Diamond p$) used in modal logic (for more discussion on interpretation of particular axioms see [6]).

4.3 Algorithm

The process of response generation is initiated when an agent receives a query about a state of the property within a certain object (it is represented as $(\mathfrak{F} = ?)(\mathfrak{o})$). The agent grounds possible answers according to the Algorithm 1. In the assumed scenario, possible answers grounded here are candidates for the final summary of a local traffic state.

Input: State query $(\varphi = ?)(\mathfrak{o})$
Output: Set of grounded statements
foreach *Concept* $\mathfrak{f}$ *describing* $\mathfrak{F}$ **do**
  if *property* $\mathfrak{F}$ *in object* $\mathfrak{o}$ *is observed and is equal to* $\mathfrak{u}$ **then**
    Add $(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to grounded.
  else
    Calculate $\lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F})$.;
    if $\lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F}) = 1$ **then**
      Add $Know(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to grounded.;
    end
    if $\lambda_{\text{minBel}} \leq \lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F}) < \lambda_{\text{maxBel}}$ **then**
      Add $Bel(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to grounded.;
    end
    if $\lambda_{\text{minPos}} \leq \lambda_{\mathfrak{f}}(t, \mathfrak{o}, \mathfrak{F}) < \lambda_{\text{maxPos}}$ **then**
      Add $Pos(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to grounded.;
    end
  end
end

Algorithm 1. Generation of grounded statements

The algorithm generates as an output a set of all grounded formulas, i.e. a set of all semantically fitting summaries. The final task which needs to be performed in order to provide a single summary to the user is to make a choice of the final summary. The usual methods from the NLG literature – involving such quality measures as the length of the summary, its informativeness and precision – can be adopted to complete this task.

5 Aggregation of Summaries

An aggregation of summaries about values of a property φ is performed according to predefined patterns. Patterns are simply sets of objects with an assigned textual description of the whole pattern. Within a traffic scenario it allows to model

following types of traffic aggregations: areas at different scale, e.g. Boroondara, East Melbourne; precise routes; complex patterns, e.g. traffic leaving MCG.

An aggregated description (a summary) for a given pattern $\mathfrak{P} = \{\mathfrak{o}_1, \mathfrak{o}_2, \ldots, \mathfrak{o}_N\}$ is evaluated based on summaries generated for its components. Components can be single links (objects) or subpatterns (sets of objects). Each object $\mathfrak{o}$ of the world is globally correlated with an assigned numerical weight $w(\mathfrak{o})$ representing its importance, e.g., in an assumed traffic scenario weights are highly correlated with an average flow through the link. A weight of the pattern is equal to a sum of weights of objects included in it.

It allows for a straightforward propagation of summaries of regions although one should be wary of the properties of systems with multi-level aggregations. This aggregation may yield different results depending on the number of decomposition levels as it does not conserve any additive property apart from aforementioned weights.

A major problem is – yet again – caused by the fact that concepts describing a certain property have an overlapping meaning. It leads to a situation where concepts need to be reinterpreted; in a sense that the presence of a certain concept in a local summary will support (with the varying strength) all related concepts as candidates to be used in the aggregated summary.

A numerical strength of this induction is technically a similarity of two fuzzy mappings. We will denote it as $\sigma(\mathfrak{f}_1, \mathfrak{f}_2)$. We expect non-symmetric similarity functions (e.g. recall and precision similarity indices) to be used in final implementations as it seems unwanted that a broader term induces strongly all narrower terms.

The task of an aggregation in an assumed system is to generate all relevant summaries given a set of 'local' summaries, weights of objects (or patterns) related to these summaries, a set of fuzzy mappings $\mu_{\mathfrak{f}}$ representing meaning of respective language concepts $\mathfrak{f}$ related to a particular property φ (object-independent mappings can be used at the aggregation level), a system of modality thresholds, and the similarity function σ. The basis for an aggregation is an evaluation of a strength $w(\mathfrak{P}, \mathfrak{f})$ with which each particular concept $\mathfrak{f}$ is induced by a set of local summaries within a pattern $\mathfrak{P}$ (the notion of time is omitted).

Definition 5. Induced Strength of Concept. *For a given a set S of 'local' summaries s, weights of objects $\mathfrak{o}_s$ (or patterns) related to these summaries, a set of fuzzy mappings $\mu_{\mathfrak{f}}$ representing meaning of respective language concepts $\mathfrak{f}$ related to a particular property φ (object-independent mappings can be used here), a system of modality thresholds, and the similarity function σ an induced strength of a concept $\mathfrak{f}$ is given as: $w(\mathfrak{P}, \mathfrak{f}) = \sum_{s \in S} \left(\hat{w}(\mathfrak{o}_s) \cdot \sigma(\mathfrak{f}, \mathfrak{f}_s) \right)$, where $\hat{w}(\mathfrak{o}_s)$ is modified weight equal to: $w(\mathfrak{o}_s)$ if no modality or a modality Know is present in the local summary s; $\lambda_{\mathrm{minBel}} \cdot w(\mathfrak{o}_s)$ if a modality Bel is present in the local summary s; $\lambda_{\mathrm{minPos}} \cdot w(\mathfrak{o}_s)$ if a modality Pos is present in the local summary s.*

Modified weights are used to reflect a lowered impact on the final outcome of those local summaries which come from an incomplete knowledge. In above formulas lowest modality thresholds for respective modal operators are used to

modify their impact. Used values depend on a particular application however they should be chosen from intervals related to respective modalities.

Unlike in [14], an execution of the aggregation procedure described by Algorithm 2 does not result in a single language statement. This is again related to the fact that multiple concepts can be overlapping in terms of their meaning and therefore multiple summaries can be semantically fitting. Analogically to the process of generation of local summaries, the final textual summary should be chosen based on criteria adopted from the NLG literature.

Input: State query $(\varphi =?)(\mathfrak{o})$
Output: Set of aggregated summaries
foreach *concept* $\mathfrak{f}$ *describing property* $\mathfrak{F}$ **do**
  Calculate $w(\mathfrak{P}, \mathfrak{f})$;
  if $w(\mathfrak{P}, \mathfrak{f}) = 1$ **then**
    Add $(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to aggregated summaries;
    Add $Know(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to aggregated summaries;
  end
  if $\lambda_{\text{minBel}} \leq w(\mathfrak{P}, \mathfrak{f}) < \lambda_{\text{maxBel}}$ **then**
    Add $Bel(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to aggregated summaries;
  end
  if $\lambda_{\text{minPos}} \leq w(\mathfrak{P}, \mathfrak{f}) < \lambda_{\text{maxPos}}$ **then**
    Add $Pos(\mathfrak{F} = \mathfrak{f})(\mathfrak{o})$ to aggregated summaries;
  end
end

Algorithm 2. Generation of aggregated summaries

An example of an aggregated summary: *I believe that there is heavy traffic from Hawthorn Bridge to Eastern Fwy via Church St, High St, Doncaster Rd.*

6 Discussion

The main advantage of providing messages using autoepistemic operators of possibility, belief, and knowledge is an information conveyed to the user about a fact that the obtained summary does not directly come from an empirical measurement of a described object, but is a result of a reasoning procedure. In typical systems entries containing incomplete data are either removed or filled using imputation procedures (see [10] for a discussion on approaches to dealing with missing data). It should be pointed out that the proposed method does not take into consideration an order of links within the pattern making it potentially more suitable for aggregating descriptions of traffic within regions rather then related to a specific route.

The theory of grounding provides a set of restrictions for the system of modality thresholds $\langle \lambda_{\text{minPos}}, \lambda_{\text{maxPos}}, \lambda_{\text{minBel}}, \lambda_{\text{maxBel}} \rangle$, but it does not specify final values to be used in a specific implementation. It is possible to incorporate an

adaptation procedure into an agent calibrate the system based on users' feedback. This would also address the fact, that different communities may have different understanding of used terms of natural language. This feedback can be also used to evaluate the preferences of terms' usage in textual summaries[15].

Automated modification of modality thresholds are already partially analysed in the literature. A language game mechanism used in [11] works based on messages exchanged about a common context shared by multiple agents. The mechanism causes the convergence of systems of modality thresholds for groups of agents interacting with each other.

7 Conclusions and Further Work

We outlined a distributed multi-agent system capable of a generation of textual summaries of a partially observed environment. The system provides summaries involving auto-epistemic operators of possibility, belief, and knowledge in situations where data is incomplete. In such cases the internal reasoning procedure is performed in order to evaluate the summary. Summaries are generated in a natural language form in order to make them comprehensible by an external user of the system.

The proposed system is based on numerical data about traffic volumes provided systematically over time. This enhances the previous approach presented in [14] which used predefined direct bijective mapping between observed values of the properties and linguistic concepts. Existing systems providing traffic information use simple indicators (e.g. colours) based on the local counts and do not reflect the quality of observation data (e.g. missing data) in their summaries.

We have presented an algorithm derived from the theory of grounding of modal statements in artificial cognitive agents. The algorithm lets an agent use its private observational history in order to generate a set of fitting summaries in situations where a respective piece of data is missing.

Summaries can be easily converted into a textual form using provided patterns, however we have not specified algorithm for the choice of the final summary. It is not directly a problem, as there are methods in NLG literature for an evaluation of summaries, however this should be further investigated as they do not address situations where autoepistemic operators are used.

An algorithm for an aggregation of local summaries according to predefined patterns provided by a domain expert is proposed. It can result in a set of multiple alternative summaries because concepts used to describe a property can have overlapping meaning. The same method for the choice of the final summary should be used as in the case of a generation of local summary.

What is particularly interesting is how relations between meaning of concepts should influence the relation of epistemic satisfaction. The constraints defined within the relation should be reinvestigated in the future based on sets of statements which can or cannot be grounded at the same time with respect to semantic relation present in the agent's thesaurus.

Acknowledgements. This research were conducted under fellowship co-financed by European Union within European Social Fund. Research reported in Section 5 was supported by Grant no. N N519 444939 funded by Polish Ministry of Science and Higher Education (2010-2013).

References

1. Bell, V.A., Johnson-Laird, P.N.: A model theory of modal reasoning. Cognitive Science 22(1), 25–51 (1998)
2. Dennett, D.: True believers: The intentional strategy and why it works. Language and Thought, 258 (2005)
3. Glöckner, I.: Optimal Selection of Proportional Bounding Quantifiers in Linguistic Data Summarization. In: Soft Methods for Integrated Uncertainty Modelling, pp. 173–181 (2006)
4. Glöckner, I., Knoll, A.: Fuzzy Quantifiers: A Natural Language Technique for Data Fusion. In: Proceedings of the Fourth International Conference on Information Fusion (Fusion 2001) (2001)
5. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena 42(1-3), 335–346 (1990)
6. Huber, F., Schmidt-Petri, C. (eds.): Degrees of Belief. Synthese Library, vol. 342. Springer, UK (2009)
7. Kacprzyk, J., Wilbik, A., Zadrozny, S.: Linguistic summarization of time series using a fuzzy quantifier driven aggregation. Fuzzy Sets and Systems 159(12), 1485–1499 (2008)
8. Katarzyniak, R.P.: The language grounding problem and its relation to the internal structure of cognitive agents. Journal of Universal Computer Science 11(2), 357–374 (2005)
9. Katarzyniak, R.P.: On some properties of grounding uniform sets of modal conjunctions. Journal of Intelligent and Fuzzy Systems 17(3), 209–218 (2006)
10. Little, R.J.A., Rubin, D.B.: Statistical analysis with missing data. John Wiley & Sons, Inc., New York (1986)
11. Lorkiewicz, W., Popek, G., Katarzyniak, R., Kowalczyk, R.: Aligning Simple Modalities in Multi-agent System. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 70–79. Springer, Heidelberg (2011)
12. Popek, G.: Strength of Formula Grounding and Context-Dependent Strategies for Grounding Experience Determination. In: Nguyen, N.T., Kolaczek, G., Gabrys, B. (eds.) Challenging Problems of Science, Computer Science, Knowledge Processing and Reasoning for Information Society, pp. 21–38. APH EXIT (2008)
13. Popek, G., Katarzyniak, R.: Agent-based generation of personal thesaurus. In: Proceedings of First Asian Conference on Intelligent Information and Database Systems, ACIIDS 2009, Vietnam, pp. 179–182. IEEE Computer Society (2009)
14. Popek, G., Kowalczyk, R., Katarzyniak, R.P.: Generating Descriptions of Incomplete City-Traffic States with Agents. In: Wang, Y., Li, T. (eds.) ISKE 2011. AISC, vol. 122, pp. 105–114. Springer, Heidelberg (2011)
15. Reiter, E., Sripada, S., Hunter, J., Yu, J., Davy, I.: Choosing words in computer-generated weather forecasts. Artificial Intelligence 167(1-2), 137–169 (2005); Connecting Language to the World
16. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning - i, iii. Inf. Sci. 8(3), 9(1), 199–249, 43–80 (1975)

Author Index