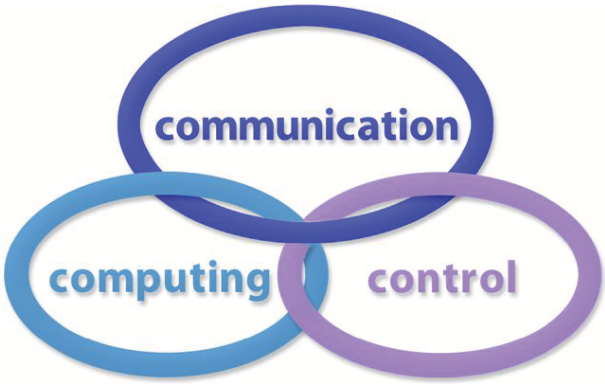
INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2015 Volume: 10 Issue: 5 (October)

This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).

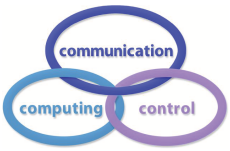
CCC Publications - Agora University Editing House

CCC Publications

http://univagora.ro/jour/index.php/ijccc/

BRIEF DESCRIPTION OF JOURNAL

Publication Name: International Journal of Computers Communications & Control.
Acronym: IJCCC; **Starting year of IJCCC:** 2006.
Abbreviated Journal Title in JCR: INT J COMPUT COMMUN.
International Standard Serial Number: ISSN 1841-9836.
Publisher: CCC Publications - Agora University of Oradea.
Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).
Founders of IJCCC: Ioan DZITAC, Florin Gheorghe FILIP and Mişu-Jan MANOLESCU.
Logo:



Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Thomson Reuters and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.
- Journal Citation Reports (JCR - Science Edition), IF = 0.746 (JCR2014).
  Subject Category:
  - Automation & Control Systems: Q4 (47 of 58);
  - Computer Science, Information Systems: Q3 (96 of 139).
- Since 2008, 3(1), IJCCC is covered in Scopus, SJR2014 = 0.450, H index = 15.
  Subject Category:
  - Computational Theory and Mathematics: Q3;
  - Computer Networks and Communications: Q2;
  - Computer Science Applications: Q2.
- Since 2007, 2(1), IJCCC is covered in EBSCO.

Focus & Scope: International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computer and control from the universities, research units and industry.

To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of original scientific papers that focus on the integration of the 3 "C" (Computing, Communication, Control).

In particular the following topics are expected to be addressed by authors:

- Integrated solutions in computer-based control and communications;
- Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence);
- Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

Copyright © 2006-2015 by CCC Publications - Agora University

IJCCC EDITORIAL TEAM

Editor-in-Chief: Florin-Gheorghe FILIP
Member of the Romanian Academy
Romanian Academy, 125, Calea Victoriei
010071 Bucharest-1, Romania, ffilip@acad.ro

Associate Editor-in-Chief: Ioan DZITAC
Aurel Vlaicu University of Arad, Romania
St. Elena Dragoi, 2, 310330 Arad, Romania
ioan.dzitac@uav.ro
&
Agora University of Oradea, Romania
Piata Tineretului, 8, 410526 Oradea, Romania
rector@univagora.ro

Managing Editor: Mişu-Jan MANOLESCU
Agora University of Oradea, Romania
Piata Tineretului, 8, 410526 Oradea, Romania
mmj@univagora.ro

Executive Editor: Răzvan ANDONIE
Central Washington University, U.S.A.
400 East University Way, Ellensburg, WA 98926, USA
andonie@cwu.edu

Reviewing Editor: Horea OROS
University of Oradea, Romania
St. Universitatii 1, 410087, Oradea, Romania
horos@uoradea.ro

Layout Editor: Dan BENTA
Agora University of Oradea, Romania
Piata Tineretului, 8, 410526 Oradea, Romania
dan.benta@univagora.ro

Technical Secretary

Simona DZITAC
R & D Agora, Romania
rd.agora@univagora.ro

Emma VALEANU
R & D Agora, Romania
evaleanu@univagora.ro

Editorial Address:
Agora University/ R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Journal website: http://univagora.ro/jour/index.php/ijccc/

IJCCC EDITORIAL BOARD MEMBERS

Luiz F. Autran M. Gomes
Ibmec, Rio de Janeiro, Brasil
Av. Presidente Wilson, 118
autran@ibmecrj.br

Boldur E. Bărbat
Sibiu, Romania
bbarbat@gmail.com

Pierre Borne
Ecole Centrale de Lille, France
Villeneuve d'Ascq Cedex, F 59651
p.borne@ec-lille.fr

Ioan Buciu
University of Oradea
Universitatii, 1, Oradea, Romania
ibuciu@uoradea.ro

Hariton-Nicolae Costin
Faculty of Medical Bioengineering
Univ. of Medicine and Pharmacy, Iaşi
St. Universitatii No.16, 6600 Iaşi, Romania
hcostin@iit.tuiasi.ro

Petre Dini
Concordia University
Montreal, Canada
pdini@cisco.com

Antonio Di Nola
Dept. of Math. and Information Sci.
Università degli Studi di Salerno
Via Ponte Don Melillo, 84084 Fisciano, Italy
dinola@cds.unina.it

Yezid Donoso
Universidad de los Andes
Cra. 1 Este No. 19A-40
Bogota, Colombia, South America
ydonoso@uniandes.edu.co

Ömer Egecioglu
Department of Computer Science
University of California
Santa Barbara, CA 93106-5110, U.S.A.
omer@cs.ucsb.edu

Janos Fodor
Óbuda University
Budapest, Hungary
fodor@uni-obuda.hu

Constantin Gaindric
Institute of Mathematics of
Moldavian Academy of Sciences
Kishinev, 277028, Academiei 5
Moldova, Republic of
gaindric@math.md

Xiao-Shan Gao
Acad. of Math. and System Sciences
Academia Sinica
Beijing 100080, China
xgao@mmrc.iss.ac.cn

Kaoru Hirota
Hirota Lab. Dept. C.I. & S.S.
Tokyo Institute of Technology
G3-49,4259 Nagatsuta, Japan
hirota@hrt.dis.titech.ac.jp

Gang Kou
School of Business Administration
SWUFE
Chengdu, 611130, China
kougang@swufe.edu.cn

George Metakides
University of Patras
Patras 26 504, Greece
george@metakides.net

Shimon Y. Nof
School of Industrial Engineering
Purdue University
Grissom Hall, West Lafayette, IN 47907
U.S.A.
nof@purdue.edu

Stephan Olariu
Department of Computer Science
Old Dominion University
Norfolk, VA 23529-0162, U.S.A.
olariu@cs.odu.edu

Gheorghe Păun
Institute of Math. of Romanian Academy
Bucharest, PO Box 1-764, Romania
gpaun@us.es

Mario de J. Pérez Jiménez
Dept. of CS and Artificial Intelligence
University of Seville, Sevilla,
Avda. Reina Mercedes s/n, 41012, Spain
marper@us.es

Dana Petcu
Computer Science Department
Western University of Timisoara
V.Parvan 4, 300223 Timisoara, Romania
petcu@info.uvt.ro

Radu Popescu-Zeletin
Fraunhofer Institute for Open
Communication Systems
Technical University Berlin, Germany
rpz@cs.tu-berlin.de

Imre J. Rudas
Óbuda University
Budapest, Hungary
rudas@bmf.hu

Yong Shi
School of Management
Chinese Academy of Sciences
Beijing 100190, China &
University of Nebraska at Omaha
Omaha, NE 68182, U.S.A.
yshi@gucas.ac.cn, yshi@unomaha.edu

Athanasios D. Styliadis
University of Kavala
Institute of Technology
65404 Kavala, Greece
styliadis@teikav.edu.gr

Gheorghe Tecuci
Learning Agents Center
George Mason University
U.S.A.
University Drive 4440, Fairfax VA
tecuci@gmu.edu

Horia-Nicolai Teodorescu
Faculty of Electronics and
Telecommunications
Technical University "Gh. Asachi" Iasi
Iasi, Bd. Carol I 11, 700506, Romania
hteodor@etc.tuiasi.ro

Dan Tufiş
Research Institute for Artificial Intelligence
of the Romanian Academy
Bucharest, "13 Septembrie" 13, 050711,
Romania
tufis@racai.ro

Lotfi A. Zadeh
Director,
Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
University of California Berkeley,
Berkeley, CA 94720-1776
U.S.A.
zadeh@eecs.berkeley.edu

DATA FOR SUBSCRIBERS
Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Fiscal code: 24747462
Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526
Bank: BANCA COMERCIALA FEROVIARA S.A. ORADEA
Bank address: P-ta Unirii Nr. 8, Oradea, Bihor, România
IBAN Account for EURO: RO50BFER248000014038EU01
SWIFT CODE (eq.BIC): BFER

INTERNATIONAL CONFERENCE on ORIENTAL THINKING and FUZZY LOGIC
(Celebration of the 50th Anniversary of Fuzzy Sets in Big Data Era)
Dalian, China, August17-20, 2015

Organizing Institutes
Fuzzy Information & Engineering Branch, Operation Research Society of China (FIEB, ORSC).
Technical Committee on Business Intelligence,Chinese Society for Management Modernization (TCBI, CSMM).
Extenics Society, Chinese Association for Artificial Intelligence (ES, CAAI).
Liaoning Technical University, China (LNTU).
Dalian University, China.
Dalian Polytechnic University, China.
Beijing Cazl Technology Service Co., Ltd.

In order to commemorate the 50th anniversary of Fuzzy Sets and Systems and cope with the challenges of Big Data Tide, we hold the International Conference on Oriental Thinking and Fuzzy Logic (Celebration of the 50th Anniversary of Fuzzy Sets in Big Data Era) during August 17-20th, 2015 in the famous and beautiful coastal city, Dalian, China. This conference is a united conference of the 8th International Conference on Fuzzy Information and Engineering held by FIEB, ORSC, the 2nd Symposium on Big Data and Data Science held by TCBI, CSMM, Forum on Extenics and Innovation Methods held by ES, CAAI and the 3rd Workshop on Intelligence Engineering and Mathematics held by LNTU.

The honorary chair for this conference will be the founder of Fuzzy Sets Theory, Prof. Lotfi A. Zadeh. His friend, Prof. Peizhuang Wang, who is the disseminator of fuzzy mathematics and its applications in China, will be the chair of the Advisory committee.

Prof. L. A. Zadeh will deliver a video speech for the conference. This conference will review the significant contributions which Fuzzy Sets Theory has made to the Data Science, and will welcome new theory and methods. Prof. Peizhuang Wang will introduce the Factor Space Theory created by Chinese scholars in his keynote speech.

Taking this opportunity, we sincerely welcome our colleagues worldwide to join us for this conference and share the latest and excited progress.

International Program Committee: *Chair*: Yong Shi;
Members: Bingyuan Cao (China), Shuili Chen (China), Yixiang Chen (China), Yongyi Chen (China), Liya Ding (China), Ioan Dzitac (Romania), Enrique Herrera Viedma (Spain), Jiali Feng (China), Qiang Fu (China), Minghu Ha (China), Liyan Han (China), Qing He (China), Baoqing Hu (China), Weidong Hu (China), Chongfu Huang(China), Gang Kou (China), Jianping Li (China), Hongxing Li (China), Taifu Li (China), Chenguang Lu(China), H. C. Lui (USA), Shengquan Ma (China), Honghai Mi (China), Duoqian Miao (China), Zhiwen Mo (China), He Ouying (China), Xiantu Peng (USA), Zhiquan Qi (China), Yan Shi (China), Zhenming Song(China), Yingjie Tian (China), Shaocheng Tong (China), LidongWang (China), Xizhao Wang(China), Xueping Wang (China), Ciyuan Xiao (China), Xiangjun Xie (China), Yang Xu (China), Fangqu Xu (China), Xiaoquan Xu (China), Zeshui Xu (China), Guojun Yan (China), Chunyan Yang (China), Liangzhong Yi (China), Ye Yin(China), Mingsheng Ying (China), Fusheng Yu (China), Xuehai Yuan (China), Wenyi Zeng (China), Bo Zhang (China), Chengyi Zhang (China), Changqing Zhang (USA), Qiang Zhang, Bin Zhao.

Honorary Chair: Lotfi A. Zadeh.

Conference Chairs: Zengliang Liu and Peizhuang Wang.

Organizing Committee Chairs: Jiren Wang and Kaiqi Zou.

7th IFAC CONFERENCE on MANAGEMENT and CONTROL of PRODUCTION and LOGISTICS (MCPL 2016)

Bremen, Germany, February 22-24, 2016

http://www.mcpl2016.logdynamics.de

The 7th IFAC Conference on Management and Control of Production and Logistics (MCPL 2016) will be held in Bremen (Germany) from 22nd to 24th of February 2016 jointly with the 5th International Conference on Dynamics in Logistics (LDIC 2016) and several satellite events. Accepted papers will be published in the proceedings of the event using the open-access IFAC-PapersOnLine.

Scope of the Conference

The conference, sponsored by IFAC, aims to bring together researchers and practitioners from different areas of production and logistics with a special focus on the engineering side of management and control of such systems. The central idea is to establish a common ground in order to promote a synergy among different disciplines for exploring new solutions for complex scientific and technical challenges. The objectives of the conference are to provide high quality research and professional interactions for the advancement of science, technology and fellowship. It also provides the participants an opportunity to present their research papers and experience reports, and to take part in open discussions.

Topics

Topics of interest include, but are not limited to: *Modeling and Simulation; Decision-Support Systems: Concepts, Methods and Algorithms; Discrete Event Systems; Cyber-physical Production and Logistic Systems; Probabilistic and Statistical Modeling; Production Planning and Scheduling; Operational Research Applications; Control Methods and Concepts; Robotics and Man-Machine Interaction; Factory Automation; Intelligent Manufacturing Systems; Advanced Process Control and Wireless Automation; Lean Six Sigma: Enterprise, Manufacturing and Healthcare; ERP and Inventory Control; Management of Organizations; Supply Chain and Green Supply Chain Management; Urban Freight Distribution and City Logistics; Information Technology in Production, Logistics and Management; Humanitarian Logistics; Socio-technical and Cognitive Aspects in Manufacturing and Logistics; Quality Management Systems and Performance Indicators.*

MCPL

The IFAC MCPL 2016 is the 7th in a very successful series of events, previously held in Fortaleza (Brazil), Campinas (Brazil), Grenoble (France), Santiago (Chile), Sibiu (Romania) and Coimbra (Portugal). This seventh edition will be organized by the BIBA Bremer Institut für Produktion und Logistik, one of the most important research centers for Production and Logistic Systems in Europe. The conference will be held in the Hanseatic City Bremen on the banks of the River Weser, one of the biggest logistics hubs in Europe. The city combines high-tech and picturesque narrow streets built in centuries past, and its rich heritage of history is greatly cherished and lovingly preserved. Bremen is the only city in Germany to have an airport ten minutes away from the city. Destinations like London, Madrid, Munich or Vienna can be reached easily by plane. Submissions The conference submission tool "PaperCept" will open in July 2015. Papers submitted to the main conference must contain original research and should not exceed six pages. Simultaneous submission to other conferences with proceedings or submission of material that has already been published elsewhere is not allowed.

Program Chairs: Jurgen Pannek (pan@biba.uni-bremen.de)
and Florin Gheorghe Filip (ffilip@acad.ro).

6th INTERNATIONAL CONFERENCE on COMPUTERS, COMMUNICATIONS and CONTROL (ICCCC 2016)

Hotel President, Baile Felix, Oradea, Romania, May 10-14, 2016

Organized by Agora University of Oradea,

under the aegis of Romanian Academy: Information Science and Technology Section.

http://univagora.ro/en/icccc2016/

Scope and Topics

The International Conference on Computers Communications and Control (ICCCC) has been founded in 2006 by I. Dzitac, F.G. Filip and M.-J. Manolescu and organized every even year by Agora University of Oradea, under the aegis of the Information Science and Technology Section of Romanian Academy and IEEE - Romania Section.

The goal of this conference is to bring together international researchers, scientists in academia and industry to present and discuss in a friendly environment their latest research findings on a broad array of topics in computer networking and control.

The Program Committee is soliciting paper describing original, previously unpublished, completed research, not currently under review by another conference or journal, addressing state-of-the-art research and development in all areas related to computer networking and control.

In particular the following topics are expected to be addressed by authors:

1) Integrated solutions in computer-based control and communications;
2) Network Optimization and Security;
3) Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence);
4) Data Mining and Intelligent Knowledge Management;
5) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies);
6) Membrane Computing - Theory and Applications;
7) Stereovision Based Perception for Autonomous Mobile Systems and Advanced Driving Assistance.

Special Sessions

Special Session 1: Network Optimization and Security, Organizer and Chair: Yezid DONOSO, University de los Andes, Colombia;

Special Session 2: Data Mining and Intelligent Knowledge Management, Organizers and Chairs: Gang KOU and Yi PENG, China;

Special Session 3: Computational Intelligence Methods, Organizers and Chairs: Razvan ANDONIE and Donald DAVENDRA, Central Washington University, USA;

Special Session 4: Advanced Decision Support Systems, Organizer and Chair: Marius CIOCA, Lucian Blaga University of Sibiu, Romania;

Special Session 5: Fuzzy Control, Modeling and Optimization, Organizer and Chair: Radu-Emil PRECUP, Politehnica University of Timisoara, Romania;

Special Session 6: Membrane Computing - Theory and Aplications, Organizers and Chairs: Marian GHEORGHE (UK) and Florentin IPATE (Romania);

Special Session 7: Stereovision Based Perception for Autonomous Mobile Systems and Advanced Driving Assistance, Organizer and Chair: Sergiu NEDEVSCHI, Technical University of Cluj-Napoca, Romania.

Keynote Speakers: Enrique HERRA VIEDMA(Spain), Zenonas TURSKIS (Lithuania), Gang KOU (China).

Conference Chairs: Ioan DZITAC, Florin Gheorghe FILIP and Misu-Jan MANOLESCU.

International Journal of Computers Communications & Control
ISSN 1841-9836, Volume 10, Issue 5, October, 2015.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):617-626, October, 2015.

Impact of Membrane Computing and P Systems in ISI WoS. Celebrating the 65th Birthday of Gheorghe Păun

I. Dzitac

Ioan Dzitac
1. Aurel Vlaicu University of Arad
Romania, 310330 Arad, Elena Dragoi, 2
ioan.dzitac@uav.ro
2. Agora University of Oradea
Romania, 410526 Oradea, Piata Tineretului, 8
rector@univagora.ro

"Congratulations, G. Paun!
Since 2000, you have been cited ... times for your article...
This means that the number of citations your article received places it in the top 0.5% within its field according to ESI.
Your work is highly influential, and is making a significant impact among your colleagues in your field of study.
Congratulations on your extraordinary career accomplishment!"
Essential Science Indicators (ESI), ISI Thomson, 2009.

Gheorghe Păun
(b. Dec. 6, 1950)

Abstract: Membrane Computing is a branch of Computer Science initiated by Gheorghe Păun in 1998, in a technical report of Turku Centre for Computer Science published as a journal paper ("Computing with Membranes" in Journal of Computer and System Sciences) in 2000. Membrane systems, as Gheorghe Păun called the models he has introduced, are known nowadays as "P Systems" (with the letter P coming from the initial of the name of this research area "father").
This note is an overview of the impact in ISI WoS of Gheorghe Păun's works, focused on Membrane Computing and P Systems field, on the occasion of his 65th birthday anniversary.
Keywords: Membrane Computing, P Systems, ISI Web of Science(WoS), scientific impact, h-index.

1 Introduction: Brief Bio-Sketch of Gheorghe Păun

"Gheorghe Păun is an example of a person affirming his own existence by writing. He is a prolific writer with a huge number of papers: tens of scientific books, hundreds of articles, several novels, poems, and books on games." (G. Ciobanu, [1]).

Copyright © 2006-2015 by CCC Publications

Gheorghe Păun (b. December 6, 1950, Romania) is a Romanian mathematician and theoretical computer scientist. He is also a writer and promoter of culture (editor-in-chief of a monthly cultural magazine "Curtea de La Argeş") [9].

G. Păun graduated the Faculty of Mathematics of the Bucharest University in 1974 and got his PhD at the same faculty in 1977 (under the supervision of Solomon Marcus). He has won many scholarships, in Germany, Finland, The Netherlands, Spain, etc.

Presently he is a senior researcher at the Institute of Mathematics of the Romanian Academy, Bucharest, and visiting researcher at Sevilla University, Spain.
Since 1997 he is a Corresponding Member, and since 2012 is a Full Member of the Romanian Academy. Since 2006 he is a member of Academia Europaea.

His main research fields are formal language theory (regulated rewriting, contextual grammars, grammar systems), automata theory, combinatorics on words, computational linguistics, DNA computing, membrane computing. He has (co)authored and (co)edited more than 60 books in these areas, and he has (co)authored more than 500 research papers.

In the last two decades he has visited many universities from Europe, USA, Canada, Japan, China, also participating to many international conferences, several times as an invited/keynote speaker. He is a member of the editorial board of numerous computer science journals and professional associations.

2 G. Păun and Membrane Computing

Gheorghe Păun is the founder of the Membrane Computing area of Computer Science, a research field inspired from the bio-chemistry of the cell (a more suitable name can be "Cell Computing"). The membrane systems, introduced in 1998 in [3] and disseminated in 2000 in [4], are usually called "P Systems". Most cited of all G. Păun's works is "Computing with Membranes" published in 2000, [4], having more than 850 citations (the second position of all cited papers of Journal of Computer and System Sciences). These 850 citing papers have also in turn over 4,900 citations, [7].

In February 2003, Gheorghe Păun answered a few questions on the occasion of ISI nomination of [4] as a fast breaking paper in the field of Computer Science, [8]:

"**ST: Why do you think your paper is highly cited?**

It is both based on several "classic" theoretical computer science tools/techniques (grammars, automata, Lindenmayer systems, regulated rewriting, grammar systems), and related to rather "modern" branches of computer science (molecular computing in general, DNA computing in particular); it is highly interdisciplinary (it proposes computation models based on the structure and the functioning of the living cell). The model is very versatile; many variants were proposed, biologically or mathematically motivated. Most of these variants have attractive computer science features: universality and/or computational efficiency; they also promise to have relevance from a biological point of view, as algorithmic models of the cell as a whole (or of other cell-like systems).

ST: Does it describe a new discovery or a new methodology that's useful to others?

It proposes a new branch of natural computing: a distributed, parallel, and nondeterministic computing model, processing multisets in a cell-like or tissue-like compartmental structure, thus extending the field of molecular computing. Topics such as communication, synchronization, (maximal) parallelism, space-time trade-off, complexity, localization, etc. can find a natural framework to be dealt with. This is also a (reductionistic) global algorithmic model of the living cell, a kind of model the biologists are waiting for in the near future.

ST: What were some of the circumstances that led you to do this research?

The research was done in the prolongation of several years of work in DNA computing, which, in their turn, have continued many years of work in formal language theory. Very specifically, the paper was written in the framework of a project supported by The Academy of Finland, during a stage in Turku Centre for Computer Science, in the scientifically very "hot" group of Arto Salomaa.

ST: Could you summarize the significance of your paper in layman's terms?

The goal is to abstract a computability model from the structure and the functioning of the living cell. In short, in the compartments determined by a "membrane structure" (a cell-like hierarchical arrangement of membranes) one places multisets of "objects" (corresponding to the chemicals swimming in solutions) and "evolution rules" (corresponding to chemical reactions). The rules are applied in a maximally parallel (each object which can evolve by a rule from the same compartment should evolve) nondeterministic (the objects and the rules are randomly chosen) manner. The objects can also pass through membranes (one can say that they are "communicated" from one compartment to another one), the membranes can be dissolved, divided, created. The application of rules and the communication of objects can be controlled in various ways, either biologically or computer science motivated. A sequence of "transitions" from a configuration of the system to another configuration is called a "computation" and with a halting computation one associates a result, e.g., in the form of the multiset of objects present in a specified output membrane at the end of the computation. Several variants were considered. Most of them are computationally universal (they can compute all Turing computable vectors of natural numbers); this holds true even in certain cases where the computation is done by communication only (no object evolves, but only changes the compartments where it is placed). When an enhanced parallelism is available (e.g., via membrane division) polynomial solutions to NP-complete problems can be obtained by a space-time trade-off. No biological implementation was reported, but there are several software implementations, as well as several (preliminary) applications, mainly in simulating biological phenomena. Details can be found at the web address http://ppage.psystems.eu, as well as in the monograph Gheorghe Păun, Membrane Computing. An Introduction, Springer-Verlag, Berlin, 2002."

3 Collaborators and Followers of Gheorghe Păun

Publications of G. Păun have been cited in more than 15,000 international works (co)authored by more than 1700 computer scientists as: Y. Matyiasevich, A. Salomaa, G. Rozenberg, A. Ehrenfeucht, M. Hagyia, J. Kral, J. Berstel, J. Beauquier, B. Rozoy, J. Dassow, M. Novotny, R. Freund, P.R.J. Asveld, I.M. Havel, R. Siromoney, K.G. Subramanian, F. Urbanek, E. Csuhaj-Varju, J. Kelemen, A. Kelemenova, J. Hromkovic, M. Latteux, M. Clerbout, E. Makinen, N. Nirmal, H.C.M. Kleijn, Al. Meduna, CC. Squier, Z. Tuza, X.M. Nguyen, F.J. Brandenburg, J. Kari, V. Niemi, H. Fernau, J. Shallit, D. Watjen, A. Lepisto, A. Carpi, T. Harju, M. Jantzen, H. Bordihn, D. Raz, J. Honkala, T. Yokomori, G. Mauri, M. Katsura, V. Manca, K. Krithivasan, J. Reif, M. Margenstern, J. Goldstine, Y. Rogozhin, H. Tanaka, M. Conrad, S. Crespi Reghizzi, M.J. Pérez-Jiménez, D. Wotschke, A. Obtulowicz, M. Holcombe, O. Ibarra, O. Ecegioglu, C. Teuscher, J. Karhumaki, L. Cardelli, E. Shapiro, J. Wiedermann, E. Moriya, C. Rossello, K. Ueda, S.G. Akl etc.

G. Păun is an very influential author, a founder of several research groups in membrane computing. Many researchers were attracted to investigate molecular computation models by G. Păun, from countries as: Romania, Republic of Moldova, Hungary, Austria, Spain, The Netherlands, Finland, China etc.

Many well know researchers were involved in studying open problems proposed G. Păun: J. Berstel, L. Boasson, J. Beauquier, B. Rozoy, P.R.J. Asveld, F. Urbanek, J. Dassow, M. Latteux, M. Clerbout, C.C. Squier, Z. Tuza, J. Cassaigne, S. Schwer, P. Seebold, E. Makinen, F.J. Brandenburg, A. Lepisto, A. Carpi, J. Kari, V. Niemi, D. Hauschildt, M. Jantzen, D. Raz, D. Pixton, M.J. Pérez-Jiménez, G. Mauri, Cl. Ferretti, K. Khrithivasan, R. Freund, M. Margenstern, Y. Rogozhin, H. J. Hoogeboom, A. Obtulowicz, etc.

Books (co)authored by Gheorghe Păun:

1. Grammars for Economic Processes, The Technical Publ. House, Bucharest, 1980 (in Romanian).
2. Matrix Grammars, The Scientific and Enciclopaedic Publ. House, Bucharest, 1981 (in Romanian).
3. Contextual Grammars,The Publ. House of the Romanian Academy, Bucharest,1982 (in Romanian).
4. Recent Results and Problems in Formal Language Theory, The Scientific and Enciclopaedic Publ. House, Bucharest, 1984 (in Romanian).
5. The Paradoxes of Hierarchies, The Scientific and Enciclopaedic Publ. House, Bucharest, 1987 (in Romanian).
6. (in collaboration with J. Dassow, Germany) Regulated Rewriting in Formal Language Theory, Akademie-Verlag, Berlin, 1989, Springer-Verlag, Berlin, 1989 (nr. 18 in series Monograph on Theoretical Computer Science).
7. (in collaboration with E. Csuhaj-Varju, Hungary; J. Dassow, Germany; J. Kelemen, Czechoslovakia) Grammar Systems. A Grammatical Approach to Distribution and Cooperation, Gordon and Breach, series Topics in Computer Mathematics, London, 1994.
8. Marcus Contextual Grammars, Kluwer Academic Publ., Dordrecht, Boston, London, 1997.
9. (in collaboration with G. Rozenberg, A. Salomaa) DNA Computing. New Computing Paradigms, Springer-Verlag, Heidelberg, 1998 (translated in Japanese, in 1999, in Chinese and Russian in 2004).
10. (in collaboration with C. Calude) Computing with Cells and Atoms, Taylor and Francis, London, 2000.
11. Membrane Computing. An Introduction, Springer-Verlag, Berlin, 2002 (translated in Chinese, in 2012).

Collective volumes (co)edited by Gheorghe Păun:

1. Mathematical Aspects of Natural and Formal Languages, World Scientific Publishing, Singapore, 1994.
2. Mathematical Linguistics and Related Topics, Editura Academiei Romane, Bucharest, 1995.
3. Artificial Life: Grammatical Models, The Black Sea University Press, Bucharest, 1995.

4. (with A. Salomaa) New Trends in Formal Languages: Control, Cooperation, Combinatorics., Lecture Notes in Computer Science 1218, Springer-Verlag, Berlin, 1997.

5. Computing with Bio-Molecules. Theory and Experiments, Springer-Verlag, Singapore, 1998.

6. (with A. Salomaa) Grammatical Models of Multi-Agent Systems, Gordon and Breach, London, 1999.

7. (with J. Karhumaki, H.A. Maurer, G. Rozenberg) Jewels are Forever, Springer-Verlag, Berlin, 1999.

8. (with G. Ciobanu) Foundamentals of Computing Theory '99, Proceedings of the FCT Conf., Iasi, 1999, Lecture Notes in Computer Science, 1684, Springer-Verlag, Berlin, 1999.

9. (with C. Calude) Finite versus Infinite. Contributions to an Eternal Dilemma, Springer-Verlag, London, 2000.

10. (with C. Calude, M.J. Dinneen) Pre-proceedings of Workshop on Multiset Processing, Curtea de Argeş, Romania, August 2000, TR 140, CDMTCS, Univ. Auckland, New Zealand, 2000.

11. (with C. Martin-Vide) Recent Topics in Mathematical and Computational Linguistics, Ed. Academiei, Bucharest, 2000.

12. (with G. Rozenberg, A. Salomaa) Current Trends in Theoretical Computer Science. Entering the 21st Century, World Scientific, Singapore, 2001.

13. (with C. Martin-Vide) Pre-proceedings of Workshop on Membrane Computing, Curtea de Argeş, Romania, August 2001, TR 16/01, Univ. Rovira i Virgili, Tarragona, Spania, 2001.

14. (with C.S. Calude, G. Rozenberg, A. Salomaa), Multiset Processing. Mathematical, Computer Science, Molecular Computing Points of View, Springer-Verlag, Lecture Notes in Computer Science 2235, Berlin, 2001.

15. (with M. Ito, S. Yu) Words, Semigroups, Transductions (Festschrift in Honour of Gabriel Thierrin), World Scientific, Singapore, 2001.

16. (with D. Dascalu, E. Pincovschi, Vl. Topa, V. Voicu) Micro and Nanostructures, Ed. Academiei, Bucharest, 2001.

17. (with C. Zandron) Pre-proceedings of Workshop on Membrane Computing, Curtea de Argeş, Romania, MolCoNet Publication No 1, 2002.

18. (with G. Rozenberg, A. Salomaa, C. Zandron) Membrane Computing. International Workshop, WMC 2002, Curtea de Argeş, Romania, August 2002. Revised Papers, Lecture Notes in Computer Science 2597, Springer-Verlag, Berlin, 2003.

19. (with M. Cavaliere, C. Martin-Vide) Proceedings of the Brainstorming Week on Membrane Computing; Tarragona, February 2003, Technical Report 26/03, Rovira i Virgili University, Tarragona, 2003.

20. (with C. Martin-Vide, V. Mitrana) Formal Language Theory and Applications, Springer-Verlag, Berlin, 2004.

21. (with A. Alhazov, C. Martin-Vide), Pre-proceedings of Workshop on Membrane Computing, WMC 2003, Tarragona, Spain, July 2003, Technical Report 28/03, Rovira i Virgili University, Tarragona, 2003.

22. (with N. Jonoska, G. Rozenberg) Aspects of Molecular Computing. Essays Dedicated to Tom Head on the Occasion of His 70th Birthday, LNCS 2950, Springer-Verlag, Berlin, 2004.

23. (with G. Rozenberg, A. Salomaa) Current Trends in Theoretical Computer Science. The Challenge of the New Century, Vol. I Algorithms and Complexity, Vol. II Formal Models and Semantics , World Scientific, Singapore, 2004.

24. (with C. Martin-Vide, G. Mauri, G. Rozenberg, A. Salomaa) Membrane Computing, International Workshop, WMC 2003, Tarragona, July 2003, Selected Papers, LNCS 2933, Springer-Verlag, Berlin, 2004.

25. (with A. Riscos-Nunez, A. Romero-Jimenez, F. Sancho-Caparrini) Proceedings of the Second Brainstorming Week on Membrane Computing, Sevilla, February 2004, Technical Report 01/04 of Research Group on Natural Computing, Sevilla University, Spain, 2004.

26. (with J. Karhumaki, H. Maurer, G. Rozenberg) Theory is Forever. Essays Dedicated to Arto Salomaa, on the Occasion of His 70th Birthday, LNCS 3113, Springer-Verlag, Berlin, 2004.

27. (with G. Mauri, C. Zandron) Pre-proceedings of Fifth Workshop on Membrane Computing, WMC5, Milano, 2004.

28. (with G. Mauri, M.J. Perez-Jimenez, G. Rozenberg, A. Salomaa) Membrane Computing, International Workshop, WMC5, Milano, Italy, 2004, Selected Papers, LNCS 3365, Springer-Verlag, Berlin, 2005.

29. (with G. Ciobanu, M.J. Pérez-Jiménez) Applications of Membrane Computing, Springer-Verlag, Berlin, 2006.

30. (with M.A. Gutierrez-Naranjo, M.J. Perez-Jimenez) Cellular Computing. Complexity Aspects, Fenix Editora, Sevilla, 2005.

31. (with R. Freund, G. Lojka, M. Oswald) Proceedings of Sixth International Workshop on Membrane Computing, WMC6, Vienna, July 18-21, 2005.

32. (with G. Ciobanu) Pre-proceedings of Workshop on Theory and Applications of P Systems, TAPS'05, Timisoara, Septembrie 26-27, 2005.

33. (with C.S. Calude, M.J. Dinneen, M.J. Perez-Jimenez, G. Rozenberg) Unconventional Computation. 4th International Conference, UC2005, Sevilla, Spain, October 2005. Proceedings, LNCS 3699, Springer-Verlag, Berlin, 2005.

34. (with R. Freund, G. Rozenberg, A. Salomaa) Membrane Computing, International Workshop, WMC6, Vienna, Austria, 2005, Selected and Invited Papers, LNCS 3850, Springer-Verlag, Berlin, 2006.

35. (with M.A. Gutierrez-Naranjo et al) Proceedings of the Fourth Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, vol.1-2, 2006.

36. (with C.S. Calude, M.J. Dinneen, G. Rozenberg, S. Stepney) Unconventional Computation. 5th International Conference, UC2006, York, UK, September 2006. Proceedings, LNCS 4135, Springer-Verlag, Berlin, 2006.

37. (with H.J. Hoogeboom, G. Rozenberg) Workshop on Membrane Computing, WMC7, Leiden, July 17-21, 2006.

38. (with L. Pan) Pre-proceedings of the International Conference Bio-Inspired Computing-Theory and Applications, BIC-TA 2006 Volume of Membrane Computing Section, Wuhan, China, September 18- 22, 2006.

39. (with H.J. Hoogeboom, G. Rozenberg, A. Salomaa) Membrane Computing, International Workshop, WMC7, Leiden, The Netherlands, 2006, Selected and Invited Papers, LNCS 4361, Springer-Verlag, Berlin, 2007.

40. (with M.A. Gutierrez-Naranjo et al.) Proceedings of the Fifth Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, 2007.

41. (with G. Eleftherakis, P. Kefalas) Proceedings of Eight Workshop on Membrane Computing, Thessaloniki, June 2007.

42. (with G. Eleftherakis, P. Kefalas, G. Rozenberg, A. Salomaa) Membrane Computing, International Workshop, WMC8, Thessaloniki, Greece, 2007, Selected and Invited Papers, LNCS 4860, Springer-Verlag, Berlin, 2007.

43. (with M.A. Gutierrez-Naranjo et al.) Proceedings of the Sixth Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, 2008.

44. (with D. Corne, P. Frisco, G. Rozenberg, A. Salomaa) Membrane Computing, International Workshop, WMC9, Edinburgh, UK, Selected and Invited Papers, LNCS 5391, Springer-Verlag, Berlin, 2008.

45. (with G. Rozenberg, A. Salomaa) Handbook of Membrane Computing, Oxford University Press, 2010.

46. (with R. Gutierrez-Escudero et al.) Proceedings of the Seventh Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, volume 1-2, 2009.

47. (with M.J. Perez-Jimenez, A. Riscos-Nunez) Pre-proceedings of Tenth Workshop on Membrane Computing, WMC10, Curtea de Argeş, August 2009.

48. (with M.J. Perez-Jimenez, A. Riscos-Nunez, G. Rozenberg, A. Salomaa) Membrane Computing, Tenth International Workshop, WMC 2009, Curtea de Argeş Romania, August 2009, Selected and Invited Papers, LNCS 5957, Springer-Verlag, Berlin, 2009.

49. (with M.A. Gutierrez-Naranjo et al.) Proceedings of the Eighth Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, 2010.

50. (with M. Gheorghe, T. Hinze) Pre-proceedings of the 11th Conference on Membrane Computing, CMC11, Jena, Germany, 2010.

51. (with M. Gheorghe, T. Hinze, G. Rozenberg, A. Salomaa) Membrane Computing. 11th International Conference, CMC11, Jena, Germany, August 24-27, 2010. Revised, Selected, and Invited Papers, LNCS 6501, Springer-Verlag, Berlin, 2010.

52. (with M.A. Martinez-del-Amor, I. Perez-Hurtado, F.J. Romero-Campero, L. Valencia-Cabrera) Proceedings of the Ninth Brainstorming Week on Membrane Computing, Sevilla, 2011, Fenix Editora, Sevilla, 2011.

53. (with M. Gheorghe, S. Verlan) Pre-proceedings of Twelfth International Conference on Membrane Computing, CMC12, Fontainebleau, Paris, August 2011.

54. (with M. Gheorghe, G. Rozenberg, A. Salomaa, S. Verlan) Membrane Computing. 12th International Conference, CMC12, Fontainebleau, France, August 2011. Revised, Selected, Invited Papers, LNCS 7184, Springer-Verlag, Berlin, 2012.

55. (with M.A. Martinez-del-Amor et al.) Proceedings of the Tenth Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, vol. 1-2, 2012.

56. (with L. Valencia-Cabrera et al.) Proceedings of the Eleventh Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, 2013.

57. (with L. Pan, M.J.Pérez-Jiménez, T. Song) Proceedings of the 9th International Symposium BIC-TA 2014, Wuhan, China, Comm. in Computer and Information Sciences, vol. 472, Springer, 2014.

58. (with G. Rozenberg, A. Salomaa) Discrete Mathematics and Computer Science, Ed. Academiei, Bucureşti, 2014. v (cu L.F. Macias-Ramos et al.) Proceedings of the Twelfth Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, 2014.

59. (with L.F. Macias-Ramos et al.) Proceedings of the 13th Brainstorming Week on Membrane Computing, Fenix Editora, Sevilla, 2015.

4 Impact of G. Păun's Works

The publications of G. Păun have been cited in more that 15,000 international works, (co)authored by more than 1700 computer scientists.

More than 200 papers (co)authored by G. Păun are indexed/abstracted in ISI WoS, having over 3,000 citations and h-index = 28 [7]. The most cited paper is "Computing with membranes", [4], published in 2000, having more than 850 citations (the second position of all cited papers from Journal of Computer and System Sciences. [7]).

In 2009, G. Păun was included by ISI (Thompson-Reuters) in the Highly Cited Researchers category, which means he was at that time among the most cited 0.5% computer scientists in the world.

The most cited paper published in International Journal of Computers Communications & Control (IJCCC) is "Spiking Neural P Systems with Anti-Spikes", coauthored by G. Păun (with L. Pan), [2], [7]. Also the papers [6] and [5] are in top 10 of IJCCC.

Nowadays there exist over 2500 works and over 60 PhD thesis in the field of Membrane Computing and P Systems (and about 50 collective volumes), with over 500 (co)authors from Romania, Austria, The Netherlands, Germany, Finland, Japan, UK, Canada, Hungary, India, Italy, Spain, Czech Republic, USA, Poland, France, Republic of Moldova, China, Switzerland, Australia, New Zealand, Filipine, Malaesia, Slovakia etc.

A webpage dedicated to P Systems is hosted in Vienna [10]. Every year three meetings dedicated to P Systems are organized: Conference of Membrane Computing (former Workshop on Membrane Computing, initiated in 2000), Brainstorming Week on Membrane Computing (since

2003), and Asian Conference on Membrane Computing (since 2012). Several conferences in natural computing or unconventional computing explicitly have Membrane Computing in their scopes.

International recognition of G. Păun consist in:

- Invitations as visiting researcher received from many universities and research institutes from Hungary, Czech Republic, Slovakia, Germany, Finland, France, Japan, Holland, Austria, Spain, USA, Canada, Poland, Italy, New Zealand, Greece, China, Singapore, etc., and finished with many collaborations with local researchers.

- Over 100 invited lectures to universities from Germany (Magdeburg, Frankfurt, Hamburg, Tubingen), Hungary (Budapesta, Gyor), Czech Republic (Brno, Prague, Opava), Slovakia (Bratislava), Spain (Tarragona, Barcelona, Madrid, Sevilla), Finland (Turku, Laapeenranta), Holland (Leiden), Japan (Kyoto, Tokyo-Chiba, Tokyo-Waseda, Tokyo-Dendai, Hiroshima), France (Paris, Lille), UK (London), Canada (Ontario), Poland (Warsaw), US (Greenvile NC, Binghamton), Italy (Milano, Roma, Brescia, Pisa, L'Aquila, Siena, Palermo, Verona), Greece (Xanthi), China (Beijing, Wuhan), etc;

- Visiting professor at Technical University of Vienna (Austria), Turku Centre for Computer Science (Finland), Rovira i Virgili University of Tarragona (Spain) and Politechnica University of Madrid (Spain), Banach Center of Poland Academy of Sciences (Poland), University of Singapore, University of Malaysia, Hungarian Academy etc.

- Humboldt scholarship in Germany at University of Magdeburg (May 1, 1992 -July 31, 1993; July-August 1999); many scholarhips in France, Finland, Spain, The Netherlands;

- Research scholarship "Ramon y Cajal" in Spain (Tarragona and Sevilla) for five years (2001-2006) and leader of a project of excellency in Spania (Sevilla, 2009-2014).

- Member of the program committees of more than 120 international conferences, organizer of the symposiums Artificial Life: Grammatical Models (Mangalia - Romania, 1994), Molecular Computing (Mangalia - Romania, 1997), Multiset Processing (Curtea de Argeş - Romania, 2000), Membrane Computing (Curtea de Argeş 2001, 2002, 2009, Tarragona 2003, Milano 2004, Viena 2005, Leiden 2006, Salonic 2007, Edinburgh 2008, Jena 2010, Fontainebleau 2011, Budapesta 2012, Chisinau 2013, Praga 2014, Valencia 2015);

- Founder and main organizer of series "Brainstorming Week on Membrane Computing" (Tarragona 2003, Sevilla 2004-2015); member of steering committee of Conferences Developments in Language Theory, Universal Machines and Computations and DNA Based Computing, and of workshops Grammar Systems and Descriptional Complexity in Formal Systems.

G. Păun is/was member in Editorial Board of many journals, as: Journal of Universal Computer Science (Springer-Verlag), Journal of Computing and Informatics (Computers and Artificial Intelligence), Acta Cybernetica, Journal of Automata, Languages, and Combinatorics, Grammars, Fundamenta Informaticae, Romanian Journal of Information Science and Technology, Computer Science Journal of Moldova, International Journal of Foundations of Computer Science (World Scientific), International Journal of Computer Mathematics, Natural Computing: An International Journal, Soft Computing (Springer)- Area editor (DNA and membrane computing), BioSystems (Elsevier); Theoretical Computer Science. Natural Computing Series (Elsevier); International Journal of Unconventional Computing; New Generation Computing (Springer and Omsha-Japonia), Progress in Natural Science (Elsevier and Science in China Press), Economic

Computation and Economic Cybernetics Studies and Research, International Journal of Computers Communication & Control, etc.

He also received several scientific honors and awards, in Romania and abroad, among others, the Doctor Honoris Causa title from:

1. University of Opava – The Czech Republic (2008),
2. University of Piteşti - Romania (2010) and
3. Agora University of Oradea - Romania (2015).

5 Summary

A quick overview of Gheorghe Păun computer science activity and, especially, of the impact of his activity is provided, focusing on the data from ISI WoS and on the developments in Membrane Computing, a branch of Natural Computing initiated by him in 1998.

Acknowledgment

Celebrating Gheorghe Păun's 65th birthday, we wish him a good health, long life, and new scientific achievements!

Bibliography

[1] G. Ciobanu (2010); Writing as a Form of Freedom and Happiness Celebrating the 60th birthday of Gheorghe Păun, *International Journal of Computers Communications & Control*, 5(5): 613-615.

[2] L. Pan, G. Păun (2009); Spiking Neural P Systems with Anti-Spikes, *International Journal of Computers Communications & Control*, 4(3):273-282.

[3] Gheorghe Păun (1998); *Computing with Membranes*, Issue 208 of TUCS technical report, Turku Centre for Computer Science, 1998.

[4] G. Păun (2000); Computing with Membranes, *Journal of Computer and System Sciences*, 61(1):108-143.

[5] G. Păun; M.J. Pérez-Jiménez (2010); Solving Problems in a Distributed Way in Membrane Computing: dP Systems, *International Journal of Computers Communications & Control*, 5(2):238-250.

[6] G. Păun; M.J. Pérez-Jiménez, A. Riscos-Nunez (2008); Tissue P systems with cell division, *International Journal of Computers Communications & Control*, 3(3):295-303.

[7] http://isiwebofknowledge.com (retrieved on July 3, 2015).

[8] http://www.esi-topics.com/fbp/2003/february03-GheorghePaun.html (retrieved on May 15, 2015).

[9] http://www.imar.ro/gpaun/ (retrieved on April 10, 2015).

[10] http://ppage.psystems.eu/ (retrieved on April 10, 2015).

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):627-642, October, 2015.

Traffic Signal Control with Cell Transmission Model Using Reinforcement Learning for Total Delay Minimisation

P. Chanloha, J. Chinrungrueng, W. Usaha, C. Aswakul

Pitipong Chanloha and **Chaodit Aswakul***
Wireless Network and Future Internet Research Group,
Department of Electrical Engineering, Chulalongkorn University,
Thailand, 10330.
*Corresponding author: chaodit.a@chula.ac.th

Wipawee Usaha
School of Telecommunication Engineering, Institute of Engineering,
Suranaree University of Technology,
Muang District, Nakhon Ratchasima,Thailand, 30000.
wipawee@sut.ac.th

Jatuporn Chinrungrueng
National Electronics and Computer Technology Center,
National Science and Technology Development Agency,
Klong Luang, Pathumthani, Thailand, 12120.
jatuporn.chinrungrueng@nectec.or.th

Abstract: This paper proposes a new framework to control the traffic signal lights by applying the automated goal-directed learning and decision making scheme, namely the reinforcement learning (RL) method, to seek the best possible traffic signal actions upon changes of network state modelled by the signalised cell transmission model (CTM). This paper employs the Q-learning which is one of the RL tools in order to find the traffic signal solution because of its adaptability in finding the real time solution upon the change of states. The goal is for RL to minimise the total network delay. Surprisingly, by using the total network delay as a reward function, the results were not necessarily as good as initially expected. Rather, both simulation and mathematical derivation results confirm that using the newly proposed red light delay as the RL reward function gives better performance than using the total network delay as the reward function. The investigated scenarios include the situations where the summation of overall traffic demands exceeds the maximum flow capacity. Reported results show that our proposed framework using RL and CTM in the macroscopic level can computationally efficiently find the proper control solution close to the brute-forcely searched best periodic signal solution (BPSS). For the practical case study conducted by AIMSUN microscopic traffic simulator, the proposed CTM-based RL reveals that the reduction of the average delay can be significantly decreased by 40% with bus lane and 38% without bus lane in comparison with the case of currently used traffic signal strategy. Therefore, the CTM-based RL algorithm could be a useful tool to adjust the proper traffic signal light in practice.
Keywords: Traffic Signal Control (TSC), Cell Transmission Model (CTM), Reinforcement Learning (RL).

1 Introduction

The opportunities in expanding physical transportation capacity within a well-established city are becoming practically limited. With continuing social and economic growth in metropolitan areas, many transportation facilities are being used to their full capabilities. The effects of demand management by using innovative local policies to match the unique nature of local

Copyright © 2006-2015 by CCC Publications

demand still remain to be explored. Without adding new facilities, attempts to operate and control the traffic by exploring existing capacity are challenging. Fortunately, the traffic problems can be handled by using advanced traffic information and control systems, which are among the most classical problems in traffic engineering. And computer technologies have been applied to find the optimal traffic signal timing for facilitating the traffic movements. More importantly, the main persistent challenge of the traffic problems is the ability to adapt traffic signals according to unexpectedly temporal traffic demand or road condition changes.

In this regard, intelligent *learning* methods to control the traffic signal and deal with the unexpected dynamics of road congestion status have been proposed in the literature. Of particular interest in this paper is an unsupervised-learning approach to find good traffic signal controls from experiences gained gradually by interacting directly with the road congestion environment. The herein adopted approach, reinforcement learning (RL) [1] and its potential to deal with the traffic engineering problems has been first proposed by [2]. More recently, in [3], Q-learning has been addressed as an RL technique to improve the control of integrated traffic corridor. For an isolated-intersection control [4], Q-learning has also been applied to control the traffic signal lights. All these existing literature have formulated RL with the road congestion environment that has been modelled in the detailed dynamics of individual mobility. Such behavior of microscopic traffic flow models can be too limited in their practical usages when the computational complexity becomes a major concern.

Alternatively, regarding the choices of environment models for RL, during the actual implementation phase of algorithm in the road network, RL can also be designed to measure directly its reward value from the observable abstraction of the considered road segments. However, a direct exposure of any learning algorithms to the actual system during a trial-and-error phase can lead to severe risks on road traffic problems and unsatisfactory public acceptance of the new control system. In this paper, we have proposed to integrate the RL concept with a computationally convenient macroscopic traffic flow model. In particular, the well-established *cell transmission model* or CTM [5] has been employed in this paper. The original version of CTM has been first proposed to model vehicle movements in an unsignalised network. Following developments of CTM to support a signalised road network have been proposed by [6], and further investigated by [7] for a TRANSYT system, and the signal optimisation with genetic algorithm has been proposed by [8] and by [9] based on a mixed-integer linear programming for two intersections. CTM has been refined recently to model the behaviors of multiple traffic classes [10] and with a more flexible topology mapping [11]. Recently, Q-learning has also been proposed with CTM to model the traffic flow dynamics [12] but the considered cases therein are successfully applicable to only a traffic route guidance problem, not to the signal optimisation problem as emphasised in this paper.

To study the traffic condition when the summation of overall traffic demand from all directions exceeds the maximum flow capacity, in the past we must rely on the whole network model. However, in this paper, by merely using a single intersection network scenario and with herein introduced boundary conditions to capture necessary vehicles backlog dynamics around the vicinity of the considered intersection, our proposed model can help reduce computational burdens a great deal. In addition, the developed CTM model can still maintain interesting insightful interpretations of control sequences. Particularly, with newly formulated RL environment using CTM parameters, three reward functions of RL to minimise the total network delay have been evaluated by considering the accumulative vehicle delays in only the directions facing the red light signal, receiving the green light signal, or in both directions. Our numerical experiments have shown that the choice of the red light delay has been found to outperform all the other choices in various traffic conditions. Moreover, the solution obtained from the macroscopic-viewpoint of RL has been compared with that from the best periodic signal solution (BPSS). Our comparison

using the microscopic simulator AIMSUN has helped confirm the applicability of the proposed CTM-based RL signal optimisation in this paper.

2 Problem Formulation

2.1 State Space

Suppose the vehicles in the systems belong to a single class e.g. personal cars. As shown

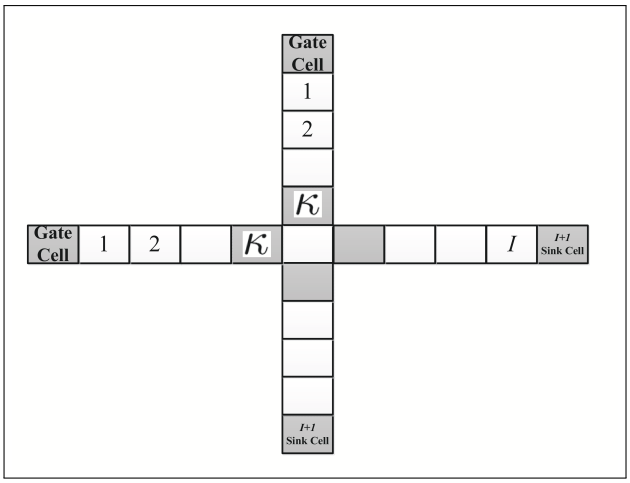


Figure 1: CTM boundaries and signalised cells

in Fig. 1, each road is partitioned into small cells $i = 1, ..., I$. The incoming demand patterns to an intersection is classified into P directions. Let $\mathcal{S}$ be the state space of the system. For each vehicle cell i in direction p at time slot t, define $s_i^p(t)$ as the number of vehicles. Let $\mathbf{s}(t) = [s_i^p(t),\ \forall(i,p)] \in \mathcal{S}$ be the state vector which represents the total number of vehicles in the system at time slot t. Note that in a real traffic scenario, the number of vehicles can be estimated from sensors on the road. To avoid the computational burden caused by the state space explosion, the quantisation technique is employed. The level of quantisations here can be represented by the number of deployed sensors in the road network. For simplification, let us define the quantised level of the total number of vehicles approaching the intersection from direction p at time slot t as $\tilde{s}^p(t) = \left\lceil \frac{\sum_{i=1}^{\kappa} s_i^p(t)}{\mathcal{C}} f \right\rceil + I(\sum_{i=1}^{\kappa} s_i^p(t) = 0)$, where $I(.)$ is the indicator function; $\mathcal{C}$ is the maximum number of vehicles totally allowable in each cell $i = 1, ..., \kappa$; and f is the total number of quantisation levels. The RL state can then be redefined as $\tilde{\mathbf{s}}(t) = [\tilde{s}^p(t),\ \forall p]$.

2.2 Cell Transmission Model

To incorporate the evolution of traffic dynamics in the system, a basic macroscopic model CTM is employed. The CTM parameters can be defined as follows [5].

Sending capability

Sending capability represents the ability to send the vehicles from cells to other cells, i.e., moving vehicles from beginning to ending cells. The sending capability can be defined as

$$\Lambda_i^p(t) = \min\left\{s_i^p(t), q_i^p(t)\right\}. \tag{1}$$

For cell i in direction p at time slot t, $\Lambda_i^p(t)$ is the sending capability; $s_i^p(t)$ is the number of vehicles; and $q_i^p(t)$ is the maximum number of vehicles that can flow through cell i.

Receiving capability

Receiving capability can be calculated by considering the remaining spaces in each cell and the maximum number of vehicles that can be present in the cell. Thus, for cell i in direction p at time slot t, its receiving capability can be defined as

$$\Psi_i^p(t) = \min\{q_i^p(t), \delta_i^p\,[c_i^p(t) - s_i^p(t)]\}, \tag{2}$$

where δ_i^p is the wave speed coefficient and $c_i^p(t)$ is the maximum number of vehicles that can be present. Note that the parameter $q_i^p(t)$ is influenced by the signal phase being chosen in cell i, direction p and time slot t in the action selection.

Cell cascading

This is the representation of the connection between two adjacent cells from the beginning cell $i-1$ and the ending cell i. The number of vehicles that flow in this cascading scenario can be calculated from the sending and receiving capability by

$$y_i^p(t) = \min\{\Lambda_{i-1}^p(t), \Psi_i^p(t)\}, \tag{3}$$

where $y_i^p(t)$ is the number of vehicles that flow into cell i in direction p at time slot t.

Flow conservation

Flow conservation is used to update the number of vehicles for the next time slot:

$$s_i^p(t+1) = s_i^p(t) + y_i^p(t) - y_{i+1}^p(t). \tag{4}$$

2.3 Action Space

To influence the system dynamics, for each time slot, the control agent (traffic controller) must select whether it would keep the current signal indication or change it. Such decision is called action. At state vector $\tilde{\mathbf{s}}$, an action must be selected from a state dependent set $\mathcal{A}(\tilde{\mathbf{s}})$. Specifically, $\mathcal{A}(\tilde{\mathbf{s}})$ is the set of all possible actions which a traffic controller can take at state $\tilde{\mathbf{s}}$. Define action a_t as the phase of signal light to be chosen (e.g, phase 1 for the green light from West to East and phase 2 for that from North to South) at time slot t. The indicator function $G^p(t)$ becomes one (zero) when vehicles in direction p get green (red) light in the chosen action at time slot t. Note that the action space $\mathcal{A}(\tilde{\mathbf{s}})$ must be defined such that all conflicting flows are not allowed to have green light at the same time.

The system dynamics are changed according to the traffic signal lights corresponding to the action taken $a_t \in \mathcal{A}(\tilde{\mathbf{s}})$. Assume that in one time slot, vehicles can move on average to the adjacent cells only. Let q_{max} be the maximum number of vehicles that can flow through each cell per time slot. For non-signalised cell i, the maximum number of vehicles that can flow through cell i in direction p at time slot t is given by $q_i^p(t) = q_{max}, \forall(p,t)$. For signalised cell i, the maximum number of vehicles that can flow through cell i is $q_i^p(t) = q_{max}$ when $G^p(t) = 1$ and $t - \tau_i(t) > L$. Otherwise, $q_i^p(t) = 0$. Note that L is the total starting/stopping loss time upon each signal change and $\tau_i(t)$ is the latest time instant where the traffic signal indication of cell i at time slot t has been changed.

Gate cell "0"

The boundary condition is here formulated by following [5]. At the boundary, input vehicle flows can be modelled by a cell pair ("00" and cell "0"). A source cell "00" with an infinite number of vehicles $s_{00}^p(t) = \infty$ ready to enter an initially empty gate cell "0" of infinite size, $c_0^p(t) = \infty$. The flow capacity $q_0^p(t)$ of the gate cell "0" is set to the desired link input flow. Thus, the boundary conditions can be can be expressed and written by $\Lambda_0^p(t) = \min s_0^p(t), q_0^p(t)$, $y_0^p(t) = q_0^p(t)$, $y_1^p(t) = \min\{\Lambda_0^p(t), \Psi_1^p(t)\}$ and $s_0^p(t+1) = s_0^p(t) + y_0^p(t) - y_1^p(t)$. Suppose that the output cell referred as the sink cell, for all exiting traffic has infinite size $c_{I+1}^p(t) = \infty$ and $q_{I+1}^p(t) = \infty$, The sink cell $I+1$ thus has the receiving capability of $\Psi_{I+1}^p(t) = \infty$.

2.4 Vehicle Delay

In RL, to quantify the consequence of the action taken at time slot t, an immediate reward in terms of vehicle delay is returned to the agent (traffic controller). Vehicle delay is defined as the number of vehicles that cannot move away from the present cell within each time slot. In this research, two types of vehicle delay are proposed, i.e., internal delay and external delay. At time slot t for each direction p, let $d_0^p(t)$ be the external vehicle delay and $d_i^p(t)$ be the internal vehicle delay in cell i. These delays can be expressed as

$$d_0^p(t) = s_0^p(t) - y_1^p(t), \tag{5}$$
$$d_i^p(t) = s_i^p(t) - y_{i+1}^p(t), \quad i = 1, 2, \ldots, I. \tag{6}$$

The external delay can be interpreted as the delay experienced by the vehicles that wait to enter the considered road network from its upstream neighbourhoods. The external delay value forms the boundary condition to capture necessary vehicle backlog dynamics around the vicinity of the considered intersection. The internal vehicle delay is the delay incurred within each cell along the considered road network. Combining both types of delay therefore reflects how well the action just taken by the agent (traffic controller) at state vector $\tilde{\mathbf{s}}$ is, by merely taking into account a single intersection. The next section provides the long term performance criteria in terms of these delay functions which will be optimised for the best possible traffic signal control by means of RL.

2.5 Performance Criteria

To evaluate the optimal policy (set of actions) that minimises the total network delay, the performance criteria $\Upsilon(t)$ at time slot t is defined as

$$\Upsilon(t) = \Upsilon_{red}(t) + \Upsilon_{green}(t), \tag{7}$$
$$\Upsilon_{red}(t) = \sum_{p=1}^{P}\sum_{i=0}^{I}(1 - G^p(t))d_i^p(t), \tag{8}$$
$$\Upsilon_{green}(t) = \sum_{p=1}^{P}\sum_{i=0}^{I} G^p(t)d_i^p(t), \tag{9}$$

where $\Upsilon_{red}(t)$ is the "red light delay" and $\Upsilon_{green}(t)$ is the "green light delay". The red (green) light delay is the total vehicle delay from all the cells in the directions that see the red (green) light.

3 Signal Optimisation By Q-learning Algorithm

Without loss of generality, let us index the signalised cells by κ as an example of CTM-based intersection model shown in Figure 1. Assume no turning movement is allowed at this intersection. The signalised cells κ are used to control the traffic flows from West to East and North to South. To tackle the road traffic problem where the system always changes, a well-known method that can learn directly from experiences is employed, namely, the Q-learning method [1]. Q-learning uses the action-value function $Q(\tilde{\mathbf{s}}, a)$ to evaluate the average future reward return expressed as a function of the current state $\tilde{\mathbf{s}}$ and action a. This section explains a step-by-step implementation of Q-learning algorithm proposed in the CTM framework.

To apply RL in a signalised CTM framework, a definite simulation length is used for periodically observing traffic behaviors within a study time-interval. When the current time slot of CTM reaches the simulation length, the system enters the next *episode*. In practice, episodes can represent the repeatable and non-repeatable traffic phenomena. On one hand, in a repeatable case, we can use RL to tackle a recurrent congestion, e.g. during rush hours, in which traffic behaviours statistically repeat themselves from one day to another. In this case, at the beginning of each episode, our road system modelled by CTM can be reset to the same initial-value cell density settings. On the other hand, in a non-repeatable case, RL can be used to deal with a non-recurrent congestion scenario resulted from unexpected incidences like accidents or road surface maintenance. In this case, our interest is on how RL would allow the signal controller to quickly learn and adapt its strategic decisions upon those unexpected changes. Consequently, the CTM state in the first time slot of next episode is defined in this case as the CTM state in the last time slot of previous episode.

Whether RL is applied in the repeatable or non-repeatable cases, within each episode, the RL-based traffic controller is designed to make a sequence of signal-light decisions. Let the decision epoch t_ω refer to the time instant when decision ω is made, where $\omega = 1, 2, \ldots$ and $t_\omega = t_1, t_2, \ldots$, respectively.

For each episode, the optimisation procedure of Q-learning can be summarised as follows. 1) *System Initialisation*
The number of vehicles in state vector $\mathbf{s}(0)$ can be intialised by (4) at the beginning of an episode to the latest observed state of the system in the previous episode in the non-repeatable case or to a nominal operating point of the system at the considered time period in the repeatable case. In practice, the number of vehicles $\tilde{s}^p(0)$ for all p can be measured from road traffic by counting from the sensors embedded on the road. The action value function $Q(\tilde{\mathbf{s}}, a)$ can be initialised to the latest updated value in the previous episode for the non-repeatable case or to zero for the repeatable case. It should be noted that, different intialisations of $Q(\tilde{\mathbf{s}}, a)$ yield different results, mainly, in terms of the time convergence (the time that the algorithm needs to learn to reach the solution). Let $\omega = 1$.
2) *Action Selection*
At decision ω, with the current state observable at $\tilde{\mathbf{s}}$, the agent (traffic controller) chooses an action $a \in \mathcal{A}(\tilde{\mathbf{s}})$ to control the traffic signal by changing $G^p(t)$ in Section 2.3. The action can be chosen by the ϵ-greedy algorithm [1], where the greedy action is here defined as

$$a = \arg\min_{a\prime} Q(\tilde{\mathbf{s}}, a\prime).$$

According to this algorithm [1], Q-learning chooses the greedy action with probability $1-\epsilon$. And, with probability ϵ, the other actions are randomly selected according to a uniform distribution. In practice, an ϵ is a small positive value representing the explorability of learning algorithm.
3) *Update of System Dynamics*
Calculate the CTM state from time slot $t = t_\omega$ to time slot $t = t_{\omega+1} - 1$. Here, the next state

vector ($\tilde{\mathbf{s}}\prime$) is calculated from the CTM state at time slot $t = t_{\omega+1} - 1$. In this paper, three Q-functions have been compared, namely, the total network delay by considering the accumulative vehicle delays in only the directions facing red light signal, receiving the green light signal, or both. The observed reward $R(\omega)$ can then be correspondingly calculated from

$$R(\omega) = \begin{cases} \sum_{t=t_\omega}^{t_{\omega+1}-1} \Upsilon(t) & \text{in case of total network delay} \\ \sum_{t=t_\omega}^{t_{\omega+1}-1} \Upsilon_{red}(t) & \text{in case of red light delay} \\ \sum_{t=t_\omega}^{t_{\omega+1}-1} \Upsilon_{green}(t) & \text{in case of green light delay.} \end{cases} \tag{10}$$

4) *Update of Action Value Function*
In this paper, the algorithm can learn from its past experiences accumulated in Q-function and the reward in (10) newly gained from the most recent action ω. By following [1], Q-function can be updated as follows

$$Q(\tilde{\mathbf{s}}, a) \leftarrow Q(\tilde{\mathbf{s}}, a) + \alpha[R(\omega) + \gamma \min_{a\prime} Q(\tilde{\mathbf{s}}\prime, a\prime) - Q(\tilde{\mathbf{s}}, a)].$$

Here, $Q(\tilde{\mathbf{s}}\prime, a\prime)$ represents the action value function for the next observable state vector $\tilde{\mathbf{s}}\prime$ and next possible action $a\prime \in \mathcal{A}(\tilde{\mathbf{s}}\prime)$. Practically, $\alpha \in (0, 1]$ is the learning rate and $\gamma \in [0, 1)$ is the discount rate applied to the future expected rewards.
5) *Update of State Variable and Timing Parameter*
Update state $\tilde{\mathbf{s}} \leftarrow \tilde{\mathbf{s}}\prime$. And update $\omega \leftarrow \omega + 1$.
6) *Stopping Condition*
Repeat steps 2)–5) until the end of episode.

4 Results and Discussions

This section is aimed at reporting the findings from our series of experiments. Firstly, the convergence time and corresponding computational complexity of the proposed Q-learning algorithm has been presented. Secondly, three reward functions in (10) have been compared in terms of the achievable minimum total network delay values. Thirdly, with the best choice in the reward value accounting for the vehicle delay in red-light traffic direction, Q-learning performance has been investigated in stationary/non-stationary stochastic loading scenarios. Lastly, the applicability of macroscopic CTM-based solution of the proposed Q-learning algorithm has been tested in microscopic mobility environments using AIMSUN. All the experimental results share the following common parameter settings.

1. System Parameters: As illustrated in Figure 1, suppose that the length of each road approaching the considered intersection is 800 metres and each road is discretised into 10 equal-length cells, i.e. $I = 10$. Each time slot has been set to 5 seconds. Each cell has the capacity $c_i^p(t)$ of 60 passenger car units (pcu) and the maximum flow rate $q_i^p(t)$ of 6.9 pcu/slot. The wave speed coefficient δ_i^p is 0.8. Note that the values of CTM parameters are based on the actual traffic data collection being calibrated for Payathai road in Bangok, Thailand [10].

2. Control Parameters: The length of each episode is 20 minutes or 240 time slots. An action has been chosen every 3 time slots. Note that the longer the action selection is, the more

outdated the decision becomes. The number of quantisation levels f has been set to 3. Practically, three levels are corresponding to the three sensors that are often deployed on the real road configuration. The first sensor at the entry of the road is used for preventing the spill-back of vehicles to upstream neighbourhoods. The second sensor is deployed in the middle of the road for the queue length estimation. The third sensor placed at the stop-line of the road is used for the wasted green prevention in an actuated signal control.

4.1 RL Validation

This paper proposes the newly developed version of the signalised CTM with RL. The validation of the RL in various traffic conditions are reported. The optimal signal timing under static traffic condition with fixed cycle length, namely, best periodic signal solution (BPSS) and the Q-learning solution by using the proposed framework have been compared. Define λ_1 and λ_2 as the average rate of arrival traffic from West to East and North to South, respectively. Consider deterministic demand patterns with $\{\lambda_1, \lambda_2\} = \{8,8\}, \{11,5\}, \{13,3\}, \{15,1\}$ pcu/slot. Note that the other traffic conditions can be achieved by other sets of demand patterns as well, but we have analysed the example of four settings given above. From trial-and-error, the RL parameters are set to $\epsilon = 0.1, \alpha = 0.01, \gamma = 0.005$ within 100 episodes. Theoretically, the learning rate (α) determines how fast the newly acquired information will override the old information. The possible value of α is in the range of $0 < \alpha \leq 1$. The discount factor (γ) determines the importance of future rewards where $0 \leq \gamma < 1$. If $\gamma = 0$, then the agent will be "opportunistic" by only considering current rewards. The parameter ϵ is a small probability, where a larger ϵ is used for a more exploration-oriented design and a smaller ϵ is used for a more exploitation-oriented design [1]. In practice, the parametric tuning for the algorithm is one of the major challenges because in different scenarios, the parameters need to be readjusted. However, the advantage of the effects of Q-learning parameters is the usable range of these parameters are wide. With the flexibility of the Q-learning parameters, the obtained solution of Q-learning can be found without readjusting as discussed in the following section of the performance in stationary/non-stationary stochastic loadings. By using our proposed red light delay (8) as the

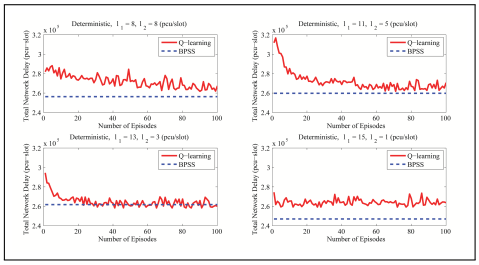


Figure 2: Total network delay from Q-learning vs BPSS

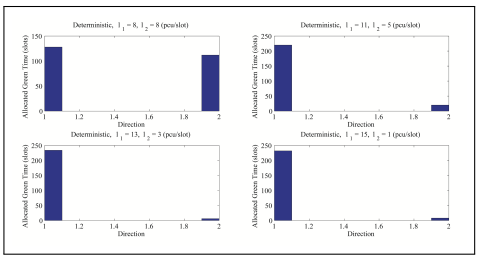


Figure 3: Allocated green time to each direction in last episode of Q-learning

reward function, Figure 2 and Figure 3 illustrate the total network delay and the allocated green time to each direction, respectively. Note that the red light delay used herein has been chosen from the following subsection focusing on the effect of reward functions. Figure 2 shows that the total network delay from Q-learning can be found close to the solution from BPSS in most scenarios. Particularly, when $\{\lambda_1, \lambda_2\} = \{15,1\}$ pcu/slot, Q-learning solution yields unsatisfactory result because of small traffic λ_2. Technically speaking, Q-learning requires the knowledge from its past experiences. But with a small traffic demand, the system cannot offer sufficient experiences to the Q-learning in order to achieve the solution properly. Figure 3 shows the number of time slots allocated to each direction. The result shows that the allocated green time in each

direction is proportional to the incoming traffic demand of that direction.

The computational complexity has been measured in terms of the required amount of memory and the computational time to achieve the final solution. Let the number of elements in the quantised state space be denoted by $|\tilde{\mathcal{S}}|$ and that in the action space be denoted by $|\mathcal{A}|$. Note that the action space $|\mathcal{A}| = P$ where P is the total number of all road network directions. Let k be the total number of the green time pairs in the overall searching space of periodic signal solutions. To search for the BPSS within these k possibilities per each state, the required amount of memory is $O(a^P)$ (approximately 748 kbytes) where a is a constant. However, the amount of memory required for Q-learning is $O(|\tilde{\mathcal{S}}|P)$ (approximately 114 kbytes). The BPSS grows exponentially depending on the number of the green time pairs to be searched whereas the growth of Q-learning depends on the quantised state space and the number of actions. The memory requirement can be saved with respect to the increasing of k. The computational time for the BPSS is $O(a^P)$ (approximately 1222 seconds) whereas the computational time for the Q-learning is $O(|\tilde{\mathcal{S}}|P)$ (approximately 15 seconds). The result shows that the computation of Q-learning for obtaining a control signal is significantly faster than BPSS.

4.2 Effect Of Reward Functions

The procedure to find the traffic signal solution has been illustrated in the RL validation. In this subsection, three different reward functions have been investigated in both symmetric and asymmetric loading patterns. To make the experiments more realistic, the traffic demand is no longer deterministic. In this subsection, the traffic demand is a Poisson process with a constant arrival rate for each direction. For symmetric loadings, both directions have equal approaching demand from $\{1,1\},\{3,3\},...,\{15,15\}$ pcu/slot, respectively. For asymmetric loadings, λ_1 has been set to 13 pcu/slot and λ_2 is varied from $1,2,...,15$ pcu/slot. The results have been obtained with the manually fine-tuned RL parameters $\epsilon = 0.1, \alpha = 0.01, \gamma = 0.005$. As illustrated

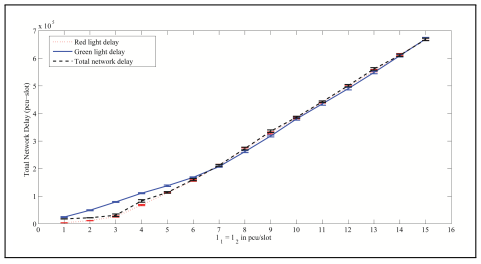


Figure 4: Total network delay from three reward functions on symmetric loadings

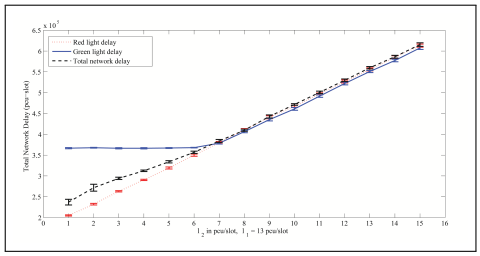


Figure 5: Total network delay from three reward functions on asymmetric loadings

in Figure 4 and Figure 5, with 95% confidence interval of both symmetric and asymmetric loadings, the proposed red light delay as the reward function decreases the total network delay in comparison with the conventional case of total network delay as the reward function and greatly decreases the total network delay in comparison with the case of green light delay as the reward function. The previous statement is valid in the loading region where the summation of overall traffic demand from all directions does not exceed its maximum flow capacity ($\lambda_1 + \lambda_2 \leq 6.9$ pcu/slot). On the contrary, when the summation of overall traffic demand from all directions exceeds the maximum flow capacity ($\lambda_1 + \lambda_2 > 6.9$ pcu/slot), the case of green light delay as the reward function yields slightly low total network delay in comparison with the case using the other two reward functions. Consider the system in the case where the summation of overall traffic demand from all directions does not exceed its maximum flow capacity. In this case, any control strategy can be used because usually there is no congestion of vehicles. In such

scenario, the control strategy is not complicated. However, the system in the case where the summation of overall traffic demand from all directions exceeds the maximum flow capacity, the control strategy has concerned because traffic congestion becomes a severe problem. Therefore, the following discussion will focus on the case of the summation of overall traffic demand from all directions exceeds the maximum flow capacity only.

Mathematical analysis when the summation of overall traffic demand from all directions exceeds the maximum flow capacity

To discuss all the results under the condition when the summation of overall traffic demand from all directions exceeds the maximum flow capacity, define the major flow (minor flow) as the incoming traffic demands that exceed (does not exceed) the capacity. Two types of the road traffic phenomena have been investigated. The experiments are concerned with a major flow conflicted with a minor flow (Ma-Mi condition) and two major flows conflicted with each other (Ma-Ma condition). For the Ma-Mi condition, consider an example demand setting $\{\lambda_1, \lambda_2\} = \{13, 3\}$ pcu/slot. Our experimental results in Figure 6, Figure 7 and Figure 8 show the total network delay in each time slot, the delay of all cells in each direction and the action chosen in each time slot, respectively. All the results in Figure 6, Figure 7 and Figure 8 have been observed at the final episode at the convergence.

With a simplified derivation, our result can be explained by using mathematical analysis as follows. Consider the derivation of accumulative delay of all cells in each direction as used in Figure 6 to Figure 8. From (5) – (6), the accumulative delay of all cells in direction p up to time slot T can be obtained from $\sum_{t=0}^{T}\sum_{i=0}^{I} d_i^p(t) = \sum_{t=0}^{T}\sum_{i=0}^{I}\big(s_i^p(t) - y_{i+1}^p(t)\big)$.

At the asymptote (all the cells in overloaded direction being fully occupied), define $\bar{\Upsilon}_{red}$, ($\bar{\Upsilon}_{green}$) as the asymptotic increasing rate of expected value of the accumulative red (green) light delay. Likewise, define $\bar{\Upsilon}$ as the asymptotic increasing rate of expected value of the accumulative total network delay. The term $y_{i+1}^p(t)$ becomes zero when calculating $\Upsilon_{red}(t)$ and becomes non-zero (6.9 pcu/slot) when calculating $\Upsilon_{green}(t)$. The calculation is therefore given by

$$\bar{\Upsilon}_{red} = \begin{cases} 3 - 0 = 3, & G^1(t) = 1 \\ 13 - 0 = 13, & G^2(t) = 1, \end{cases} \tag{11}$$

$$\bar{\Upsilon}_{green} = \begin{cases} 13 - 6.9 = 6.1, & G^1(t) = 1 \\ \max(3 - 6.9, 0) = 0, & G^2(t) = 1, \end{cases} \tag{12}$$

$$\bar{\Upsilon} = \begin{cases} 3 + 6.1 = 9.1, & G^1(t) = 1 \\ 13 + 0 = 13, & G^2(t) = 1 \end{cases} \tag{13}$$

From (11), if the reward function is $\Upsilon_{red}(t)$, then the minimum total network delay can be achieved by allocating the green light signal to the major flow (λ_1). Likewise, in (12), if the reward function is $\Upsilon_{green}(t)$, then the minimum total network delay can be achieved by allocating the green light signal to the minor flow (λ_2). Using $\Upsilon_{green}(t)$ as the reward function leads to the wasted green scenario (green light allocation to a particular direction without remaining vehicles) as illustrated by the term $max(3 - 6.9, 0)$. However, if $\Upsilon(t)$ is chosen as the reward function, then the minimum total network delay can be achieved by allocating the green light signal to the major flow (λ_1). The total network delay is a bit higher than the case of $\Upsilon_{red}(t)$. To explain why the total network delay from $\Upsilon(t)$ is higher than $\Upsilon_{red}(t)$. There are two concerned effects in using $\Upsilon_{red}(t)$ or $\Upsilon(t)$ as the reward function. One is the indistinguishable effect from $\Upsilon(t)$ where the agent only knows the overall network delay (7). Regardless of whether proper or improper

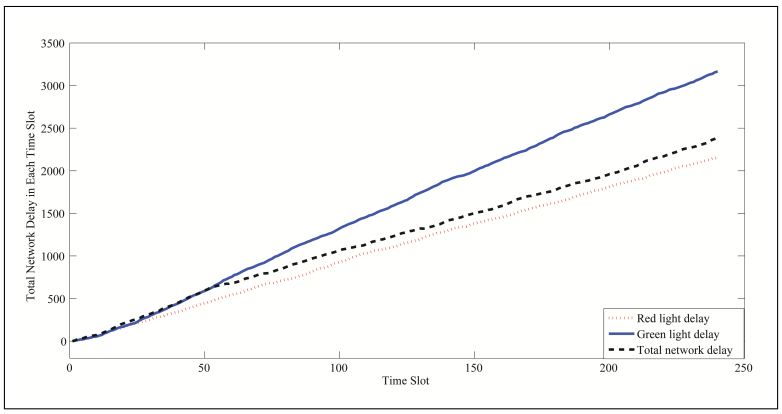


Figure 6: Ma-Mi: Total delay in each time slot

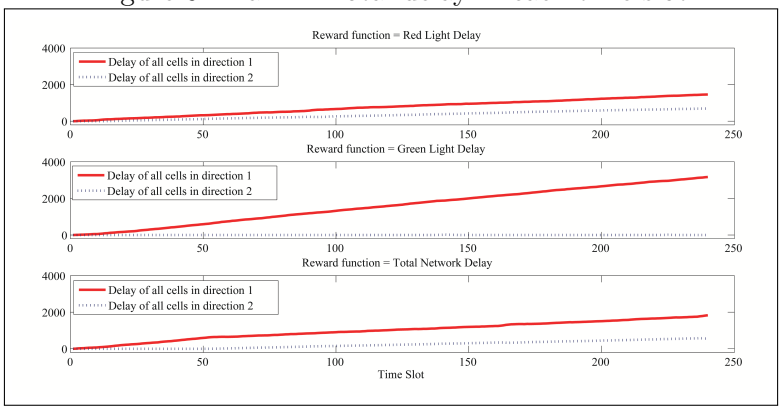


Figure 7: Ma-Mi: Three types of reward functions and its delay in each component

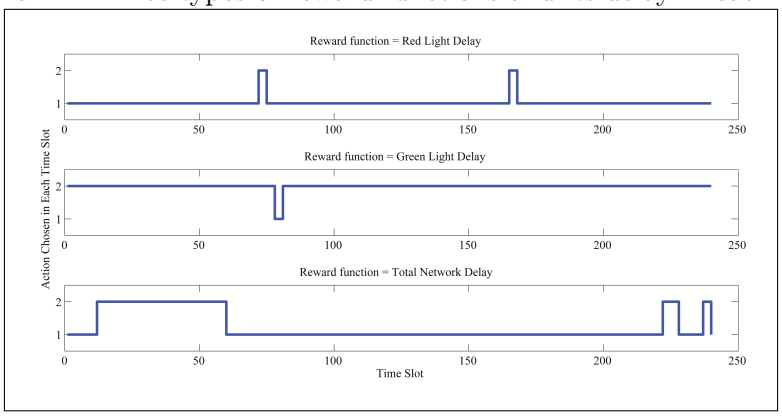


Figure 8: Ma-Mi: Action chosen in each time slot

action has been chosen, the value of reward in terms of total network delay is indifferent due to the summation of all vehicle delays in the system. The indistinguishable effect results in an inaccuracy (an improper action selection) and an inefficiency (an increasing of undesirable total network delay) of the action selection from Q-learning. Another is the timing effect of switched actions. In this case, the more often the action switches, the worse the total network delay is. From the discussion in the Ma-Mi condition, the recommended reward function would be the red light delay ($\Upsilon_{red}(t)$), which gives the lowest total network delay in comparison with the other two reward functions.

Likewise the Ma-Mi condition, the Ma-Ma condition can be calculated using the same ideas

above. For Ma-Ma conditions, the proper management of traffic signal control becomes a major concern. The next recommended traffic signal would preferably remain the same as the current traffic signal to avoid the occurrence of the system loss time. The any reward function can be used because the traffic is jammed. The reduction of total network delay becomes insignificant.

The goal is for RL to minimise the total network delay. Surprisingly, by using the total network delay as a reward function, the results were not necessarily as good as initially expected. Rather, both simulation and mathematical derivation results confirm that using the newly proposed red light delay as the RL reward function gives better performance than using the total network delay as the reward function. Note that a good reward function must be able to allow the algorithm to steer its instantaneous searching directions towards the final goal of minimising the total network delay. But that reward function itself needs not be the objective function i.e. the total network delay. Instead, from our numerical experiments, one should rather opt for using the red-light delay as the reward function so that the effect on future expected total network delay can be reflected within only a few time slots after an action decision has been made. On the contrary, if the total network delay is used as the reward function, then the algorithm eventually cannot find the proper solution.

4.3 Q-Learning Performance In Stationary/Non-Stationary Stochastic Loadings

In the RL validation section, four different traffic demand patterns have been investigated. In fact, such simplification can be relaxed to more realistic case by considering on the random source probabilities. Let the traffic demand be a Poisson process with a constant arrival rate for each direction. From the previous subsection, the red light delay has been chosen as a reward function. The performance of Q-learning in adapting its solution to reach the convergence will be examined. The experiments have been set into two scenarios. Firstly, the stationary test, the change of traffic demand from a deterministic to a Poisson has been illustrated in Figure 9. Secondly, the non-stationary test, in reality, road network capacity changes upon time (early morning, rush hour, etc.) as illustrated in Figure 10. Starting from uncongested traffic condition, the 1^{st} episode until the 100^{th} episode, the traffic demand pattern is $\{\lambda_1, \lambda_2\} = \{6, 6\}$ pcu/slot. And then, the road network becomes congested (jammed) condition, the $101^{st} - 140^{th}$ episodes, traffic demand pattern is therefore changed to $\{\lambda_1, \lambda_2\} = \{13, 3\}$ pcu/slot. The congested condition returns to uncongested condition, the $141^{st} - 180^{th}$ episodes, the traffic demand pattern is $\{\lambda_1, \lambda_2\} = \{6, 6\}$ pcu/slot. Finally, the congested condition happened again, the episodes 181^{st} the traffic demand pattern is $\{\lambda_1, \lambda_2\} = \{11, 5\}$ pcu/slot. The results show the adaptability of

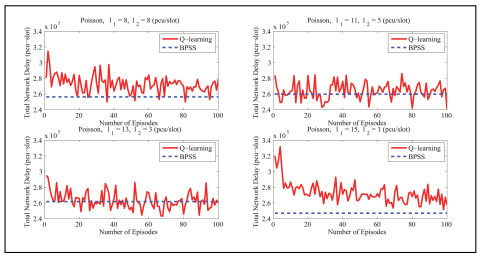


Figure 9: Total network delay from Q-learning with Poisson arrival

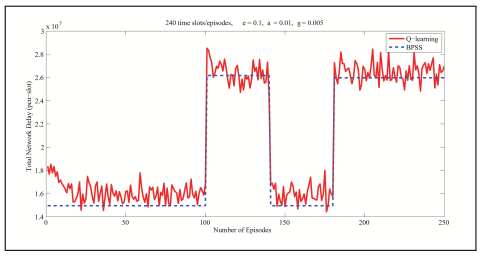


Figure 10: Total network delay obtained from Q-learning with the change of load patterns

Q-learning in reaching the solution close to the obtained solution from the BPSS method in both experiments. The abrupt change of the traffic demand patterns from uncongested to congested conditions have been imposed. However, the Q-learning still performs well in tracking closer to

the BPSS solution. Therefore, with significantly less demanding computational time than BPSS, the Q-learning algorithm can be used in real-time learning-based scenarios.

4.4 Comparison in Practical Case Study

In this subsection, we investigate an isolated intersection located in the middle of Bangkok, Thailand. As illustrated in Figure 11, the investigation area covers the Ratchapruek and the Ratchadaphisek roads where two segments are the arterial. While operating in the rush-hour periods, two roads are fully occupied. In reality, the intersection is controlled by the 3-phase signal timing plan. However, from the measured data, one of these three signal phases can be considered as the minor road segments because there are relatively few vehicles in comparison with the other two directions. So, the signal phases have been simplified to only two signal phases allocated to the Ratchapruek and the Ratchadaphisek road, respectively. The simulation settings have been set to the real data measured from the embedded sensors on each road segment at the intersection. All the considered road segments are 3-lane; the length of each road is 800 metres and each road is equally divided into 10 equal-length cells. The system parameters have been based on the previous subsection. The maximum green time is set to 120 time slots. The control plan of the CTM-based RL obtained from the MATLAB will be applied to the AIMSUN for the study in microscopic levels.

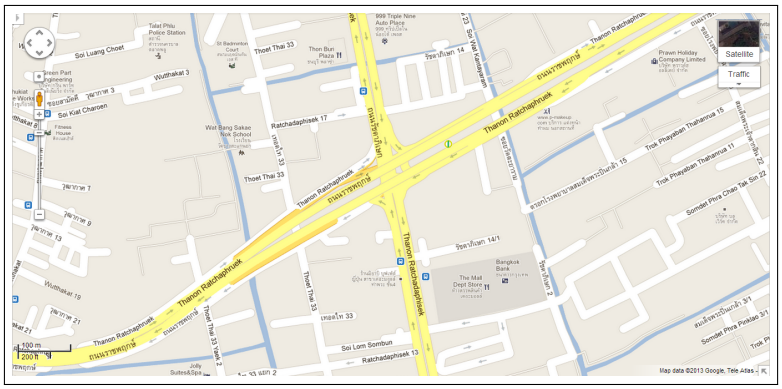


Figure 11: Ratchadaphisek-Ratchapruek intersection

Figure 12 and Figure 13 illustrate the time history of the number of vehicles approaching the Ratchadaphisek and Ratchapruek intersection. For the microscopic simulation settings, a bus

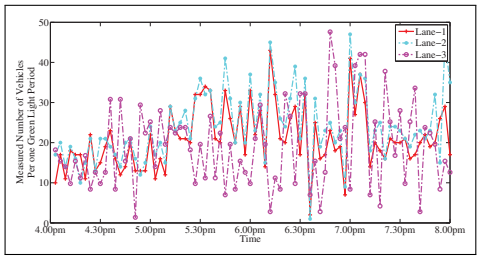


Figure 12: Number of vehicles from the Ratchadaphisek road approaching the intersection

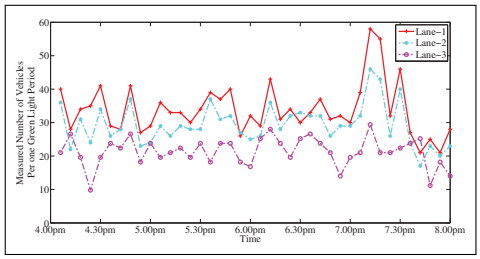


Figure 13: Number of vehicles from the Ratchapruek road approaching the intersection

stop is placed on the Ratchadaphisek road at 200 metres before its stopping line. As illustrated

in Figure 14, each bus waits at the bus stop for picking up the passengers. The 1^{st} lane of the Ratchadaphisek road has been occupied by buses and taxis and the 2^{nd} lane has been affected from those public vehicles. Therefore, the part of the road segment around the bus stop becomes a temporary blockage. In this scenario, the bus stop has been transparently embedded into the CTM model by permanently reducing the average vehicle speed on the bus lane. With the measured data, the simulation testing in the AIMSUN has been set to 4 hours in the evening rush-hour period from 4.00 pm to 8.00 pm. From the recorded data, the average speed of the vehicle is 50 km/h. However, the average speed in the bus lane has been reduced from 50 km/h to 8 km/h and the average speed of the middle lane has been reduced from 50 km/h to 30 km/h, respectively. The results have been obtained on three different signal control strategies which are the actual control obtained from the sensors, the fully actuated control using the embedded sensors deployed along the road segments and the control plan from CTM-based RL obtained from MATLAB. As illustrated in Table. 1, the proposed CTM-based RL reveals that

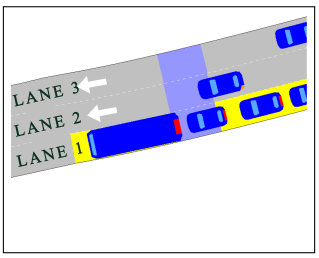


Figure 14: Road network with a bus lane

Table 1: The comparison among three types of traffic signal control

	Actual control	Actuated control	CTM-based RL
average vehicle delay (seconds)	137	97.9	82.7
average vehicle delay reduction (percent)	-	28.54	39.64
time spent for vehicle dissipations (hours)	3.4	3.1	2.6

the reduction of the average delay can be significantly decreased by 39.64%. From the recorded data, the green light status has been changed too often. By that, this control plan is not good as expected because of the system loss time from the frequent signal switching. For the fully actuated control, the maximum-gap distance technique between vehicles has been employed. Note that the actuated control employs vehicle detectors installed around an intersection to change the traffic signals of that intersection. Once vehicle detectors response for actuation, the actuated phase normally starts with a minimal preset green time, and green time phase is automatically extended [13]. The average speed of the vehicles has been reduced from the bus stop. Therefore, the traffic signal has been changed upon their own gap distance and individual speed. As illustrated in the previous subsections, the green light will be opened as long as possible up to the maximum green time. The CTM-based RL tries to avoid the system loss time; therefore, the average delay can be reduced. Moreover, the total time spent for vehicle dissipations has also decreased from 3.4 hours to 2.6 hours (approximately reduced by 22.53%). As illustrated in Table. 2, the scenario has been slightly changed by removing the bus lane from the actual bus stop whereas the traffic arrivals from the other directions are totally unchanged. In such scenario, the average speed of the vehicle is set 50 km/h because the temporal blockage from buses and taxis has been removed. For the case when the removing the bus stop, the proposed CTM-based RL reveals that the reduction of the average delay can be significantly

Table 2: The comparison among three types of traffic signal control without bus lane

	Actual control	Actuated control	CTM-based RL
average vehicle delay (seconds)	89.5	60.3	55.7
average vehicle delay reduction (percent)	-	32.63	37.77
time spent for vehicle dissipations (hours)	2.7	2.5	2.1

decreased by 37.77% and significantly reduced the time spent for vehicle dissipations by 22.22%.

5 Conclusion

A new framework to control the traffic signal lights by applying one of the reinforcement learning tools, namely, the Q-learning has been proposed to seek the best possible solution to control the traffic signals where the network state has been modelled by the signalised cell transmission model. The road traffic condition is mainly focused on the situation when the summation of overall traffic demand from all directions exceeds the maximum flow capacity.

The proposed framework is used to find the best traffic signal strategy. Surprisingly, using the newly proposed red light delay as the RL reward function gives better performance than using the total network delay as the reward function. The results have been reported from the series of experiments which are the RL validation, the effect of reward functions, the RL performance in stationary/non-stationary stochastic loadings and the applicability of the CTM-based solution of the RL algorithm in the microscopic mobility environments using AIMSUN.

The simulation results show that our proposed framework can computationally efficiently find the proper solution for road traffic systems by comparing with the best periodic signal solution (BPSS). The effect of reward functions has also been investigated and the adaptability of the RL algorithm in adjusting its solution with Poisson arrival upon the change of time has also been observed. The results from the macroscopic level show that RL yields the results similar to the BPSS method. For the practical case study conducted by AIMSUN, the proposed CTM-based RL reveals that the reduction of the average delay can be significantly decreased by 39.64% from the actual traffic signal strategy. For the case when the removing the bus stop, the CTM-based RL also has also been reduced the average delay by 37.77%. Therefore, the practical case study from the urbanised isolated intersection can provide substantially impact to the transportation problems.

With the newly proposed reward function applied to an isolated intersection, this paper has reported the results and its applicabilities. The extension of our proposed framework for a road network scale is currently ongoing and the results will be reported in the forthcoming papers.

Acknowledgments

The authors would like to acknowledge support received from Thailand Graduate Institute of Science and Technology (TGIST), associated with National Science and Technology Development Agency (NSTDA), the Special Task Force for Activating Research (STAR) Funding in Wireless Network and Future Internet Research Group, Chulalongkorn University and Associate Professor Dr. Sorawit Narupiti at transport engineering division of Department of Civil Engineering, Chulalongkorn University for technical support in transportation infrastructures and intelligence controls.

Bibliography

[1] R. Sutton, A.G. Barto (1998); *Reinforcement Learning: An Introduction.* The MIT Press, Cambridge, Massachusetts.

[2] B. Abdulhai, L. Kattan (2003); Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering*, 30(6), 981–991.

[3] C. Jacob, B. Abdulhai (2005); Integrated traffic corridor control using machine learning. *International Conference on Systems, Man & Cybernetics*, 3460–3465.

[4] D.D. Oliveira et al (2006); Reinforcement learning-based control of traffic lights in non-stationary environments: a case study in a microscopic simulator. *Forth European Workshop on Multi Agent Systems.*

[5] C.F. Daganzo (1995); The cell transmission model part II: Network traffic. *Transportation Research Part B: Methodological*, 29b(2), 79–93.

[6] H.K. Lo et al (2001); Dynamic network traffic control. *Transportation Research Part A: Policy and Practice*, 35(8), 721–744.

[7] M. Maher, O. Feldman (2002); The application of the cell transmission model to the optimisation of signals on signalised roundabouts. *European Transport Conference*, 1–13.

[8] H.K. Lo, A.H.F. Chow (2004); Control strategies for oversaturated traffic. *Journal of Transportation Engineering*, 466–478.

[9] W.H. Lin, C. Wang (2004); An enhanced 0-1 mixed-integer LP formulation for traffic signal control. *IEEE Transactions on Intelligence Transportation Systems*, 5(4): 238–245.

[10] K. Tueprasert, C. Aswakul (2010); Multiclass cell transmission model for heterogeneous mobility in general topology of road network. *Journal of Intelligent Transportation Systems*, 14(2): 68–82.

[11] G. Flotterod, K. Nagel (2005) Some practical extensions to the cell transmission model. *Proceedings of the 8th Internationall IEEE Conference on Intelligent Transportation Systems.*

[12] A. Sadek, N. Basha (2006); Self-learning intelligent agents for dynamic traffic routing on transportation networks. *International Conference on Complex Systems*, 503–518.

[13] N.H. Gartner et al (1995); Development of advanced traffic signal control strategies for Intelligent Transportation Systems : multilevel design, *Transportation Research Record*, 98–105.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):643-653, October, 2015.

ANN Training Method with a Small Number of Examples Used for Robots Control

E. Ciupan , F. Lungu, C. Ciupan

Emilia Ciupan, Florin Lungu, Cornel Ciupan*
Technical University of Cluj-Napoca
Romania, 400641 Cluj-Napoca, Bd. Muncii, 103-105
emilia.ciupan@mis.utcluj.ro, florin.lungu@mis.utcluj.ro
*Corresponding author: cornel.ciupan@muri.utcluj.ro

Abstract: This paper presents a method for obtaining a neural model used in industrial robots control. The method refers to the forming of a small number of examples used in the training of a neural network that lead to the creation of a suitable model. This paper constitutes a development of the work [2] in order to increase the opportunities for its application in various fields. The description of the method is generally done, without relying on a specific application in the domain of industrial robots. The testing and the validation of the shown method were completed using the example of a system in which the relationship between inputs and outputs is described by means of mathematical functions. The set of learning examples, generated through the proposed method, served to the ANN training by a cross-validation technique, in case of these functions. The evaluation of the proposed method has been done by analysing the results obtained by applying it compared to those obtained with a known method, namely the uniform generation of training examples. The use of the method in the field of industrial robots' control was illustrated by a concrete application in the case of a robot with 6 degrees of freedom.
Keywords: ANN, training, method, small number, robot, control.

1 Introduction

The control proper is achieved by the robot's control equipment by generating a control value for each joint, so that the joint achieve coordinate q_i resulted from the inverse kinematics, and the effector move through points of coordinates that belong to the trajectory. Therefore determining the coordinates of the joints is of capital importance [7].

Neural networks can perform complex learning and adaptation tasks by imitating the function of biological neural systems, and thus can be used as models for nonlinear, multi-variable systems, trained by using input-output data observed on the system [3].

The application of neural networks to robots control is well known [1, 4, 6] and an alternative to the adaptive control is represented by the neural controllers [8].

The models based on neural networks show an advantage in terms of model simplification and, first and foremost, of the operations performed, as they consist solely of multiplications and additions. A model based on neural networks that would answer parallel robot control involves appropriate modelling and very good network training. The large number of the training data necessary for a high-quality neural model, which can reach thousands of examples [5], can often be problematic. That is why the problem that this paper solves consists in the completion of a method for obtaining a high-quality neural network with a small number of training examples, one that grants both the desired precision in the entire robot's workspace and a reduction of the training time. Let us consider a robot with six degrees of freedom having three translation axes that give the positioning movement X, Y, Z, and other three rotation axes that give the orientation movement ψ, θ, φ.

Copyright © 2006-2015 by CCC Publications

Based on the robot's kinematic scheme [4], there has been determined the mathematical model for the direct kinematics (1)-(2), as well as the model for inverse kinematic analysis (3)-(4):

$$X = q_1 \cdot i_{Tx} + X_0; Y = q_2 \cdot i_{Ty} + Y_0; Z = q_3 \cdot i_{Tz} + Z_0; \tag{1}$$

$$\psi = q_4 \cdot i_{T\psi} + \psi_0; \theta = q_5 \cdot i_{T\theta} + \theta_0; \varphi = q_6 \cdot i_{T\varphi} + \varphi_0; \tag{2}$$

$$q_1 = \frac{X - X_0}{i_{Tx}}; q_2 = \frac{Y - Y_0}{i_{Ty}}; q_3 = \frac{Z - Z_0}{i_{Tz}}; \tag{3}$$

$$q_4 = \frac{\psi - \psi_0}{i_{T\psi}}; q_5 = \frac{\theta - \theta_0}{i_{T\theta}}; q_6 = \frac{\varphi - \varphi_0}{i_{T\varphi}}; \tag{4}$$

where i_{Tx}, i_{Ty}, i_{Tz}, $i_{T\psi}$, $i_{T\theta}$, $i_{T\varphi}$ represent the transfer functions of the transformation mechanisms that generate the given movements and X_0, Y_0, Z_0, ψ_0, θ_0, φ_0 represent the values of the system origin for which q_i=0, i=1,...,6.

2 Description of the method used for the generation of training examples and its validation

The majority of robots have six degrees of freedom and, consequently, six joints. The current paper aims at finding a method of reducing the number of training examples of a neural network for the control of a robot with six degrees of freedom. In order to describe the method, a random system with 6 input signals, and 6 output ones, is being considered. In the process of building the neural network that is to shape the system, there is used a set of training examples in which the input is represented by vectors of values (q_1, q_2, q_3, q_4, q_5, q_6), where $q_i \in [q_{min}, q_{max}]$, $\forall i = \overline{1,6}$. The output values of the system are described by the values of the vectorial function $F = (f_1, f_2, f_3, f_4, f_5, f_6)$, where F:$[q_{min}, q_{max}]^6 \rightarrow R^6$.

According to the method proposed, the set of training examples results from the imposition of a successive move of the signals q_i, i=1,...,6, by the successive move of each signal by a step p, followed by mixed moves of several axes, for each set j of input data $q_{i,j}$, i=1,...,6 and j=1,...,m, where m is the total number of sets of this type, resulting in a set of output data $F(q_{1j}, q_{2j}, q_{3j}, q_{4j}, q_{5j}, q_{6j})$ that is used for the network training. The training and testing data is determined according to the mathematical model for the function F by modifying the input signals according to Table 1.

In the mathematical model for function F, sets of the input signals (q_1, q_2, q_3, q_4, q_5, q_6) are gradually being introduced, and the output values are calculated. The data obtained after the completion of Table 1 represents the set of training examples. The novelty of this method consists in the way used to determine the neural network training data in order to achieve a high-quality neural network by means of a small number of training examples. A high-quality neural network is defined here as a neural model that offers acceptable errors in the entire domain of function F. In order to train the network effectively with a relatively reduced number of training examples, we suggest the use of an algorithm of generating the set of training examples which implies the completion of 30 phases expressed concisely in Table 1. A phase, denoted here by h, corresponds to a set of pairs of six (q_i, i=1,...,6), one for each factor z of multiplication of the step p, as it is shown explicitly in phase 1 in the Table 1. The modification of the inputs is done based on the relation below:

Phase	Input values (q_i, i=1,…,6)						Output values $P_{h,z}(f_1,f_2,f_3,f_4,f_5,f_6)$, h=1,…,30, z=0,…,j					
h	q_1	q_2	q_3	q_4	q_5	q_6	f_1	f_2	f_3	f_4	f_5	f_6
1.	$\mathbf{q_{1min}+z\cdot p}$	$q_{2\,min}$	$q_{3\,min}$	$q_{4\,min}$	$q_{5\,min}$	$q_{6\,min}$	$f1_{1,0}$	$f2_{1,0}$	$f3_{1,0}$	$f4_{1,0}$	$f5_{1,0}$	$f6_{1,0}$
							…	…	…	…	…	…
							$f1_{1,z}$	$f2_{1,z}$	$f3_{1,z}$	$f4_{1,z}$	$f5_{1,z}$	$f6_{1,z}$
							…	…	…	…	…	…
							$f1_{1,j}$	$f2_{1,j}$	$f3_{1,j}$	$f4_{1,j}$	$f5_{1,j}$	$f6_{1,j}$
2.	q_{1min}	$\mathbf{q_{2\,min}+z\cdot p}$	$q_{3\,min}$	$q_{4\,min}$	$q_{5\,min}$	$q_{6\,min}$	$f1_{2,z}$	$f2_{2,z}$	$f3_{2,z}$	$f4_{2,z}$	$f5_{2,z}$	$f6_{2,z}$
3.	q_{1min}	$q_{2\,min}$	$\mathbf{q_{3\,min}+z\cdot p}$	$q_{4\,min}$	$q_{5\,min}$	$q_{6\,min}$	$f1_{3,z}$	$f2_{3,z}$	$f3_{3,z}$	$f4_{3,z}$	$f5_{3,z}$	$f6_{3,z}$
4.	q_{1min}	$q_{2\,min}$	$q_{3\,min}$	$\mathbf{q_{4\,min}+z\cdot p}$	$q_{5\,min}$	$q_{6\,min}$	$f1_{4,z}$	$f2_{4,z}$	$f3_{4,z}$	$f4_{4,z}$	$f5_{4,z}$	$f6_{4,z}$
5.	q_{1min}	$q_{2\,min}$	$q_{3\,min}$	$q_{4\,min}$	$\mathbf{q_{5\,min}+z\cdot p}$	$q_{6\,min}$	$f1_{5,z}$	$f2_{5,z}$	$f3_{5,z}$	$f4_{5,z}$	$f5_{5,z}$	$f6_{5,z}$
6.	q_{1min}	$q_{2\,min}$	$q_{3\,min}$	$q_{4\,min}$	$q_{5\,min}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{6,z}$	$f2_{6,z}$	$f3_{6,z}$	$f4_{6,z}$	$f5_{6,z}$	$f6_{6,z}$
7.	$\mathbf{q_{1min}+z\cdot p}$	$q_{2\,med}$	$q_{3\,med}$	$q_{4\,med}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{7,z}$	$f2_{7,z}$	$f3_{7,z}$	$f4_{7,z}$	$f5_{7,z}$	$f6_{7,z}$
8.	q_{1med}	$q_{2\,min}+z\cdot p$	$q_{3\,med}$	$q_{4\,med}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{8,z}$	$f2_{8,z}$	$f3_{8,z}$	$f4_{8,z}$	$f5_{8,z}$	$f6_{8,z}$
9.	q_{1med}	$q_{2\,med}$	$q_{3\,min}+z\cdot p$	$q_{4\,med}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{9,z}$	$f2_{9,z}$	$f3_{9,z}$	$f4_{9,z}$	$f5_{9,z}$	$f6_{9,z}$
10.	q_{1med}	$q_{2\,med}$	$q_{3\,med}$	$q_{4\,min}+z\cdot p$	$q_{5\,med}$	$q_{6\,med}$	$f1_{10,z}$	$f2_{10,z}$	$f3_{10,z}$	$f4_{10,z}$	$f5_{10,z}$	$f6_{10,z}$
11.	q_{1med}	$q_{2\,med}$	$q_{3\,med}$	$q_{4\,med}$	$q_{5\,min}+z\cdot p$	$q_{6\,med}$	$f1_{11,z}$	$f2_{11,z}$	$f3_{11,z}$	$f4_{11,z}$	$f5_{11,z}$	$f6_{11,z}$
12.	q_{1med}	$q_{2\,med}$	$q_{3\,med}$	$q_{4\,med}$	$q_{5\,med}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{12,z}$	$f2_{12,z}$	$f3_{12,z}$	$f4_{12,z}$	$f5_{12,z}$	$f6_{12,z}$
13.	$\mathbf{q_{1min}+z\cdot p}$	$q_{2\,max}$	$q_{3\,max}$	$q_{4\,max}$	$q_{5\,max}$	$q_{6\,max}$	$f1_{13,z}$	$f2_{13,z}$	$f3_{13,z}$	$f4_{13,z}$	$f5_{13,z}$	$f6_{13,z}$
14.	q_{1max}	$\mathbf{q_{2\,min}+z\cdot p}$	$q_{3\,max}$	$q_{4\,max}$	$q_{5\,max}$	$q_{6\,max}$	$f1_{14,z}$	$f2_{14,z}$	$f3_{14,z}$	$f4_{14,z}$	$f5_{14,z}$	$f6_{14,z}$
15.	q_{1max}	$q_{2\,max}$	$\mathbf{q_{3\,min}+z\cdot p}$	$q_{4\,max}$	$q_{5\,max}$	$q_{6\,max}$	$f1_{15,z}$	$f2_{15,z}$	$f3_{15,z}$	$f4_{15,z}$	$f5_{15,z}$	$f6_{15,z}$
16.	q_{1max}	$q_{2\,max}$	$q_{3\,max}$	$\mathbf{q_{4\,min}+z\cdot p}$	$q_{5\,max}$	$q_{6\,max}$	$f1_{16,z}$	$f2_{16,z}$	$f3_{16,z}$	$f4_{16,z}$	$f5_{16,z}$	$f6_{16,z}$
17.	q_{1max}	$q_{2\,max}$	$q_{3\,max}$	$q_{4\,max}$	$\mathbf{q_{5\,min}+z\cdot p}$	$q_{6\,max}$	$f1_{17,z}$	$f2_{17,z}$	$f3_{17,z}$	$f4_{17,z}$	$f5_{17,z}$	$f6_{17,z}$
18.	q_{1max}	$q_{2\,max}$	$q_{3\,max}$	$q_{4\,max}$	$q_{5\,max}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{18,z}$	$f2_{18,z}$	$f3_{18,z}$	$f4_{18,z}$	$f5_{18,z}$	$f6_{18,z}$
19.	$\mathbf{q_{1min}+z\cdot p}$	$\mathbf{q_{2\,min}+z\cdot p}$	$q_{3\,med}$	$q_{4\,med}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{19,z}$	$f2_{19,z}$	$f3_{19,z}$	$f4_{19,z}$	$f5_{19,z}$	$f6_{19,z}$
20.	q_{1med}	$q_{2\,med}$	$\mathbf{q_{3\,min}+z\cdot p}$	$\mathbf{q_{4\,min}+z\cdot p}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{20,z}$	$f2_{20,z}$	$f3_{20,z}$	$f4_{20,z}$	$f5_{20,z}$	$f6_{20,z}$
21.	q_{1med}	$q_{2\,med}$	$q_{3\,med}$	$q_{4\,med}$	$\mathbf{q_{5\,min}+z\cdot p}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{21,z}$	$f2_{21,z}$	$f3_{21,z}$	$f4_{21,z}$	$f5_{21,z}$	$f6_{21,z}$
22.	$\mathbf{q_{1min}+z\cdot p}$	$\mathbf{q_{2\,min}+z\cdot p}$	$\mathbf{q_{3\,min}+z\cdot p}$	$q_{4\,med}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{22,z}$	$f2_{22,z}$	$f3_{22,z}$	$f4_{22,z}$	$f5_{22,z}$	$f6_{22,z}$
23.	q_{1med}	$q_{2\,med}$	$q_{3\,med}$	$\mathbf{q_{4\,min}+z\cdot p}$	$\mathbf{q_{5\,min}+z\cdot p}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{23,z}$	$f2_{23,z}$	$f3_{23,z}$	$f4_{23,z}$	$f5_{23,z}$	$f6_{23,z}$
24.	$\mathbf{q_{1min}+z\cdot p}$	$\mathbf{q_{2\,min}+z\cdot p}$	$\mathbf{q_{3\,min}+z\cdot p}$	$\mathbf{q_{4\,min}+z\cdot p}$	$q_{5\,med}$	$q_{6\,med}$	$f1_{24,z}$	$f2_{24,z}$	$f3_{24,z}$	$f4_{24,z}$	$f5_{24,z}$	$f6_{24,z}$
25.	q_{1med}	$q_{2\,med}$	$q_{3\,min}+z\cdot p$	$\mathbf{q_{4\,min}+z\cdot p}$	$\mathbf{q_{5\,min}+z\cdot p}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{25,z}$	$f2_{25,z}$	$f3_{25,z}$	$f4_{25,z}$	$f5_{25,z}$	$f6_{25,z}$
26.	$\mathbf{q_{1min}+z\cdot p}$	$\mathbf{q_{2\,min}+z\cdot p}$	$\mathbf{q_{3\,min}+z\cdot p}$	$\mathbf{q_{4\,min}+z\cdot p}$	$\mathbf{q_{5\,min}+z\cdot p}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{26,z}$	$f2_{26,z}$	$f3_{26,z}$	$f4_{26,z}$	$f5_{26,z}$	$f6_{26,z}$
27.	$\mathbf{q_{1min}+z\cdot p}$	$q_{2\,min}$	$\mathbf{q_{3\,min}+z\cdot p}$	$q_{4\,min}$	$\mathbf{q_{5\,min}+z\cdot p}$	$q_{6\,min}$	$f1_{27,z}$	$f2_{27,z}$	$f3_{27,z}$	$f4_{27,z}$	$f5_{27,z}$	$f6_{27,z}$
28.	$\mathbf{q_{1min}+z\cdot p}$	$q_{2\,max}$	$\mathbf{q_{3\,min}+z\cdot p}$	$q_{4\,max}$	$q_{5\,min}+z\cdot p$	$q_{6\,max}$	$f1_{28,z}$	$f2_{28,z}$	$f3_{28,z}$	$f4_{28,z}$	$f5_{28,z}$	$f6_{28,z}$
29.	q_{1min}	$\mathbf{q_{2\,min}+z\cdot p}$	$q_{3\,min}$	$\mathbf{q_{4\,min}+z\cdot p}$	$q_{5\,min}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{29,z}$	$f2_{29,z}$	$f3_{29,z}$	$f4_{29,z}$	$f5_{29,z}$	$f6_{29,z}$
30.	q_{1max}	$\mathbf{q_{2\,min}+z\cdot p}$	$q_{3\,max}$	$\mathbf{q_{4\,min}+z\cdot p}$	$q_{5\,max}$	$\mathbf{q_{6\,min}+z\cdot p}$	$f1_{30,z}$	$f2_{30,z}$	$f3_{30,z}$	$f4_{30,z}$	$f5_{30,z}$	$f6_{30,z}$

Table 1: Generation of training examples

$$q_i = q_{imin} + z \cdot p; \tag{5}$$

where z=0,...,j for all the 30 phases, except phases 2-6, where z=1,...,j and p represents the increase step given by the relation:

$$p = \frac{q_{imax} - q_{imin}}{j}; \tag{6}$$

and j is the number of steps. For the validation of the method, there has been considered the example in which the input signals $q_i \in [0, 90]$, $\forall i = \overline{1,6}$, the step p=5, and the components of the vectorial function F are defined by the following equations:

$$f_1(q_1, q_2, q_3, q_4, q_5, q_6) = 10 \cdot sinq_1 \tag{7}$$

$$f_2(q_1, q_2, q_3, q_4, q_5, q_6) = 10 \cdot (1 - cosq_2) \tag{8}$$

$$f_3(q_1, q_2, q_3, q_4, q_5, q_6) = \frac{10}{q_{max}^2} \cdot q_3^2 \tag{9}$$

$$f_4(q_1, q_2, q_3, q_4, q_5, q_6) = \frac{q_4}{9} \tag{10}$$

$$f_5(q_1, q_2, q_3, q_4, q_5, q_6) = \frac{q_5}{9} \tag{11}$$

$$f_6(q_1, q_2, q_3, q_4, q_5, q_6) = \frac{q_6}{9} \tag{12}$$

According to the rule described concisely in Table 1, there has been generated a set of training data of the neural network formed by pairs of examples $((q_1, q_2, q_3, q_4, q_5, q_6), F(q_1, q_2, q_3, q_4, q_5, q_6))$. After the elimination of the repeated values, there has been obtained a set of 382 examples, marked as trainProp. In order to compare the method proposed with the known method, there has been generated the set trainKnown, with a number of elements (375) close to that of the set trainProp. The training examples of this set have been obtained by means of the same approach as in the case of the robots inverse kinematic analysis. The starting point was made of uniformely distributed values (step 3.3) in the image of function F, respectively $[0, 10]^6$. The values of the corresponding input signals q_i, i=1,...,6 has been determined by means of the inverses of the functions $f_1, ..., f_6$. Table 2 illustrates the two sets, trainProp and trainKnown.

In the case of both sets, there has been applied the technique 4-fold cross-validation for the random division of each initial set of training examples into four subsets $sSetProp_l$, and $sSetKnown_l$, $l = \overline{1,4}$, respectively. The subsets of each set have been used, in a combination of three, to the training of the network, while the fourth subset has been used for the validation of the model throughout the training process. We mark the neural models obtained as a result of the instruction as $netProp_k$ and $netKnown_k$,$k = \overline{1,4}$, respectively. The errors obtained in the training process are close in the case of the two methods used, and they are of order 10^{-7}. In order to test the netProp_k and netKnown_k models, k=$\overline{1,4}$, there has been constituted the set of input-output data rendered in Table 3.

The data in Table 4 show better results in the cases in which, for the training stage, there has been used the set obtained by means of the method proposed, as compared to the set achieved by means of the method known. More detailed data concerning the individual errors of the output signals are comprised in the tables 5 and 6, both in the case of the method proposed, and in that of the method known. The variations of the absolute errors of the output signals f_1, f_2 and f_3 are shown in the graphs that belong to Table 5, while those of the signals f_4, f_5 and f_6, are captured in the graphs corresponding to Table 6.

It can be noticed the lesser size order of the individual errors in the case of the network training by means of data obtained through the method proposed, as compared to the one known.

Proposed method												
No.	Input values (q_i, i=1,...,6)						Output values $F(q_i$, i=1,...,6)					
	q1	q2	q3	q4	q5	q6	f1	f2	f3	f4	f5	f6
1	0	0	0	0	0	0	0.00	0	0	0	0	0
2	5	0	0	0	0	0	0.87	0	0	0	0	0
3	10	0	0	0	0	0	1.74	0	0	0	0	0
4	15	0	0	0	0	0	2.59	0	0	0	0	0
5	20	0	0	0	0	0	3.42	0	0	0	0	0
6	25	0	0	0	0	0	4.23	0	0	0	0	0
7	30	0	0	0	0	0	5.00	0	0	0	0	0
8	35	0	0	0	0	0	5.74	0	0	0	0	0
9	40	0	0	0	0	0	6.43	0	0	0	0	0
10	45	0	0	0	0	0	7.07	0	0	0	0	0
11	50	0	0	0	0	0	7.66	0	0	0	0	0
12	55	0	0	0	0	0	8.19	0	0	0	0	0
13	60	0	0	0	0	0	8.66	0	0	0	0	0
14	65	0	0	0	0	0	9.06	0	0	0	0	0
15	70	0	0	0	0	0	9.40	0	0	0	0	0
16	75	0	0	0	0	0	9.66	0	0	0	0	0
17	80	0	0	0	0	0	9.85	0	0	0	0	0
18	85	0	0	0	0	0	9.96	0	0	0	0	0
19	90	0	0	0	0	0	10	0	0	0	0	0
...												
365	90	5	90	5	90	5	10	0.04	10	0.56	10	0.56
366	90	10	90	10	90	10	10	0.15	10	1.11	10	1.11
367	90	15	90	15	90	15	10	0.34	10	1.67	10	1.67
368	90	20	90	20	90	20	10	0.60	10	2.22	10	2.22
369	90	25	90	25	90	25	10	0.94	10	2.78	10	2.78
370	90	30	90	30	90	30	10	1.34	10	3.33	10	3.33
371	90	35	90	35	90	35	10	1.81	10	3.89	10	3.89
372	90	40	90	40	90	40	10	2.34	10	4.44	10	4.44
373	90	45	90	45	90	45	10	2.93	10	5.00	10	5.00
374	90	50	90	50	90	50	10	3.57	10	5.56	10	5.56
375	90	55	90	55	90	55	10	4.26	10	6.11	10	6.11
376	90	60	90	60	90	60	10	5.00	10	6.67	10	6.67
377	90	65	90	65	90	65	10	5.77	10	7.22	10	7.22
378	90	70	90	70	90	70	10	6.58	10	7.78	10	7.78
379	90	75	90	75	90	75	10	7.41	10	8.33	10	8.33
380	90	80	90	80	90	80	10	8.26	10	8.89	10	8.89
381	90	85	90	85	90	85	10	9.13	10	9.44	10	9.44
382	90	90	90	90	90	90	10	10	10	10	10	10

Known method												
No.	Input values (q_i, i=1,...,6)						Output values $F(q_i$, i=1,...,6)					
	q1	q2	q3	q4	q5	q6	f1	f2	f3	f4	f5	f6
1	0.00	47.93	51.70	29.70	29.70	29.70	0	3.3	3.3	3.3	3.3	3.3
2	19.27	47.93	51.70	29.70	29.70	29.70	3.3	3.3	3.3	3.3	3.3	3.3
3	41.30	47.93	51.70	29.70	29.70	29.70	6.6	3.3	3.3	3.3	3.3	3.3
4	90.00	47.93	51.70	29.70	29.70	29.70	10	3.3	3.3	3.3	3.3	3.3
5	19.27	0.00	51.70	29.70	29.70	29.70	3.3	0	3.3	3.3	3.3	3.3
6	19.27	70.12	51.70	29.70	29.70	29.70	3.3	6.6	3.3	3.3	3.3	3.3
7	19.27	90.00	51.70	29.70	29.70	29.70	3.3	10	3.3	3.3	3.3	3.3
8	19.27	47.93	0.00	29.70	29.70	29.70	3.3	3.3	0	3.3	3.3	3.3
9	19.27	47.93	73.12	29.70	29.70	29.70	3.3	3.3	6.6	3.3	3.3	3.3
10	19.27	47.93	90.00	29.70	29.70	29.70	3.3	3.3	10	3.3	3.3	3.3
11	19.27	47.93	51.70	0.00	29.70	29.70	3.3	3.3	3.3	0	3.3	3.3
12	19.27	47.93	51.70	59.40	29.70	29.70	3.3	3.3	3.3	6.6	3.3	3.3
13	19.27	47.93	51.70	90.00	29.70	29.70	3.3	3.3	3.3	10	3.3	3.3
14	19.27	47.93	51.70	29.70	0.00	29.70	3.3	3.3	3.3	3.3	0	3.3
15	19.27	47.93	51.70	29.70	59.40	29.70	3.3	3.3	3.3	3.3	6.6	3.3
16	19.27	47.93	51.70	29.70	90.00	29.70	3.3	3.3	3.3	3.3	10	3.3
17	19.27	47.93	51.70	29.70	29.70	0.00	3.3	3.3	3.3	3.3	3.3	0
18	19.27	47.93	51.70	29.70	29.70	59.40	3.3	3.3	3.3	3.3	3.3	6.6
19	19.27	47.93	51.70	29.70	29.70	90.00	3.3	3.3	3.3	3.3	3.3	10
...												
358	19.27	90.00	90.00	90.00	90.00	90.00	3.3	10	10	10	10	10
359	41.30	90.00	90.00	90.00	90.00	90.00	6.6	10	10	10	10	10
360	90.00	0.00	90.00	90.00	90.00	90.00	10	0	10	10	10	10
361	90.00	47.93	90.00	90.00	90.00	90.00	10	3.3	10	10	10	10
362	90.00	70.12	90.00	90.00	90.00	90.00	10	6.6	10	10	10	10
363	90.00	90.00	0.00	90.00	90.00	90.00	10	10	0	10	10	10
364	90.00	90.00	51.70	90.00	90.00	90.00	10	10	3.3	10	10	10
365	90.00	90.00	73.12	90.00	90.00	90.00	10	10	6.6	10	10	10
366	90.00	90.00	90.00	0.00	90.00	90.00	10	10	10	0	10	10
367	90.00	90.00	90.00	29.70	90.00	90.00	10	10	10	3.3	10	10
368	90.00	90.00	90.00	59.40	90.00	90.00	10	10	10	6.6	10	10
369	90.00	90.00	90.00	90.00	0.00	90.00	10	10	10	10	0	10
370	90.00	90.00	90.00	90.00	29.70	90.00	10	10	10	10	3.3	10
371	90.00	90.00	90.00	90.00	59.40	90.00	10	10	10	10	6.6	10
372	90.00	90.00	90.00	90.00	90.00	0.00	10	10	10	10	10	0
373	90.00	90.00	90.00	90.00	90.00	29.70	10	10	10	10	10	3.3
374	90.00	90.00	90.00	90.00	90.00	59.40	10	10	10	10	10	6.6
375	90.00	90.00	90.00	90.00	90.00	90.00	10	10	10	10	10	10

Table 2: Training data generation

No	Inputs						Outputs					
	q1	q2	q3	q4	q5	q6	f1	f2	f3	f4	f5	f6
1	0.5	0.5	0.5	0.5	0.5	0.5	0.087265	0.000381	0.000309	0.055556	0.055556	0.055556
2	1.5	1.5	1.5	1.5	1.5	1.5	0.261769	0.003427	0.002778	0.166667	0.166667	0.166667
3	2.5	2.5	2.5	2.5	2.5	2.5	0.436194	0.009518	0.007716	0.277778	0.277778	0.277778
4	3.5	3.5	3.5	3.5	3.5	3.5	0.610485	0.018652	0.015123	0.388889	0.388889	0.388889
5	4.5	4.5	4.5	4.5	4.5	4.5	0.784591	0.030827	0.025	0.5	0.5	0.5
...												
86	85.5	85.5	85.5	85.5	85.5	85.5	9.969173	9.215409	9.025	9.5	9.5	9.5
87	86.5	86.5	86.5	86.5	86.5	86.5	9.981348	9.389515	9.237346	9.611111	9.611111	9.611111
88	87.5	87.5	87.5	87.5	87.5	87.5	9.990482	9.563806	9.45216	9.722222	9.722222	9.722222
89	88.5	88.5	88.5	88.5	88.5	88.5	9.996573	9.738231	9.669444	9.833333	9.833333	9.833333
90	89.5	89.5	89.5	89.5	89.5	89.5	9.999619	9.912735	9.889198	9.944444	9.944444	9.944444

Table 3: Testing data

k	Mean square error	
	$netProp_k$	$netKnown_k$
1	2.25E-04	8.37E-02
2	1.83E-04	8.34E-02
3	2.39E-04	8.45E-02
4	1.93E-04	7.88E-02

Table 4: Testing results

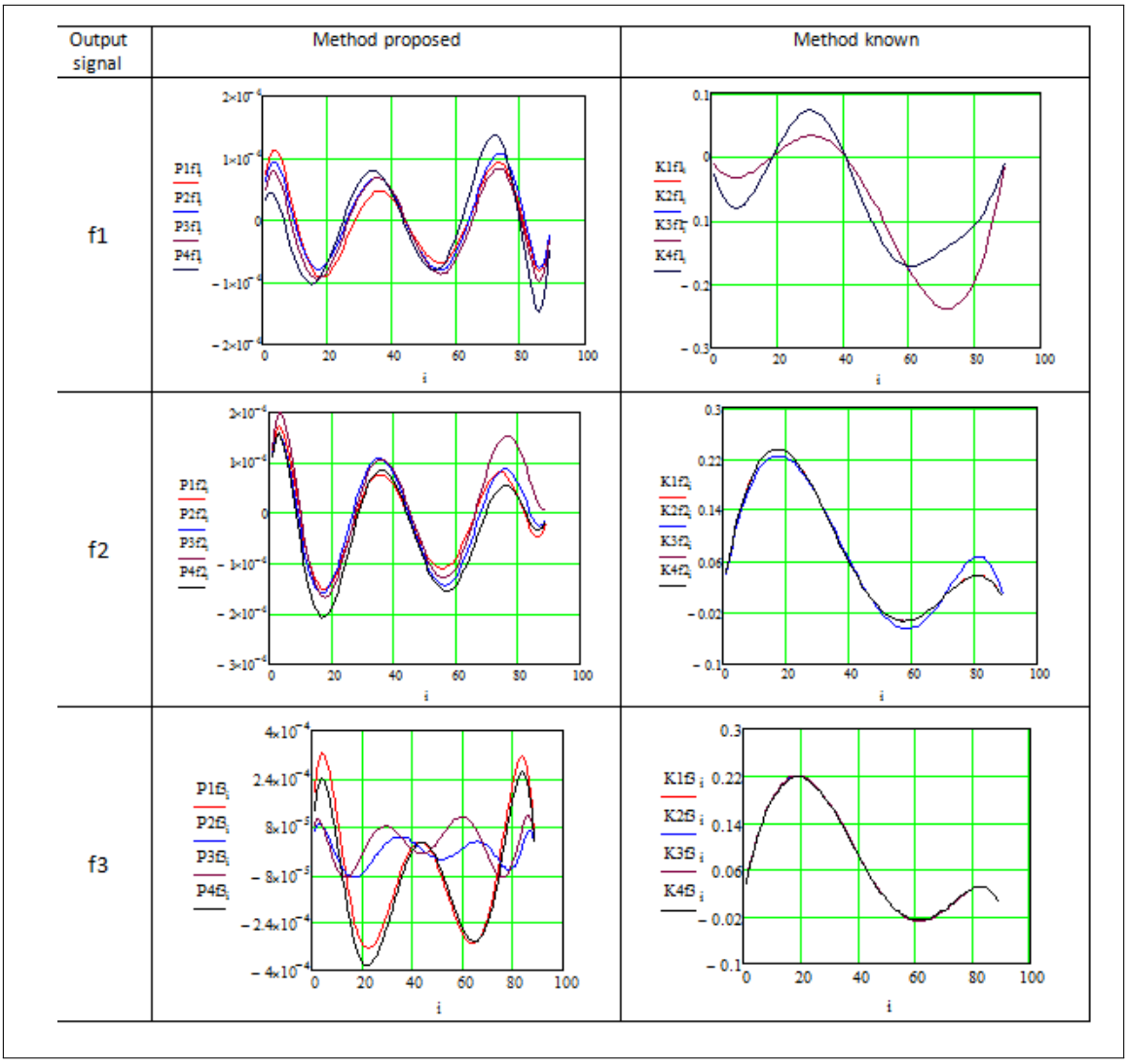


Table 5: The error variation in the case of the signals f_1, f_2, f_3

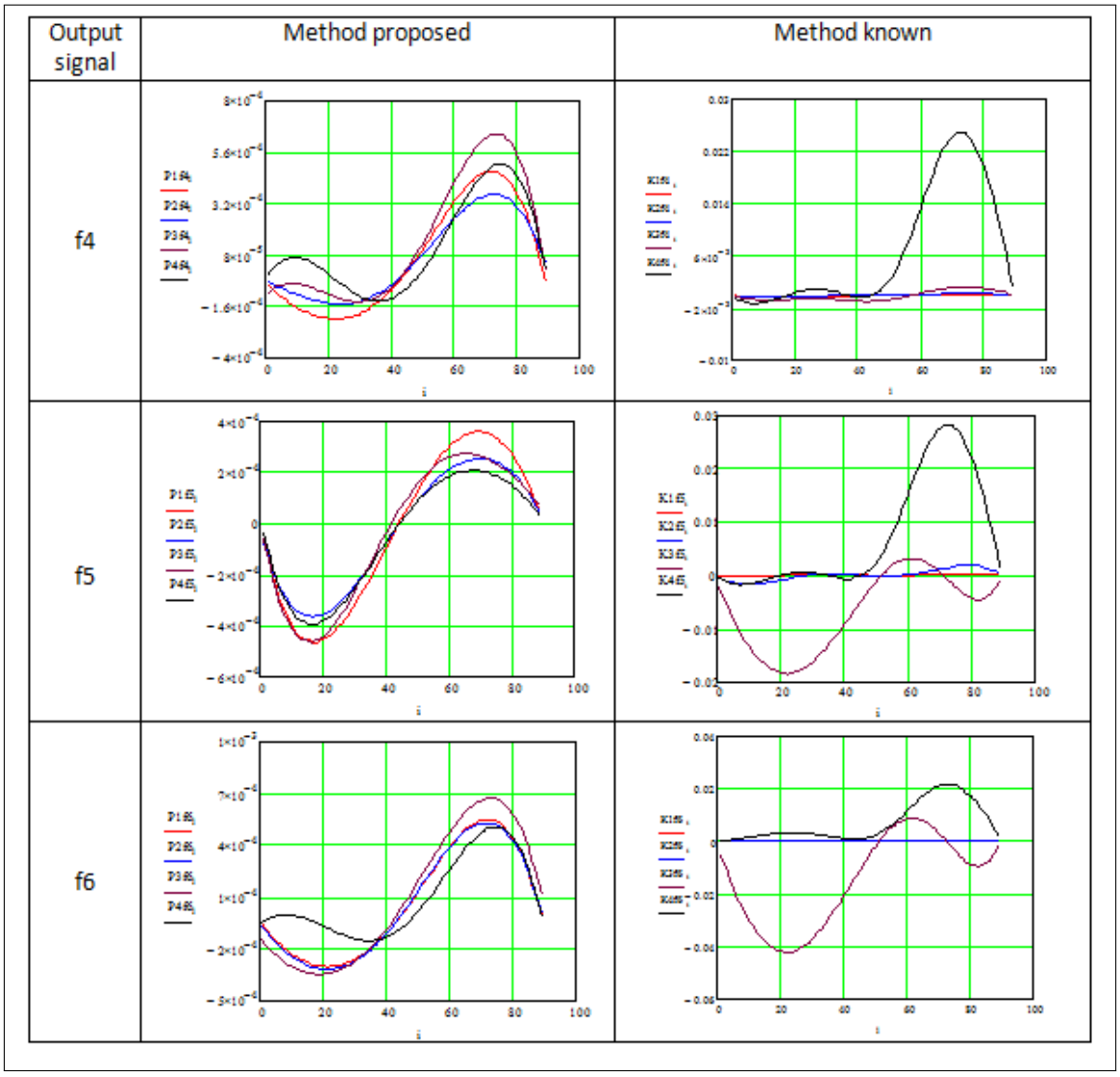


Table 6: The error variation in the case of the signals f_4, f_5, f_6

3 Application for a robot with 6 degrees of freedom

Let us consider the robots workspace as a cube with the side of 300 mm. Based on the algorithm shown in Table 1, there will be generated three sets of training examples for the method proposed here, taking into account in the model for direct kinematic analysis (1)-(2) the following constants:

$$i_{Tx} = i_{Ty} = i_{Tz} = 10; i_{T\psi} = i_{T\theta} = i_{T\varphi} = 3; \tag{13}$$

$$X_0 = 0; Y_0 = 0; Z_0 = 300; \psi_0 = \theta_0 = \varphi_0 = -60^o; \tag{14}$$

The above value of -60^o for the angular variables was chosen so that the coordinates of the joints differs from 0 in the system reference point.

In order to validate the method suggested here, an alternative is considered to be the method of formation the training examples by the evenly distributed choice of the coordinates of the effector in the robot's workspace. For the method known, of evenly distributed choice of the coordinates, there will be generated three sets of data. The training data is obtained from the model for inverse kinematic analysis (3)-(4) with initial values from relation (13)-(14). In order to generate each set of training data (Set 1, Set 2, Set 3) corresponding to the two methods, there has been used the data in Table 7.

Data set		q_1	q_2	q_3	q_4	q_5	q_6	No. of examples n
Set 1	q_{imin}	20	20	20	20	20	20	n=104
	q_{imax}	50	50	50	50	50	50	
	p	10	10	10	10	10	10	
	j	3	3	3	3	3	3	
Set 2	q_{imin}	20	20	20	20	20	20	n=194
	q_{imax}	50	50	50	50	50	50	
	p	5	5	5	5	5	5	
	j	6	6	6	6	6	6	
Set 3	q_{imin}	20	20	20	20	20	20	n=448
	q_{imax}	50	50	50	50	50	50	
	p	2	2	2	2	2	2	
	j	15	15	15	15	15	15	
		X	Y	Z	ψ	θ	φ	
	min	200	200	200	0	0	0	
	max	500	500	500	90	90	90	

Data set	X	Y	Z	ψ	θ	φ	No. of examples n
Set 1	200	200	500	0	0	0	n=3x3x3x4=108
	200	200	500	90	0	0	
	350	350	650	0	90	0	
	500	500	800	0	0	90	
Set 2	200	200	200	0	0	0	n=4x4x4x4=256
	300	300	300	90	0	0	
	400	400	400	0	90	0	
	500	500	500	0	0	90	
Set 3	200	200	500	0	0	0	n=4x4x4x6=448
	300	300	600	45	0	0	
	400	400	700	0	45	0	
	500	500	800	0	0	45	
				90	0	0	
				0	90	0	
				0	0	90	

Table 7: Training data sets

The number of a set's training examples was denoted by n and it acquired different values in the case of each set. Using the data in Tables 1 and 7 there has been generated for each method a set of training data. In order to make a comparison between the results of the two methods, there have been used sets of data having the cardinality of the set of training examples (n) close (Set 1, Set 2) or identical (Set 3).

Table 8 shows in parallel the set of training data for the proposed method, and Set 1 for the known method. There has been modelled a three-layer neural network (6-20-6) and the training with the three sets of data resulted in three neural models for each method. For each set of data and for each method, there have been obtained close errors of training ($e < 10^{-5}$). In order to compare the two methods, the neural networks have been tested for each set and method. For the comparison of the results, there has been taken into account the move of the effector in the robot's workspace on a diagonal between the coordinate points P_1(210, 210, 670) and P_2(490, 490, 790). For the orientation of the effector, there has been considered a rotation $\varphi = 90^o$. The values of the input signals are shown in columns 1-6 in Table 9. The positioning error in Table 9 represents the resultant of the positioning coordinates error (X, Y, Z). Similarly, the angular error represents the resultant of the angular errors (ψ, θ, φ). Table 10 shows graphically the

Proposed method

No.	Stage	Joints coordinates ($q_{i,n}$, i=1,...,6; n=1,...,104)						Effector coordinates $P_n(X,Y,Z,\psi,\vartheta,\varphi)$, n=1,...,104					
t	f	q_1	q_2	q_3	q_4	q_5	q_6	X	Y	Z	ψ	ϑ	φ
1	1	20	20	20	20	20	20	200	200	500	0	0	0
2		30	20	20	20	20	20	300	200	500	0	0	0
3		40	20	20	20	20	20	400	200	500	0	0	0
4		50	20	20	20	20	20	500	200	500	0	0	0
5	2	20	30	20	20	20	20	200	300	500	0	0	0
6		20	40	20	20	20	20	200	400	500	0	0	0
7		20	50	20	20	20	20	200	500	500	0	0	0
	...												
17	6	20	20	20	20	20	30	200	200	500	0	0	30
18		20	20	20	20	20	40	200	200	500	0	0	60
19		20	20	20	20	20	50	200	200	500	0	0	90
20	7	20	35	35	35	35	35	200	350	650	45	45	45
21		30	35	35	35	35	35	300	350	650	45	45	45
22		40	35	35	35	35	35	400	350	650	45	45	45
23		50	35	35	35	35	35	500	350	650	45	45	45
	...												
40	12	35	35	35	35	35	20	350	350	650	45	45	0
41		35	35	35	35	35	30	350	350	650	45	45	30
42		35	35	35	35	35	40	350	350	650	45	45	60
43		35	35	35	35	35	50	350	350	650	45	45	90
44	13	20	50	50	50	50	50	200	500	800	90	90	90
45		30	50	50	50	50	50	300	500	800	90	90	90
46		40	50	50	50	50	50	400	500	800	90	90	90
60	18	50	50	50	50	50	20	500	500	800	90	90	0
61		50	50	50	50	50	30	500	500	800	90	90	30
62		50	50	50	50	50	40	500	500	800	90	90	60
63	19	20	20	35	35	35	35	200	200	650	45	45	45
	...												
74	21	35	35	35	35	50	50	350	350	650	45	90	90
75	22	20	20	20	35	35	35	200	200	500	45	45	45
	...												
82	23	35	35	35	50	50	50	350	350	650	90	90	90
83	24	20	20	20	20	35	35	200	200	500	0	45	45
	...												
90	25	35	35	50	50	50	50	350	350	800	90	90	90
91	26	30	30	30	30	30	30	300	300	600	30	30	30
92		40	40	40	40	40	40	400	400	700	60	60	60
93	27	30	20	30	20	30	20	300	200	600	0	30	0
94		40	20	40	20	40	20	400	200	700	0	60	0
	...												
102	30	50	20	50	20	50	20	500	200	800	0	90	0
103		50	30	50	30	50	30	500	300	800	30	90	30
104		50	40	50	40	50	40	500	400	800	60	90	60

Known method

Stage	Joints coordinates ($q_{i,n}$, i=1,...,6; n=1,...,108)						Effector coordinates $P_n(X,Y,Z,\psi,\vartheta,\varphi)$, n=1,...,108					
f	q_1	q_2	q_3	q_4	q_5	q_6	X	Y	Z	ψ	ϑ	φ
1	20	20	20	20	20	20	200	200	500	0	0	0
2	35	20	20	20	20	20	350	200	500	0	0	0
3	50	20	20	20	20	20	500	200	500	0	0	0
4	20	35	20	20	20	20	200	350	500	0	0	0
5	35	35	20	20	20	20	350	350	500	0	0	0
6	50	35	20	20	20	20	500	350	500	0	0	0
7	20	50	20	20	20	20	200	500	500	0	0	0
8	35	50	20	20	20	20	350	500	500	0	0	0
9	50	50	20	20	20	20	500	500	500	0	0	0
10	20	20	35	20	20	20	200	200	650	0	0	0
...												
19	20	20	50	20	20	20	200	200	800	0	0	0
20	35	20	50	20	20	20	350	200	800	0	0	0
21	50	20	50	20	20	20	500	200	800	0	0	0
22	20	35	50	20	20	20	200	350	800	0	0	0
23	35	35	50	20	20	20	350	350	800	0	0	0
24	50	35	50	20	20	20	500	350	800	0	0	0
25	20	50	50	20	20	20	200	500	800	0	0	0
26	35	50	50	20	20	20	350	500	800	0	0	0
27	50	50	50	20	20	20	500	500	800	0	0	0
28	20	20	20	50	20	20	200	200	500	90	0	0
29	35	20	20	50	20	20	350	200	500	90	0	0
30	50	20	20	50	20	20	500	200	500	90	0	0
...												
51	50	35	50	50	20	20	500	350	800	90	0	0
52	20	50	50	50	20	20	200	500	800	90	0	0
53	35	50	50	50	20	20	350	500	800	90	0	0
54	50	50	50	50	20	20	500	500	800	90	0	0
55	20	20	20	20	50	20	200	200	500	0	90	0
56	35	20	20	20	50	20	350	200	500	0	90	0
57	50	20	20	20	50	20	500	200	500	0	90	0
...												
78	50	35	50	20	50	20	500	350	800	0	90	0
79	20	50	50	20	50	20	200	500	800	0	90	0
80	35	50	50	20	50	20	350	500	800	0	90	0
81	50	50	50	20	50	20	500	500	800	0	90	0
82	20	20	20	20	20	50	200	200	500	0	0	90
83	35	20	20	20	20	50	350	200	500	0	0	90
84	50	20	20	20	20	50	500	200	500	0	0	90
...												
105	50	35	50	20	20	50	500	350	800	0	0	90
106	20	50	50	20	20	50	200	500	800	0	0	90
107	35	50	50	20	20	50	350	500	800	0	0	90
108	50	50	50	20	20	50	500	500	800	0	0	90

Table 8: Set1 of training data for proposed and known method

Testing points	Effector coordinates						Proposed method (PrM)						Known method (KnM)					
							Set 1		Set 2		Set 3		Set 1		Set 2		Set 3	
	$P_n(X,Y,Z,\psi,\theta,\varphi)$, n=1,...,11						Position Error	Angular Error	Position Error	Angular Error	Position Error	Angular Error	Position Error	Angular Error	Position Error	Angular Error	Position Error	Angular Error
Pn							[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
n	X	Y	Z	ψ	θ	φ	0.0833	0.0000	0.0015	0.0000	0.0418	0.0000	1.4620	0.0000	0.1597	0.0000	0.0000	0.0000
1	210	210	510	0	0	0	0.2298	0.0020	0.0093	0.0002	0.0795	0.0028	18.6663	0.1243	6.0701	0.1582	0.0914	0.2883
2	230	230	530	0	0	6.4	0.3284	0.0023	0.0102	0.0007	0.0774	0.0060	31.9270	0.4877	8.5636	0.5283	0.0831	1.1327
3	250	250	550	0	0	12.88	0.4247	0.0017	0.0105	0.0009	0.0426	0.0039	39.0282	1.0491	7.7047	0.9262	0.0408	2.4159
4	270	270	570	0	0	19.32	0.5645	0.0038	0.0144	0.0009	0.0316	0.0047	39.9710	1.7607	5.9324	1.1810	0.0143	3.8805
5	290	290	590	0	0	25.76	0.7743	0.0118	0.0203	0.0010	0.0783	0.0125	35.3201	2.5621	9.6531	1.1573	0.0076	4.8799
6	310	310	610	0	0	32.2	1.0592	0.0265	0.0291	0.0014	0.0917	0.0083	26.4539	3.3790	17.6429	0.7891	0.0145	4.0579
7	330	330	630	0	0	38.64	1.4172	0.0459	0.0471	0.0029	0.0674	0.0115	15.9495	4.1216	26.2353	0.1009	0.0428	0.0715
8	350	350	650	0	0	45.08	1.8491	0.0650	0.0808	0.0060	0.0424	0.0425	10.3252	4.6882	33.4875	0.7905	0.0621	6.4541
9	370	370	670	0	0	51.52	2.3479	0.0769	0.1337	0.0116	0.0767	0.0759	15.6619	4.9747	38.0005	1.6983	0.0619	11.3875
10	390	390	690	0	0	57.96	2.8678	0.0746	0.2087	0.0201	0.1089	0.1033	22.0037	4.8883	38.8002	2.4044	0.0427	12.9225
11	410	410	710	0	0	64.4	3.3076	0.0545	0.3082	0.0322	0.1113	0.1215	24.4418	4.3602	35.3529	2.7036	0.0168	11.6250
12	430	430	730	0	0	70.84	3.5670	0.0221	0.4345	0.0482	0.0981	0.1319	21.6759	3.3542	27.5287	2.4402	0.0170	8.5784
13	450	450	750	0	0	77.28	3.6831	0.0015	0.5897	0.0682	0.1296	0.1390	13.4760	1.8705	15.5068	1.5280	0.0197	4.5242
14	470	470	770	0	0	83.72	3.8700	0.0394	0.7739	0.0919	0.2424	0.1476	1.4620	0.0005	0.1648	0.0002	0.0404	0.0000
15	490	490	790	0	0	90	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0832	0.0000
Standard deviation							1.4209	0.028563	0.2389	0.0282	0.0543	0.058182	12.8225	1.9209	14.2201	0.9163	0.0292	4.5026
Maximum error							3.8700	0.076854	0.7739	0.0919	0.2424	0.147576	39.9710	4.9747	38.8002	2.7036	0.0914	12.9225

Table 9: Simulated results

positioning and angular errors (represented in the graph as Pos. Er. and Ang. Er.) resulted from the simulation of the move between points P_1 and P_2. The results simulated with each neural model (Set 1, Set 2, Set 3) of each method has been marked graphically with PrM for the proposed method and KnM, for the known method.

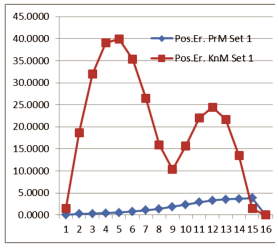
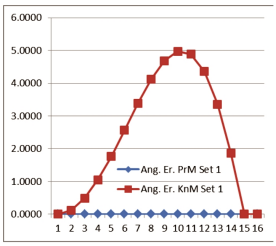
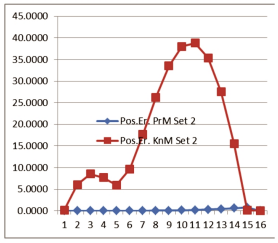
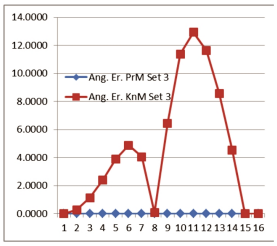
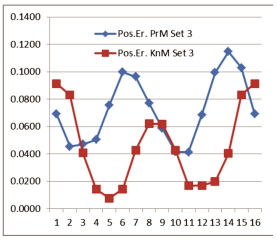
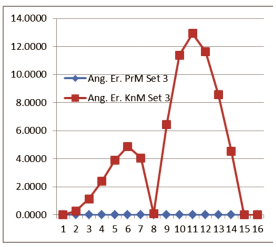
NM	Positioning error	Angular error
Set 1		
Set 2		
Set 3		

Table 10: Graphical representation of the errors resulted when simulating

4 Conclusions

The analysis of the results in Tables 9 and 10 in terms of positioning, angular and cumulated errors demonstrates the following:

- When simulating the move in a straight line, all the three neural models obtained based on the proposed method PrM offer much better results than the models obtained based on the KnM.

- The neural model Set 2-PrM offers positioning precision corresponding to some handling applications ($e < 1$ mm) although the set of training examples is relatively small (n=108). By contrast, even if for the neural model Set 2-KnM there has been used a relatively more significant number of training examples (n=256), this gives greater positioning errors ($e < 39$ mm) which cannot be accepted.

- The neural model Set 3-PrM (n=448) offers a positioning precision corresponding to the majority of the handling applications ($e < 0.25$ mm). The neural model Set 3-KnM (n=448)

offers the best positioning precision ($e < 0.1$ mm), but it does not solve the orientation problem, the angular error being greater than 100.

It has been noted that the neural models generated by the proposed method offer better results as compared to the method of evenly distributed training data in the robot's workspace. The authors find that the proposed method offers superior results, and that it can be used in order to obtain high-quality neural results, with a reduced number of training data.

Bibliography

[1] Campean E.; Itul T.P.; Tanase I.; Pisla A.; Workspace Generation for a 2 - DOF Parallel Mechanism Using Neural Networks, *Applied Mechanics and Materials*, Vol 162 (2012), pp 121-130, doi: 10.4028/www.scientific.net/AMM.162.121.

[2] Ciupan E.; Lungu F.; Ciupan C. (2014); A Method for Training a Neural Network with a Small Number of Examples Used for Robot Control, *ICCCC 2014, 5th International Conference on Computers Communications & Control*, Romania, Oradea, May 6-10.

[3] Dumitrache, I. (2008); From Model-Based Strategies to Intelligent Control Systems, *Proc. of 9th WSEAS International Conference on Automation and Information*, 408-415.

[4] Feng Y.; Wanh Y.; Yang Y. (2012); Inverse Kinematics Solution for Robot Manipulator based on Neural Network under Joint Subspace, *International Journal of Computers Communications & Control*, 7(3):459-472.

[5] Koker R. (2013); A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization, *Information Sciences*, 222, 528-543.

[6] Lewis F. L.; Jagannathan S.; Yesildirak A. (1998); *Neural network control for robots manipulators and nonlinear systems*, Tylor & Francis.

[7] Negrean I.; Vuscan I.; Haiduc N. (1998); *Kinematic and Dynamic Modelling*, Editura Didactica si Pedagogica, Bucuresti, ISBN 973-30-5958-7.

[8] Zilouchian A.; Jamshidi M. (2001); *Intelligent Control Systems using Soft Computing Methodologies*, CRC Press LLC, ISBN 0-8493-1875-0.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):654-666, October, 2015.

Mining Periodic Traces of an Entity on Web

X. Huang, X. Wang, Y. Zhang, J. Zhao

Xinyan Huang
1. Shandong University
2. Shandong University of Finance and Economics
Num 1500, SunHua Road in High Tech Industrial Development Zone
Ji'nan, China
20063462@sdufe.edu.cn

Xinjun Wang*, Yan Zhang, Jinxin Zhao
Shandong University
Num 1500, SunHua Road in High Tech Industrial Development Zone
Ji'nan, China
wxj@sdu.edu.cn, zy@sdu.edu.cn, zjx@sdu.edu.cn
*Corresponding author: wxj@sdu.edu.cn

Abstract: A trace of an entity is a behavior trajectory of the entity. Periodicity is a frequent phenomenon for the traces of an entity. Finding periodic traces for an entity is essential to understanding the entity behaviors. However, mining periodic traces is of complexity procedure, involving the unfixed period of a trace, the existence of multiple periodic traces, the large-scale events of an entity and the complexity of the model to represent all the events. However, the existing methods can't offer the desirable efficiency for periodic traces mining. In this paper, Firstly, a graph model(an event relationship graph) is adopted to represent all the events about an entity, then a novel and efficient algorithm, TracesMining, is proposed to mine all the periodic traces. In our algorithm, firstly, the cluster analysis method is adopted according to the similarity of the activity attribute of an event and each cluster gets a different label, and secondly a novel method is proposed to mine all the Star patterns from the event relationship graph. Finally, an efficient method is proposed to merge all the Stars to get all the periodic traces. High efficiency is achieved by our algorithm through deviating from the existing edge-by-edge pattern-growth framework and reducing the heavy cost of the calculation of the support of a pattern and avoiding the production of lots of redundant patterns. In addition, our algorithm could mine all the large periodic traces and most small periodic traces. Extensive experimental studies on synthetic data sets demonstrate the effectiveness of our method.
Keywords: Event, Periodic Trace, Pattern.

1 Introduction

An event is something that happens at some specific time. Nowadays, due to the popularity of the internet, an increasing number of events about an entity are reported on web every day. However, since these events are usual scattered and redundant, meaningful information can not be extracted by people, such as a behavior trajectory of an entity which is named a trace.

Periodicity is one of the most common phenomena that these traces show, such as the launch of new products, the product promotion and so on, which are named periodic traces by us. In addition, a periodic trace can be loosely defined as the repetitive events series with several kinds of event relationship between them, such as causal, part of and following relationship etc. Periodic traces can provide an insightful and concise explanation over the long development history of one entity, which are very valuable in the prediction of future events' happening of an entity.

Copyright © 2006-2015 by CCC Publications

Unfortunately, mining periodic traces from an entity's long and noisy history data is a challenge, which includes the following major issues:

Firstly, the period of a trace are usually unfixed. However, in previous work, Li Z [8,10] studied the frequent periodic behaviors for moving objects with the period fixed. Unfortunately, the fixed period is unfit for the periodic trace. For example, the period of the launch of new products is usually varying with the changes of the competitive environment or others factors.

Secondly, the model to represent so large events set is a complex issue. Considering the big scale of the events and the relatively complex relationship between them, a big graph is employed, in which the events are represented as vertices and the events relationship (such as, causal relationship, following relationship etc) are represented as edges. And the big graph is named the event relationship graph of the entity.

At the same times, both the structure of periodic traces and the mining of periodic traces become more complex. The problem of mining periodic traces is transformed into mining frequent subgraph from a big graph. For nowadays, the general methods of frequent subgraph mining from a single graph are based on edge-by-edge (i.e., incremental) pattern-growth [12], which are more fit for a small graph than our big graph with several thousand vertices.

In this paper, the problem of periodic traces mining is solved through the following steps:

Firstly, the cluster analysis method is adopted according to the similarity of the activity attribute of an event and each cluster gets a different label. Secondly, a novel method is proposed to mine all the vertice-edge-vertice patterns from the event relationship graph firstly and then all the Star patterns are mined based on the preceding patterns. Finally, an efficient method is proposed to merge all the Stars to get all the periodic traces of the entity.

In our paper, we address an important problem of mining periodic traces. In our algorithm, high efficiency is achieved through deviating from the existing edge-by-edge pattern-growth framework. At the same time, the heavy cost of the calculation of the support of a pattern is reduced and the production of lots of redundant patterns can be avoided. In addition, our algorithm could mine all the large periodic traces and most small periodic traces.

The rest of the paper is organized as follows. In Section 2, some related work is discussed. Section 3 gives the problem formulation. Section 4 and section 5 provides an outline of our algorithm to discover the periodic traces. We report our experimental results in Section 6, conclude our study in Section 7.

2 Related Work

At present, some studies [20] about how to organize events and events relationship have been investigated. However, less studies aim to mine some meaningful information from the events data of entities.

An event relationship migration graph was proposed by Zhaoman Zhong in [1] to organize events, in which a typical event is defined as a behavior sequence or a series of state changes.

A concept of event network is proposed by Zongtian Liu [3], which is used to organized the events and the relationship between them in an article.

Christopher C. Yang [4] introduces the concept of event evolution graph in which events are organized by the time sequence, which can help to efficiently browse the whole of event evolution process.

Heng Ji [5] proposes to identify the "centroid entities" which are frequently involved in events and then link the events involving the same centroid entity along a time line. However, no more studies are done based on the work ahead.

Zhenhui Li proposed an concept of periodic behavior in [8,10], which is mined from the spatiotemporal data over a long histoy and propose a two-stage algorithm, Periodica, to detect

the periods in complex movements and to mine periodic movement behaviors.

Similar to method in literature [3], all events of an entity is firstly linked according to three types of relationships between them. But the difference is that their goal is to create an event ontology but not to mine the meaningful information on the event ontology. However, our main goal is to organize the events of an entity for a long history and to mine the meaningful periodic traces, so much more vertices are involved than [3].

The model of a periodic trace adopted by us is a graph structure which is different from surprising periodic patterns [13] and the periodic behavior in [8,10] which is characterized as a probabilistic model.

SUBDUE [15] is probably the most well-known algorithm for mining frequent subgraphs in a single graph, however, it tends to mine small patterns with high frequency.The SpiderMine algorithm in [7] is proposed to mine top-K largest frequent patterns from a single massive network. SUBDUE and SpiderMine are both approximate algorithms, which aren't fit for our problem to mine all the periodic traces.Although MoSS is an algorithm to mine the complete pattern set, it suffers from a significantly runtime Complexity issue as the input graph size grows [7,17]. So we propose an efficient method fit for our problem. Our algorithm is to mine the complete pattern set, which achieves its efficiency through reducing the heavy cost of calculation of the support of a pattern and avoiding the production of lots of redundant patterns.

3 Problem Formulation

All the following work is based on the assumption that the event relationship graph $G = (V, E)$ has been established already.

The mining of periodic traces is to mine the periodic events series with three kinds of event relationship from the events data of an entity. With the help of the event relationship graph, the main task of periodic traces mining is now transformed to mine the frequent subgraph from the event relationship graph.

We formally define the notions related with the mining of periodic traces as follows:

Definition 1 (Event)**.** An event is something that happens at some specific time, and often some specific place, which is usual a phrase or sentence in the web pages including an activity which is the main word most clearly expressing an event occurrence. Similar to [2], in our paper, the event of an entity is defined as a model E with five attributes, described as follows:

$$E\langle \text{subject}, \text{activity}, \{\text{object}\}, \text{time}, \{\text{location}\}\rangle.$$

Among them, subject, activity and time elements are required.

Definition 2 (Event Relationship)**.** The dependent relationship from an event to another is named an event relationship. There are three kinds of relationships that we considered in this paper, which are causal relationship [9], part of relationship and following [19] relationship.

For these three kinds of relationships, we consider an event relationship as a logical dependency between two events.

If an event e_1 is a component of an event e_2, then there is a part of relationship between e_2 and e_1.

Besides, if the occurrence of an event e_2 always depends on the occurrence of an event e_1, then there must be a causal relationship or a following relationship from e_1 to e_2. If e_1 surely leads to the occurrence of e_2, then there is a causal relationship between e_1 and e_2. The rest are following relationships in which e_2 always occurs after e_1.

Definition 3 (Event Relationship Graph)**.** Event relationship graph is to link all the events of an entity according to the relationship between them. An event relationship graph is denoted as a directed graph $G = (V, E)$, where V are the vertices representing events and E are the directed edges representing event relationships. a fragment of an event relationship graph is shown in Fig 1, in which R_c represents causal relationship, R_f represents part of relationship and R_p represents part of relationship.

Definition 4 (Trace)**.** A trace is denoted as a directed graph $G^* = (V^*, E^*)$, which represent a behavior trajectory of the entity.

Definition 5 (Periodic Trace)**.** A periodic trace is also denoted as a directed graph $G\prime = (V\prime, E\prime)$ composed of periodicly happening events as the vertices $V\prime$ and periodicly happening event relationships as the directed edges $E\prime$ between vertices $V\prime$, where $V\prime \in V$, $E\prime \in E$. And the support of $G\prime$ in G is beyond σ, σ is the minimum support that we set.

The problem of mining periodic traces is to mine all the periodic traces $G\prime = (V\prime, E\prime)$ from an event relationship graph G, and the occurrence of $G\prime$ in G is beyond σ.

We will solve the problem in the following sections.

4 Find All the Event Classes

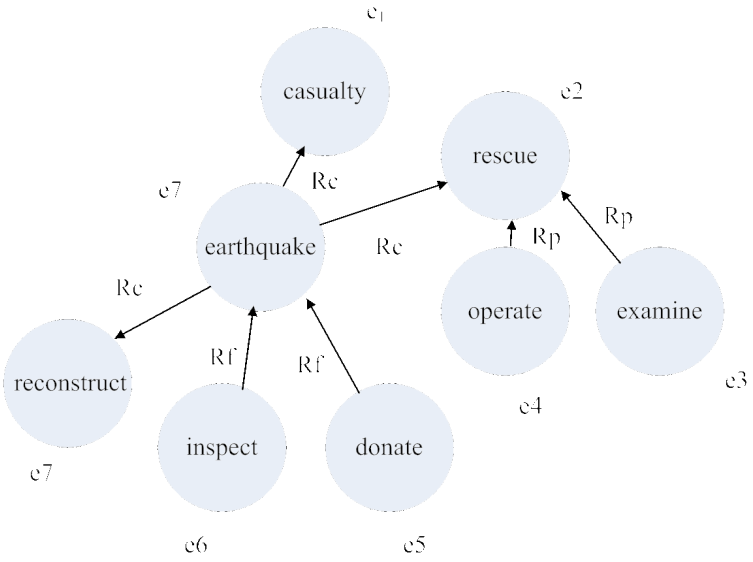


Figure 1: A fragment of an event relationship graph

A fragment of an event relationship graph is shown in Fig 1, in which all the events is represented by it's activity attribute. In this section, our main goal is to find all the event classes $EC = \{ec_1, ec_2, \ldots, ec_D\}$ from all the events, ec_i is an event class in which all the events has the similar activity and is represented like $ec_i = \{e_2, e_6, e_j, \ldots, e_n\}$.

In this section, a clustering method is employed. With the help of wordnet, all the events are clustered into groups according to semantic similarity of activity attribute of them, because the activity attribute could mostly express an event, hence the same kind of events get together and get an identical label, such as A, B etc. We name the events with the similar activity as an event class. For example, a launch event class, a promotion event class etc.

Then the event relationship graph is turned to a labeled directed graph, in which there are three kind of edges labeled as 1, 2, 3 etc.

5 Results and Discussion

In this section, we will describe the periodic trace mining algorithm, TracesMining, which could mine all the periodic traces from the event relationship graph.

The main algorithm of mining periodic traces is shown in Algorithm 1.

Algorithm 1 The periodic traces mining algorithm

Require: input all the event classes $EC = \{ec_1, ec_2, \ldots, ec_D\}$, the event relationship graph G of an entity, support threshold σ.

Ensure: all the periodic traces S.

1: Initialize $S \leftarrow \Phi$;
2: $T\prime \leftarrow \text{MinsMining}(G, EC, \sigma)$;
3: `/* mine all the minimum patterns- vertice-edge-vertice;*/`
4: $S\prime \leftarrow \text{StarsMining}(G, T\prime, \sigma)$;
5: `/* mine all the patterns-Stars based on` $T\prime$`;*/`
6: $S \leftarrow \text{StarsMerging}(S\prime, \sigma)$;
7: `/* Merge all Stars whenever possible*/`
8: Return S;

5.1 Mining All the Frequent Patterns of Vertice-Edge-Vertice

In the MinsMining(G, EC, σ) algorithm, for every two event classes,we should get all the patterns of vertice-edge-vertice and all their corresponding instances in G, here an instance of an pattern is just a physical occurrence of the pattern.

As is shown in Fig 2, $A \xrightarrow{R_f} B$ is a vertice-edge-vertice pattern, the frequency that the physical events in one event class A has the event relationship R_f with the physical events in another B in the event relationship graph G must be beyond the threshold σ, R_f is one of the three relationships we defined ahead.

The generation of a pattern $A \xrightarrow{R_f} B$ and it's instances is illustrated in Fig 2, in which e_i refers to an physical event, ec_i refers to an event class the label of which is B and σ is set to 2.

5.2 Mining All the Patterns of Stars

In the algorithm 1, the StarsMining$(G, T\prime, \sigma)$ algorithm is to generate all the patterns of Stars and their corresponding instances based on all the veritice-edge-vertice patterns produced ahead.

Just as Fig 3 illustrated, the so called Star is just a pattern the number of occurrence of which in the event relationship graph is beyond σ and in which a vertice serves as the center, 1 is the radius, the edge represent one of three kinds of event relationships and the direction of edge is optional which can point to or be away from the center. In Fig 3, the label of e_1 and e_2 is B, the corresponding label of e_7 and e_8 is D etc.

Then we propose an efficient method to mine all the stars based on all the vertice-edge-vertice patterns, as shown in Algorithm 2.

In the algorithm 2, in line 5, the function find(C, L_i) is to find all the vertice-edge-vertice pattern that have a vertice with the label L_i. In line 10, the function Count$(T[i], T[j])$ is to

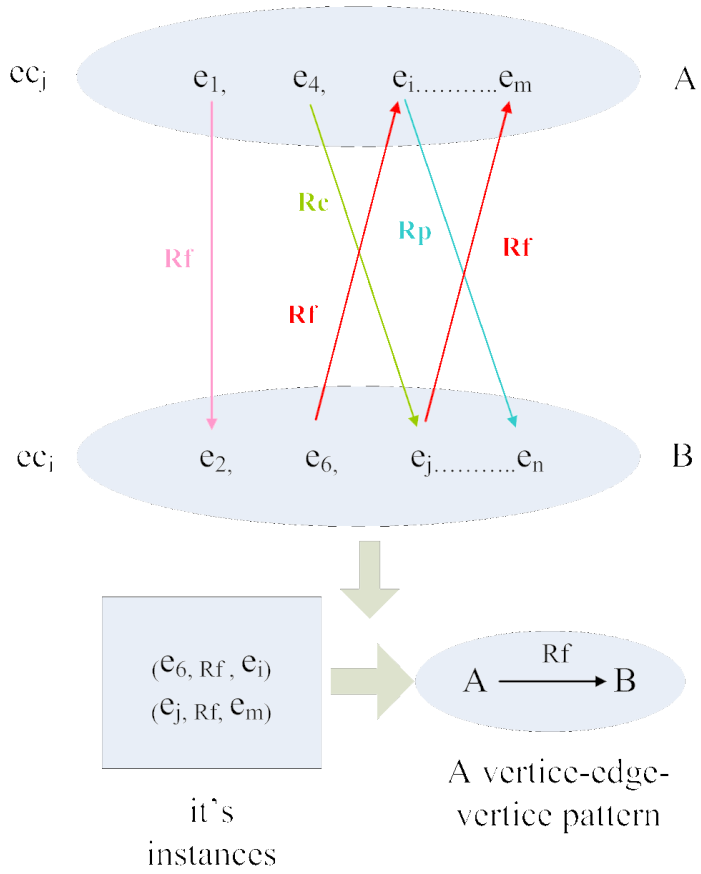


Figure 2: Generation of a vertice-edge-vertice pattern and it's instances

count the number of the same physical events that the label Li in T[i] and T[j] stands for and judge whether the number is beyond σ. In line 13, the function Merge$(T[i], T[j], T\prime)$ is to merge $T[i]$ and $T[j]$ if the number is beyond σ. In line 14, the function Check$(T, S\prime)$ is to put all the patterns which has never been merged into $S\prime$. Ultimately, the $S\prime$ stores all the stars centered by Li, for example, the stars centered by label A are named like Star(A1), Star(A2) etc according to the different physical events set that the label A stands for. In the algorithm 2, we shall finally find all the Star patterns.

In this paper, Stars could help efficiently mine the periodic traces due to the following reasons:

1. Stars reduce the heavy cost of the calculation of support of a pattern.
   Judging whether two patterns can be merged, what we only need to do is to determine which Star the merging bases on and further confirm the number of the same physical events that the center label of the Star stand for. The number is the support of the merged pattern, which could greatly reduce the heavy cost of the calculation of support of a pattern.

2. Stars reduce the complexity of merging.
   The inherent character of Stars with one label as the center makes the merging of pattern convenient. To merge a star, a pattern is just to judge whether one of the borders of the pattern can merge with the star. And the merging makes the patterns grow by Stars

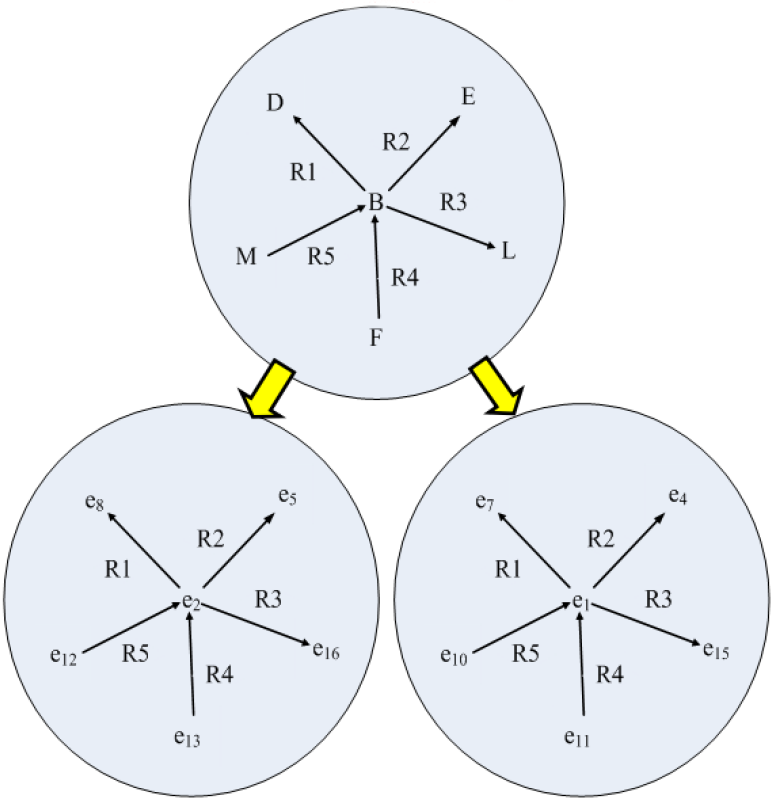


Figure 3: A Star pattern and it's instances

deviating from the existing edge-by-edge pattern-growth way [14], which greatly improve the efficiency of merging.

The last and most important step of generating the periodic traces is to merge the Stars, details is shown in 5.3.

5.3 Generating Periodic Traces

To discover all the periodic traces, the last step is to merge all the Stars. Then an effecient method StarsMerging() is proposed to merge all the Stars that can be merged until no more frequent patterns can be found. Although the number of underlying periodic traces is usually unknown, the algorithm StarsMerging() could at the same time determine the optimal number of periodic traces while generating all the periodic traces, details shown in. Algorithm 3.

S is implemented as a queue into which all the Stars are put. Star(S) (line 3) points to the current Star to be dealt with. p.border (line 6) points to all the boundaries of the pattern p. The function compare(b, s) (line 10) is to count the number of the same physical events that the vertice b and the center of s stand for.

At line 4–13, each current Star as far as possible merges the Stars in S through checking it's borders one by one until no stars can be merged to. The function cleanUp(S, σ) (line 21) is employed to clean all the stars that has been merged and the number of remainder physical events of the center vertice is not beyond σ from S.

Compared with the algorithm in literature [7], our algorithm is more effective and efficient. In Our algorithm, a pattern with 4 Stars just need 3 merging without additional patterns generated, which does not generated any redundant patterns and does not need any additional merging,

Algorithm 2 The StarsMining algorithm

Require: input all the vertice-edge-vertice pattern $C\{C_1, C_2, C_3, \ldots, C_i, \ldots, C_e\}$, all the instances of C, all the labels $L\{L_1, L_2, L_3, \ldots, L_i, \ldots, L_f\}$ appears in C, support threshold σ.

Ensure: all the Stars S.

```
1: Initialize S ← Φ;
2: for i = 1 to f do
3:   S′ ← Φ;
4:   flag = false;
5:   T ← find(C, L_i)
     /*find all the vertice-edge-vertice patterns that have a vertice with the
     label Li*/
6:   repeat
7:     T′ ← Φ;
8:     for i = 0 to T.length do
9:       for j = i + 1 to T.length do
10:        n ← Count(T[i], T[j]);
           /*to count the number of the same physical events that the label Li
           in T[i] and T[j] stands for*/
11:        if n ≥ σ then
12:          flag = ture;
13:          Merge(T[i], T[j], T; );   /*add the merged pattern to T′*/
14:        end if
15:      end for
16:    end for
17:    Check(T, S′)
       /*all the pattern in which the number of the physical events that label
       Li stand for is beyond σ and which hasn't been merged are put into S′*/
18:    T = T′
19:  until flag  /* the merger is completed*/
20:  S′ = S′ ∪ T;
21:  S = S ∪ S′;
22: end for
23: return S
```

while the algorithm in [7] averagely needs 5 additional patterns generated and needs 6 merging. With the number of Stars growing, more obvious advantages are shown in our algorithm.

More details of the comparison with [7] are shown in Section 6.

6 Experiments

In this section, we performed extensive experiments to evaluate the performance of our algorithm of mining the periodic traces on synthetic data. All experiments were done on a 2.8GHz Intel Pentium IV PC with 1GB main memory, operating system Windows XP. Our algorithm is implemented in Python 2.7.

The mining of periodic traces is experimented on a single graph which is generated by the Erdős - Rényi random network model. The Erdős - Rényi model is a well-known model to generate random graphs. Using the G(n, p) function, our synthetic single graph is constructed.

Algorithm 3 StarsMerging

Require: All the Stars queue S, support threshold σ.
Ensure: The periodic traces $S\prime$.

```
1: S' ← Φ;
2: while S.length > 0 do
3:   p = Star(S);  /* select one star from S*/
4:   while true do
5:     flag = false;  /*a flag whether merge happened*/
6:     border ← p.border;  /*check the borders of p one by one*/
7:     for b in border do
8:       ss = lookupS(b);  /*find all the stars of which the label of the center
         vertice is b*/
9:       for s in ss do
10:        n = compare(b, s)
           /*count the same physical events that the vertice b and the center
           vertice of s stand for*/
11:        if n > σ then
12:          p.merge(s);
13:          flag = true;
14:        end if
15:      end for
16:    end for
17:    if !flag then
18:      break;
19:    end if
20:  end while
21:  S' = S' ∪ p;  /*put these patterns that don't grow to S'*/
22:  cleanUp(S, σ);
23: end while
24: return S'
```

However, the graph generated by the Erdős - Rényi model is an undirected graph, which is different from our event relationship graph which is a directed graph. So some changes made based on the Erdős - Rényi model is that 0.5 probability is adopted to decide the direction of the edge. Besides, the labels of vertices and the labels of edges are distributed randomly under the condition that labels of any two adjacent vertices can't be identical.

In the experiment, in order to calculate the recall and precision ratio, a set of large patterns as well as a set of small patterns are injected into the graph. Our goal is to find all the large frequent patterns and all the small patterns from the big graph. Nowadays despite lots of studies in graph mining, few algorithms are capable of the mining task in a big single graph due to the exponentially high combinatorial complexity and support computation. And since Spidermine [7] has been compared with the other algorithms (SEuS [16] (version 1.0), MoSS [17] (version 5.3) and ORIGAMI [18]) and shows a tremendous advantage in efficiency and effectiveness, we just compare our algorithm with Spidermine and the well-known SUBDUE [15] (version 5.2.2), which is a classic approximate algorithm on a single graph.

Firstly, we generate 4 different data sets (labeled Data 1 to 4) with varied parameter settings referring to [7]. The description of the data sets is given in Fig 4. The details of the parameters is given as follows. $|V|$ is the number of vertices. l_V is the number of vertice labels and le is

Data	$\|V\|$	l_V	l_e	d	m	$\|V_L\|$	e_L	n	$\|V_S\|$	e_S
1	400	70	3	2	5	30	2	5	3	2
2	600	130	3	4	5	30	2	20	3	2
3	1000	250	3	4	5	30	2	5	3	20
4	1500	350	3	4	10	30	2	10	3	2

Figure 4: Data Sets

the number of edge labels. d is the average degree. $|V_L|$ (or $|V_S|$) is the number of vertices of each injected large (resp. small) pattern. m (or n) is the number of large (resp. small) patterns injected. e_L (or e_S) is the number of instances of each large (resp. small) pattern injected.

We implement the SpiderMine [7] according to it's algorithms in python. But a little changes are made to SpiderMine, because SpiderMine is designed for undirected graph without labels for it's edges while our graph is directed with labeled edges. At same times, SpiderMine is to mine top K large patterns from a big graph. Here we set $K = 5$, Dmax $= 4$.

Figs 5 to 8 show the distribution of patterns mined by SpiderMine, SUBDUE and our algorithm TracesMining for different data sets in Fig 4. The minimum support threshold is set to 2.

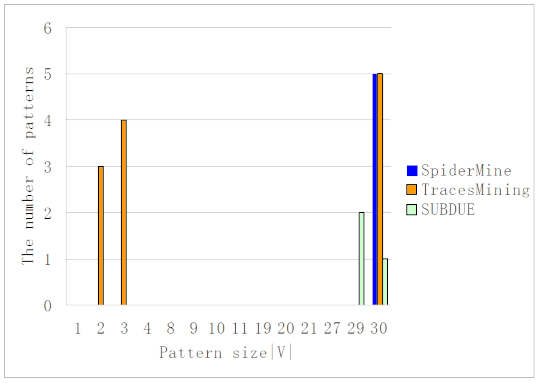


Figure 5: Data 1

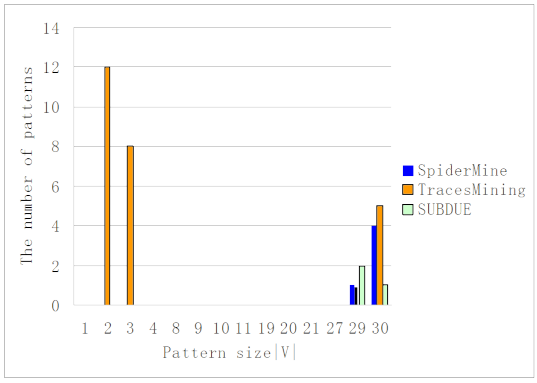


Figure 6: Data 2

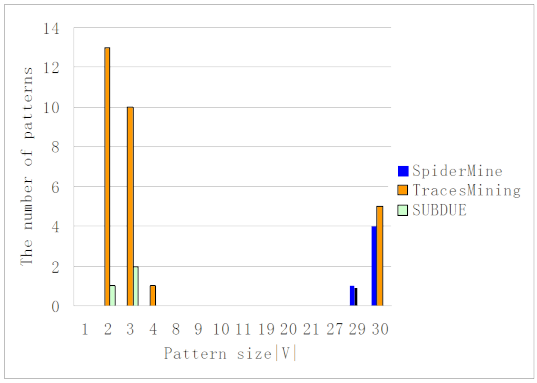


Figure 7: Data 3

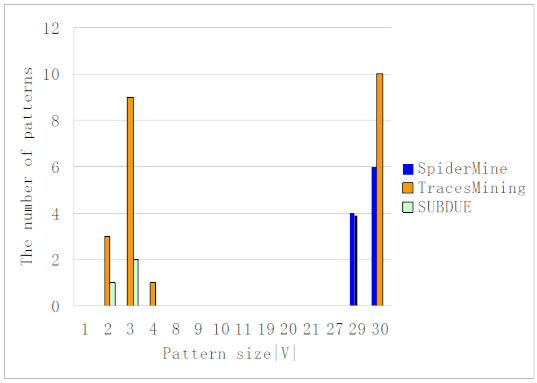


Figure 8: Data 4

In Figs 5, 6, 7 and 8, we can see that compared with SpiderMine() and SUBDUE, our algorithm, TracesMining could mine all the large periodic traces and most small periodic traces injected into the graph.While SpiderMine() focus on the large patterns in four different data sets and SUBDUE tends to mine smaller patterns .

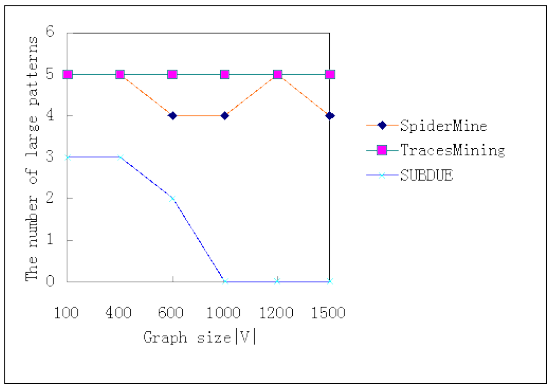


Figure 9: Large Pattern

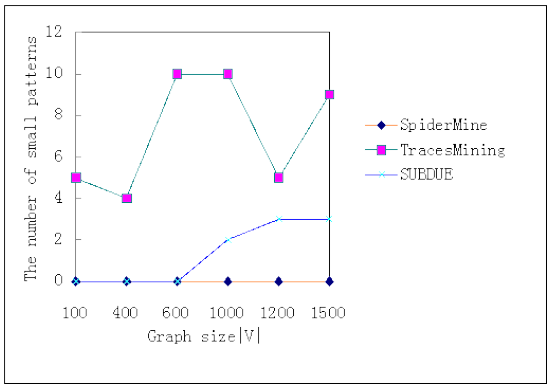


Figure 10: Small pattern

Run Time (seconds)		
Data	TracesMining	SpiderMine
1	0.686	0.704
2	2.248	3.951
3	10.089	13.041
4	13.190	16.272

Figure 11: Run Time

In Figs 9 and 10, TracesMining shows a tremendous advantage compared with SpiderMine and SUBDUE in both large and small patterns mining. SpiderMine is good at mining large patterns and SUBDUE tends to mine smaller patterns with the growing of graph, which fully illustrate that SpiderMine() and SUBDUE don't suit our task to find all the periodic traces.

Since SpiderMine has compare the run time with SUBDUE in [7], the runtime comparison of SpiderMine and TracesMining is shown in Fig 9 on the three data sets. We can see that in runtime, our algorithm has a clear advantage over SpiderMine.

7 Conclusion

In this paper, we address an important and difficult problem: Mining Periodic Traces of an entity on web. We propose a novel and efficient framework to solve the aforementioned problem. Our algorithm achieves its efficiency through deviating from the existing edge-by-edge pattern-growth framework and reducing the heavy cost of the calculation of the support of a pattern and avoiding the production of lots of redundant patterns. In addition, our algorithm could mine all the large periodic traces and most small periodic traces. Experiments demonstrate the efficiency as well as scalability of our algorithm.

Acknowledgement

The first author acknowledges the support by the Natural Science Foundation of China (No. 61303005, No. 61303089, No. 61202151, No. 61103117).

Bibliography

[1] Z. Zhong, Z. Liu, Z. Wen(2009), The Model of Event Relation Representation, *Journal Of Chinese Information Processing*, 23(6): 56–60.

[2] Z. Zhong, C. Li (2013), Web News Oriented Event Multi-Elements Retrieval, *Journal of Software*, 24(10):2366–2378.

[3] Z. Liu, M. Huang, W. Zhou(2009), Research on Event-oriented Ontology Model, *Computer Science*, 36(11): 191–195.

[4] C.C. Yang, X. Shi, C.P. Wei (2009), Discovering event evolution graphs from news corpora, *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(4): 850–863.

[5] H. Ji, R. Grishman, Z. Chen et al (2009), Cross-document Event Extraction and Tracking: Task, Evaluation, Techniques and Challenges, *RANLP*, 166–172.

[6] W. Liu, D. Wang, W. Xu et al (2012), A Sub-topic Partition Method based on Event Network, *The Seventh International Conference on Internet and Web Applications and Services*, 194–199.

[7] F. Zhu, Q. Qu, D. Lo et al(2011), Mining top-k large structural patterns in a massive network, *Proc. of the VLDB Endowment*, 4(11): 807–818.

[8] Z. Li, J. Han, B. Ding et al (2012), Mining periodic behaviors of object movements for animal and biological sustainability studies, *Data Mining and Knowledge Discovery*, 24(2): 355–386.

[9] S. Bethard, J.H. Martin (2008), Learning semantic links from a corpus of parallel temporal and causal relations, *Proc. ACL-HLT*, 177–180.

[10] Z. Li, J. Han, M. Ji et al (2011), Movemine: Mining moving object data for discovery of animal movement patterns, *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4): 37.

[11] Z. Guan, X. Yan, L.M. Kaplan (2012), Measuring two-event structural correlations on graphs, *Proceedings of the VLDB Endowment*, 5(11): 1400–1411.

[12] L. Gao, G.-M. Qin Gui, X.-F. Zhou (2008), An Overview of Algorithms for Mining Frequent Patterns in Graph Data, *Acta Electronica Sinica*, 36(8): 1603–1609.

[13] J. Yang, W. Wang, S.Y. Philip (2008), Mining surprising periodic patterns, *Data Mining and Knowledge Discovery*, 9(2): 189–216.

[14] M. Wörlein, T. Meinl, I. Fischer et al(2005), *A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston*, Springer Berlin Heidelberg.

[15] L.B. Holder, D.J. Cook, S. Djoko (1994), Substucture Discovery in the SUBDUE System, *KDD workshop*, 169–180.

[16] S. Ghazizadeh, S.S. Chawathe (2002), SEuS: Structure extraction using summaries, *Discovery science*, Springer Berlin Heidelberg, 71–85.

[17] M. Fiedler, C. Borgelt (2007), Support Computation for Mining Frequent Subgraphs in a Single Graph, *MLG*.

[18] M. Al Hasan, V. Chaoji, S. Salem et al(2007), Origami: Mining representative orthogonal graph patterns, *ICDM*, 153–162.

[19] Z. Li, F. Wu, M.C. Crofoot (2013), Mining Following Relationships in Movement Data, *ICDM*, 458–467.

[20] I.C. Resceanu, G.C. Călugăru, C.F. Resceanu et al(2012), Cooperative Robot Structures Modeled After Whale Behavior and Social Structure, *International Journal of Computers Communications & Control*, 7(5): 945–956.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):667-677, October, 2015.

Advisory, Negotiation and Intelligent Decision Support System for Leadership Analysis

R. Gudauskas, A. Kaklauskas, S. Jokubauskiene
V. Targamadze, L. Budryte, J. Cerkauskas, A. Kuzminske

Renaldas Gudauskas, Loreta Budryte
Martynas Mazvydas National Library of Lithuania
Gedimino pr. 51, LT-01504 Vilnius, Lithuania
r.gudauskas@lnb.lt, l.budryte@lnb.lt

Arturas Kaklauskas*, Justas Cerkauskas, Agne Kuzminske
Vilnius Gediminas Technical University
Sauletekio al. 11, LT-10223 Vilnius, Lithuania
arturas.kaklauskas@vgtu.lt, justas.cerkauskas@vgtu.lt, Agne.kuzminske@vgtu.lt
*Corresponding author: arturas.kaklauskas@vgtu.lt

Saule Jokubauskiene, Vilija Targamadze
Vilnius University
Universiteto g. 3, LT-01513 Vilnius, Lithuania
saule.jokubauskiene@kf.vu.lt, vilija.targamadze@fsf.vu.lt

Abstract: The development of the Leader Model for quantitative and qualitative analyses began with the goal of integrating managerial, organizational, technical, technological, economic, legal/regulatory, innovative, social, cultural, ethical, psychological, religious, ethnic and other aspects involved in the process of a leaders life cycle. The need to determine the most efficient life cycle of a leader led to the development of the Advisory, Negotiation and Intelligent DEcision support System for Leadership Analysis (ANDES). The objective of the authors of this work for integrating text analytics, advisory, negotiation and decision support systems is to improve the quality and efficiency of intelligent decision-making regarding a leaders life cycle. This ANDES consists of an intelligent database, database management system, model-base, model-base management system and user interface.
Keywords: Leader, Model, Intelligent Systems, Integration.

1 Introduction

Leadership, according to Uhl-Bien and Russ Marion [1], is multi-level, processable, contextual, and interactive. Today's organizational leaders are faced with unprecedented complexity in the wake of increasing globalization [2].

Various systems are being developed globally for leadership analysis. These systems include information [3, 4], intelligent [5, 6], knowledge [7, 8], expert [9, 10] and decision support [11–17].

The aforementioned systems analyze their managerial, organizational, technical, technological, economic, legal/regulatory, innovative and similar aspects. However, neither the integrated economic, legal/regulatory, technical, technological, organizational, managerial, quality of life, social, cultural, political, ethical nor psychological aspects are generally paid hardly any attention at all.

The structure of this paper is as follows: Section 2, which follows this introduction, analyses the Advisory, Negotiation and Intelligent Decision Support System for Leadership Analysis. Certain concluding remarks appear in Section 3.

Copyright © 2006-2015 by CCC Publications

2 Advisory, Negotiation and Intelligent Decision Support System for Leadership Analysis

First, the existing different intelligent systems were analyzed. It included i information [3, 4], intelligent [5, 6], knowledge [7, 8], expert [9, 10] and decision support [15, 16, 18–21] systems, plus developed multiple criteria methods (CODEC, COPRAS, DUMA and DAM [18, 19]. The purpose was to determine the most efficient Advisory, Negotiation and Intelligent Decision Support System for Leadership Analysis (ANDES) to analyze a leader's life cycle. This developed support system consists of a database, database management system, model-base, model-base management system and user interface (Figure 1).

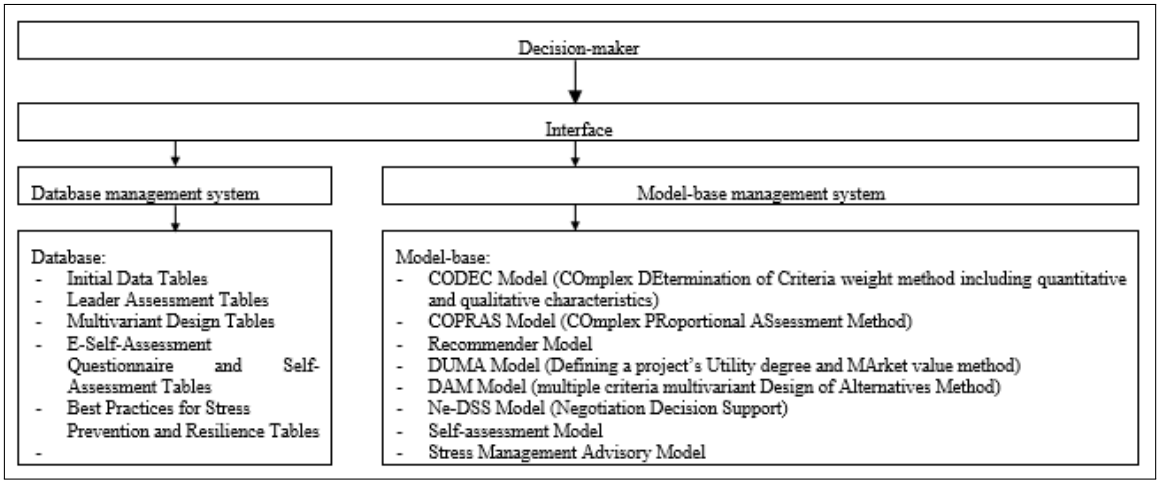


Figure 1: ANDES components

ANDES is an information system that accumulates data and information from various sources and processes them by extensive use of artificial intelligence techniques. It utilizes various multiple-criteria and artificial intelligence models and provides a decision-maker with data, information and knowledge needed for analyzing, compiling and assessing possible alternative resolutions. It can make a decision, derive the received results and safeguard them. Thus the ANDES can be based on data from various sources, allow users to transform a huge amount of unprocessed data, information and knowledge into an analysis of a problem under consideration. Development of the initial version of the *Advisory, Negotiation and Intelligent Decision Support System for Leadership Analysis* was in 2004. The testing of the *System* has been ongoing since then. A total of 176 distance-learning students tested this system. The continuous testing results resulted in improvements to the *System*. The testing of the *System* was the subject of some 18 final master theses.

2.1 Database

Presently the structure of relational databases is the most appropriate in light of the requirements raised by ANDES. A relational database stores information in tables. Every table is given a name for storing it in the external memory of the computer as a separate file. The indexes common for this table are logically interconnected. Thereby the entirety of the logically interrelated table comprises the model.

The data play very meaningful roles. Decisions are made using these as a basis. The more comprehensive the accumulated data are about an object under consideration, the more effective

the decision made can be. For example, various economic, social, legal, technical, technological and other factors from the external environment impact knowledge management. The possible operations of an organization objectively change for better or for worse, as external conditions change. Usually an organization can organize its operations for more than one market. Therefore it is very important to understand and evaluate the constantly changing external micro, meso and macro environment and its impact on an organization's operations in different markets. The external and internal environments of an organization's operations can be described for each time period by basing it on specific data, information and knowledge. Organizations must react to the fluctuating external environment by making adequate strategic, tactical and operative decisions on the basis of such specific information. Since decision-making is an informational process, all of its stages, from the time of setting objectives to the ending of their implementation and evaluating their consequences, must be substantiated by searching for, visualizing, processing and analyzing necessary data, information and knowledge.

A leader interacts with a number of interested parties, all of whom are pursuing various goals and all of whom have different potentials, educational levels and backgrounds of experience. Therefore all the aforementioned parties in this field approach decision-making in various ways. It is often necessary to define these players in terms of their economic, legal/regulatory, technical, technological, organizational, managerial, quality of life, social, cultural, political, ethical and psychological aspects along with other types of information (see Figure 1). This is done to analyze the available alternatives thoroughly and to obtain an efficient compromise solution. Such information needs to appear in as much of a user-oriented manner as possible.

The presentation of information in ANDES, which is needed for decision-making, may be in conceptual (digital, numerical), textual, graphic (diagrams, graphs, drawing, etc), photographic, audio (sound), visual (video) and quantitative forms. The presentation of quantitative information involves criteria systems and models, units of measurement, values and initial weights, which fully define the variants provided. Conceptual information conceptually describes alternative solutions, criteria and ways used to determine the values and weights of the criteria and the like. Upon demand, the ANDES provides conceptual information (images, audio, video, and so on), which aids the user to get a better understanding of the alternatives in question and their defining criteria.

This way ANDES provides a decision-maker with different conceptual and quantitative information about a leader from a database and a model-base. It allows the decision-maker to analyze the above factors and determine an efficient solution.

An analysis of database structures in decision support systems by the type of problem they resolve reveals their various utilities. There are three basic types of database structures: hierarchical, network and relational. ANDES has a relational database structure, when the information is stored in the form of tables. These tables contain quantitative and conceptual information. Each table has a given name by which the computer's external memory saves it, as a separate file. Logically linked parts of the table constitute a relational model. The ANDES database consists of the following tables:

- Initial Data Tables. The data covers general facts about the leader under consideration. The leader's requirements and their significances as well as an intended salary are included.

- Leader Assessment Tables. Quantitative and conceptual information about alternative leaders on the ANDES website [22] show the input data for the multiple criteria analysis in ANDES.

- Multivariant Design Tables. These contain quantitative and conceptual information on the interconnecting elements in the life cycle of a leader in the organization, their compatibilities

and possible combinations, as well as data for the complex multivariant design of the elements previously described herein.

- E-Self-Assessment Questionnaire and Self-Assessment Tables.

An analysis of the available alternatives is necessary for designing and realizing an effective leader's life cycle. A computer-aided multivariant design requires the availability of tables containing data on the interconnecting elements of the life cycle of a leader in the organization, along with their compatibilities, possible combinations and a multivariant design.

The development of possible variants is possible using the aforementioned tables as the basis for a multivariant design of a leader's life cycle. The development of millions of alternatives of a leader's life cycle (including the project on the life cycle of the leader in the organization) is possible by using a multivariant design method. The capacity of these versions to meet various requirements is checked. Alternative versions that are unsatisfactory in terms of the requirements raised are excluded from further consideration. A problem involving the significance of criteria compatibilities arises in the process of designing a number of variants of a leader's life cycle. Thereby the performance of a complex evaluation of the alternatives determines that the value of a criterion weight is dependent on the overall criteria under assessment, as well as on their values and initial weights.

Numerous studies have been conducted worldwide analyzing the reliability of self-assessments. This is quite a controversial issue. A great many researchers attained reliable results proving that self-assessments are sufficiently reliable. Our investigations also demonstrate that self-assessments are sufficiently reliable. The basis for the E-Self-Assessment Questionnaire and Self-Assessment Tables is the presumption that it is possible to determine a leader's level of stress rather accurately by assigning questions for leaders according to some certain methodology and then processing them in accordance with a certain algorithm.

2.2 Model-base

ANDES models are subdivided as quantitative and qualitative by their presentations. Qualitative models (multicriteria, based on expertise) are based on subjective opinions, experiences and assessments of experts. However, when different experts assess the same characteristics of the object, the derived results are often different. This occurs due to the different experiences, educational levels, purposes, available opportunities and the like of different experts. The derived data can be made more objective by applying the expert methods. Quantitative models (i. e. text analytics) reflect the objective features of the objects under consideration, independently of the subjective assessments by experts. Such features of an object can be expressed directly by physical units of measurement (monetary units, degrees, percents, ratios and such). Qualitative models have as many positive and negative features and quantitative models have. Objects being considered by quantitative models are objectively but often not comprehensively reflected. Contrariwise, ANDES qualitative models reflect reality subjectively and comprehensively. Therefore the rationality of applying quantitative and qualitative methods often depends on specific, decision-making situations. Frequently decision-making requires a comprehensive application of quantitative and qualitative models. For example, it is best to apply qualitative research methods when analyzing the qualitative leadership characteristics (emotions, culture, religious, traditions, ethical leader behaviors, psychological capital). However, when analyzing how much money will be spent over the entire process of an office's life cycle, such as the costs of its purchase or construction, exploitation, maintenance upkeep, insurance expenses, taxes and the like, the application of quantitative methods is better.

A determination regarding the efficiency of alternative leaders often takes into account economic, legal/regulatory, technical, technological, organizational, managerial, quality of life, social, cultural, political, ethical, psychological and other factors. Therefore the model-base of the ANDES should include models enabling a decision-maker to analyze the available variants comprehensively and arrive at a suitable choice. The intention for the following models of the model-base is to perform this function:

- CODEC Model (Complex Determination of Criteria Weight Method including quantitative and qualitative characteristics)
- COPRAS Model (Complex Proportional Assessment Method)
- Recommender Model
- DUMA Model (Defining a Project's Utility Degree and Market Value Method)
- DAM Model (Multiple Criteria, Multivariant Design of Alternatives Method)
- Ne-DSS Model (Negotiation Decision Support)
- Self-Assessment Model
- Stress Management Advisory Model
- Text Analytics

Development of the CODEC Model (Complex Determination of Criteria Weight Method including quantitative and qualitative characteristics) allows for the calculation and coordination of the significances of the described quantitative and qualitative characteristics of the criteria.

Development of the multiple criteria COPRAS Model (Complex Proportional Assessment Method) enables a user to obtain a reduced criterion for determining the complex (overall) efficiency of alternatives. This generalized criterion is directly proportional to the relative effect of the values and weights of the criteria under consideration regarding the efficiency of the alternatives.

The DUMA Model determines the utility degree and market salary of leaders. It establishes the competitive salary for a leader on the market. The basis for this model is a complex analysis of all the benefits and drawbacks of a leader. A leader's utility degree and the estimated market salary of a leader are directly proportional to the system of the criteria, which adequately describe them, and the values and weights of those criteria, according to this Model.

Development of the DAM Model of a multiple criteria, multivariant design of a leader's life cycle enables a user to design, with the aid of a computer, up to 100,000 alternative versions. Quantitative and conceptual information are the bases for any life cycle variant obtained in this way.

The bases for the ANDES are the models described above. ANDES can generate millions of alternative versions of leader life cycles (including a project for determining the life cycle of the current leader in the organization). ANDES performs a multiple criteria analysis of a leader's life cycle, determines a leader's utility degree and selects the most beneficial variant for a leader.

A model base, management system can provide various models according to a user's needs. The results of its obtained calculations become the initial data for some other models when using some certain model. Such a model could be for determining the initial weights of the criteria, for designing a leader's life cycle in a multivariant method (including a project on the life cycle of the current leader in the organization), for analyzing by multiple criteria and for setting priorities. Meanwhile the results of the latter, in turn, are acceptable as the initial data for some other

models (such as for determining the leader utility degree, for providing recommendations and for other undertakings).

The *Ne-DSS Negotiation Decision Support Model* is for use by leaders engaged in different negotiation circumstances. One such example could be the purchase of office premises. A leader performing a multi-criteria analysis of all real estate alternatives selects the objects for starting the negotiations. The leader marks (ticks a box with a mouse) the desirable negotiation objects. The Letter Writing Model generates a negotiations e-mail after the selection of the desired objects. It then sends the e-mail to all real estate sellers once the user clicks "Send". The buyer and the seller may perform real calculations (the utility degree, market salary and purchase priorities) of the real estate with the help of ANDES during negotiations. The bases for these calculations are the characteristics describing the real estate's alternatives, obtained during negotiations (explicit and tacit criteria system and criteria values and weights). Development of the final comparative table is in accordance with the received results. Use of ANDES permits the performance of the multiple criteria analysis and selection of the best real estate for purchase version, following the development of the final comparative table.

There are two main categories of rules and procedures in the Ne-DSS Model:

- It compiles suggestions for leaders to employ along with the reasons for the recommended further negotiations with a particular leader.

- It composes a comprehensive, negotiation e-mail for each of the selected broker leaders. The Ne-DSS uses information inherited from the previous ANDES calculations and predefined rules and procedures and composes a negotiation e-mail for each of the selected broker leaders. The e-mail includes a reasonable suggestion for a decrease in the price of the real estate. The e-mail also references the calculations performed by ANDES.

Development of the Self-assessment Model took place while analyzing similar studies that were conducted worldwide. Areas of special attention were the criteria systems used, integration of such criteria into one general assessment, use of aggregation methods, reliability level of the results and tendencies of the results. The authors of this work based their work on the aforementioned and other studies.

A brief analysis of the above COPRAS Model (Complex Proportional Assessment Method) and DUMA Model (Defining a Project's Utility Degree and Market Value Method) follows, as an example.

COPRAS Model (Complex Proportional Assessment Method) The determination of the utility degree and value of the alternative under investigation and establishment of the priority order for its implementation does not present much difficulty if the criteria numerical values and weights have been obtained and the multiple criteria decision making methods are used.

The results of the comparative analysis of the alternatives are presented as a grouped decision making matrix where columns contain n alternatives being considered, while all quantitative and conceptual information pertaining to them is found in lines.

This method assumes direct and proportional dependence of significance and priority of investigated versions on a system of criteria adequately describing the alternatives and on values and weights of the criteria. The system of criteria is determined and the values and initial weights of criteria are calculated by experts. All this information can be corrected by interested parties taking into consideration their pursued goals and existing capabilities. Hence, the assessment results of alternatives fully reflect the initial data jointly submitted by experts and interested parties.

The determination of significance and priority of alternatives is carried out in four stages.

Stage 1. The weighted normalized decision making matrix D is formed. The purpose of this stage is to receive dimensionless weighted values from the comparative indexes. When the dimensionless values of the indexes are known, all criteria, originally having different dimensions, can be compared. The following formula is used for this purpose:

$$d_{ij} = \frac{x_{ij} \cdot q_i}{\sum_{j=1}^{n} x_{ij}}, i = \overline{1,m}; j = \overline{1,n}. \tag{1}$$

where x_{ij} - the value of the i-th criterion in the j-th alternative of a solution; m - the number of criteria; n - the number of the alternatives compared; q_i - weight of i-th criterion.

The sum of dimensionless weighted index values d_{ij} of each criterion x_i is always equal to the weight q_i of this criterion:

$$q_i = \sum_{j=1}^{n} d_{ij}, i = \overline{1,m}; j = \overline{1,n}. \tag{2}$$

In other words, the value of weight q_i of the investigated criterion is proportionally distributed among all alternative versions a_j according to their values x_{ij}.

Stage 2. The sums of weighted normalized indexes describing the j-th version are calculated. The versions are described by minimizing indexes S_{-j} and maximizing indexes S_{+j}. The lower value of minimizing indexes (salary, costs of lifelong learning, etc.) is better. The greater value of maximizing indexes is better (innovation, labour productivity, etc.). The sums are calculated according to the formula:

$$S_{+j} = \sum_{i=1}^{m} d_{+ij}; S_{-j} = \sum_{i=1}^{m} d_{-ij}, i = \overline{1,m}; j = \overline{1,n}. \tag{3}$$

In this case, the values S_{+j} (the greater is this value (alternative 'pluses'), the more satisfied are the interested parties) and S_{-j} (the lower is this value (alternative 'minuses'), the better is goal attainment by the interested parties) express the degree of goals attained by the interested parties in each alternative. In any case the sums of 'pluses' S_{+j} and 'minuses' S_{-j} of all alternatives is always respectively equal to all sums of weights of maximizing and minimizing criteria:

$$\begin{gathered} S_{+} = \sum_{j=1}^{n} S_{+j} = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{+ij}, \\ S_{-} = \sum_{j=1}^{n} S_{-j} = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{-ij}, i = \overline{1,m}; j = \overline{1,n}. \end{gathered} \tag{4}$$

In this way, the calculations made may be additionally checked.

Stage 3. The significance (efficiency) of comparative versions is determined on the basis of describing positive alternatives ('pluses') and negative alternatives ('minuses') characteristics. Relative significance Q_j of each alternative a_j is found according to the formula:

$$Q_j = S_{+j} + \frac{S_{-min} \cdot \sum_{j=1}^{n} S_{-j}}{S_{-j} \cdot \sum_{j=i}^{n} \frac{S_{-min}}{S_{-j}}}, j = \overline{1,n} \tag{5}$$

Stage 4. Priority determination of alternatives. The greater is the Q_j the higher is the efficiency (priority) of the alternative.

The analysis of the method presented makes it possible to state that it may be easily applied to evaluating the alternatives and selecting most efficient of them, being fully aware of a physical meaning of the decision-making process. Moreover, it allowed to formulate a reduced criterion Q_j which is directly proportional to the relative effect of the compared criteria values x_ij and weights q_i on the end result.

DUMA Model (Defining a Project's Utility Degree and Market Value Method) The leader's utility degree and market salary are determined in seven stages. **Stage 1.** The formula used for the calculation of leader's a_j utility degree N_j is given below:

$$N_j = (Q_j : Q_{max})100\%, \tag{6}$$

here Q_j and Q_{max} are the significances of the leader obtained from the equation 5.

The degree of utility N_j of leader a_j indicates the level of satisfying the needs of the stakeholders interested in the leader. The more goals are achieved and the more important they are, the higher is the degree of the leader utility.

A degree of leader utility reflects the extent to which the goals pursued by the interested parties are attained. Therefore, it may be used as a basis for determining leader market salary. The more objectives are attained and the more significant they are the higher will be leader degree of utility and its market salary.

Stage 2. The efficiency degree E_{ji} of money (salary, etc.) invested into leader a_j is calculated. It shows by how many percent it is better (worse) to invest money into leader a_j compared with leader a_i. E_{ji} is obtained by comparing the degrees of utility of the leaders considered:

$$E_{ij} = N_j - N_i. \tag{7}$$

The received results are presented as a matrix clearly showing utility differences of the leaders.

Stage 3. The average deviation k_j of the utility degree N_j of the leader a_j from the same index of other leaders $(n-1)$ is being calculated.

$$k_j = \sum_{i=1}^{n} E_{ji} : (n-1). \tag{8}$$

Stage 4. The development of a grouped decision making matrix for leader multiple criteria analysis. The market salary of a leader being valuated is calculated according to a block-diagram presented in Figure 2.

At the beginning, a grouped decision making matrix for leader multiple criteria analysis is developed, the first criterion of which is based on the actual salary of the leader compared and the value of a leader being valuated. The initial value of a leader being valuated is obtained from the following equation:

$$x_{11} = \sum_{j=2}^{n} x_{1j} : (n-1). \tag{9}$$

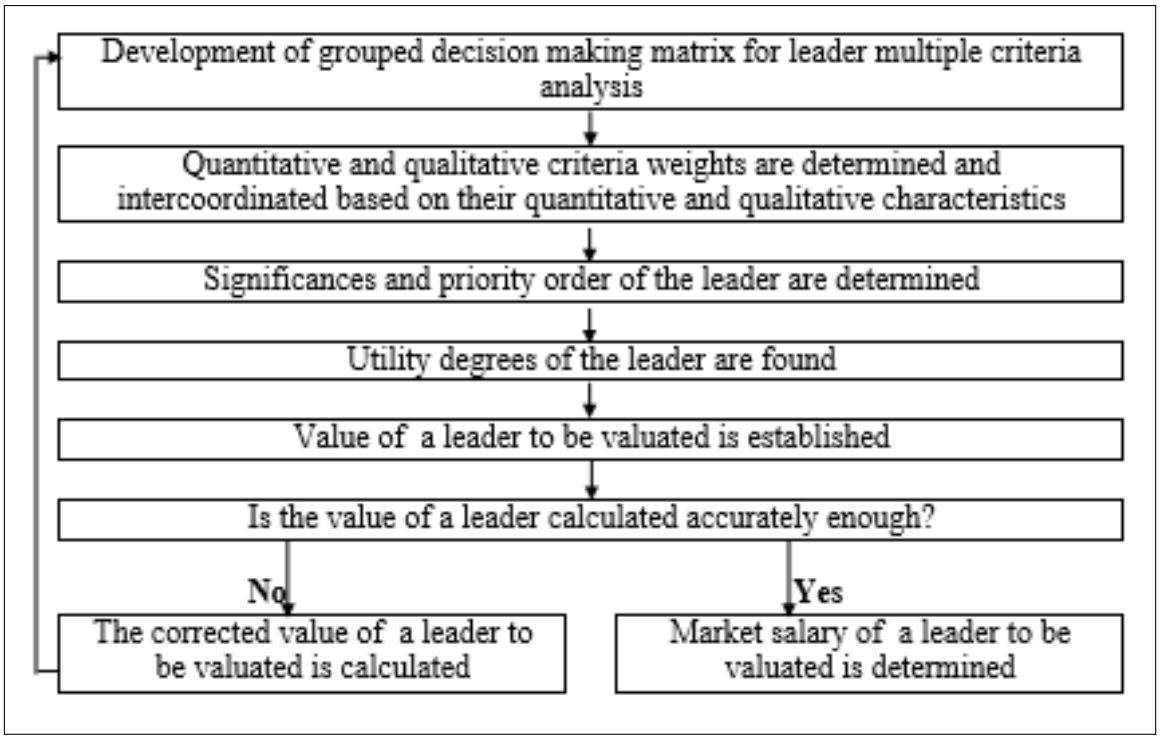


Figure 2: Block-diagram of leader market salary estimation

In this matrix, a leader a_1 to be valuated should be assigned the market salary (x_{11-R}). Other leaders $(a_2 - a_n)$ salaries $(x_{12} - x_{1n})$ known. All the values and weights of the criteria relating to other leaders are also known.

The problem may be stated as follows: what market salary x_{11-R} of a valuated leadera_1 will make it equally competitive on the market with comparison standard leaders $(a_2 - a_n)$? This may be determined if a complex analysis of the benefits and drawbacks of the leader is made.

Using a grouped decision making matrix and the equations 1-9 the calculations are made

Stage 5. The corrected value x_{11-p} of a leader to be valuated a_1 is calculated:

$$x_{11-p} = x_{11} * (1 + k_1 : 100). \tag{10}$$

Stage6. It is determined whether the corrected value $x_1 1 - R$ of a leader being valuated a_1 had been calculated accurately enough:

$$|k_1| < s, \tag{11}$$

where s is the accuracy, %, to be achieved in calculating the market salary x_{11-p} of a leader a_1. For example, given $s = 0,5\%$, the number of approximations in calculation will be lower than at $s = 0,1\%$.

Stage 7. The market salary x_{11-R} of a leader a_1 to be valuated is determined. If inequality 11 is satisfied the market salary of a leader a_1 may be found as follows:

$$x_{11-R} = x_{11-p} \tag{12}$$

If inequality 11 is not satisfied this means that the value of a leader being valuated had not been calculated accurately enough and the approximation cycle should be repeated. In this case, the corrected value$x_{11} = x_{11-p}$ of a leader being valuated is substituted into a grouped decision making matrix of leader multiple criteria analysis and the calculations according to the formulae 1-12 should be repeated until the inequation 2.20 is satisfied.

Solving the problem of determining the market salary x_{11-R} of a leader a_1 being valuated, which would make it equally competitive on the market compared with the leaders $(a_2 - a_n)$, a particular method of defining the utility degree and market salary of a leader was suggested. This was based on a complex analysis of all the benefits and drawbacks of the leader considered.

According to this method the leader utility degree and the market salary of a leader being estimated are directly proportional to the system of the criteria adequately describing them and the values and weights of these criteria.

The Case Study [22] presented illustrates the efficiency of the ANDES in practice.

3 Conclusions

An Advisory, Negotiation and Intelligent Decision Support System for Leadership Analysis (ANDES) was offered as an example for demonstrating the integration of advisory, negotiation and decision support systems. In the future, there are plans to extend the use of the ANDES in Martynas Mazvydas National Library of Lithuania. The plans for the next stage in the development of the ANDES involve integrating this System with biometrics (Computer Mouse Advisory System), Recommended Biometric Stress Management System and Pupil Analysis System [18] systems, which the authors herein have also developed. Such an integration of intelligent and biometrics systems would provide even better assessments of the emotional states of leaders and the submissions of personalized recommendations to them.

Bibliography

[1] M. Uhl-Bien, R. Marion (2009); Complexity leadership in bureaucratic forms of organizing: A meso model, *The Leadership Quarterly*, 20(4): 631–650.

[2] C.M. Youssef, F. Luthans (2012); Positive global leadership, *Journal of World Business*, 47(4): 539–547.

[3] J. Cho, I. Park, J.W. Michel (2011); How does leadership affect information systems success? The role of transformational leadership, *Information & Management*, 48(7): 270–277.

[4] N.K. Dimitrios, D.P. Sakas, D.S. Vlachos (2013); The Role of Information Systems in Creating Strategic Leadership Model, *Procedia – Social and Behavioral Sciences*, 73: 285–293.

[5] M. Rao, R. Dong, V. Mahalec (1994); Intelligent system for safe process startup, *Engineering Applications of Artificial Intelligence*, 7(4): 349–360.

[6] M. Seah, M.H. Hsieh, P.-D.Weng (2010); A case analysis of Savecom: The role of indigenous leadership in implementing a business intelligence system, *International Journal of Information Management*, 30(4): 368–373.

[7] Y.W. Chun, K. Tak (2009); Songgye, a traditional knowledge system for sustainable forest management in Choson Dynasty of Korea, *Forest Ecology and Management*, 257(10): 2022–2026.

[8] B. Mckenna, D. Rooney, K.B. Boal (2009); Wisdom principles as a meta-theoretical basis for evaluating leadership, *The Leadership Quarterly*, 20(2): 177–190.

[9] F. Lehner (1992); Expert systems for organizational and managerial tasks, *Information & Management*, 23(1): 31–41.

[10] M. A. de Oliveira, O. Possamai, D. Valentina, L. V. O., C.A. Flesch (2012); Applying Bayesian networks to performance forecast of innovation projects: A case study of transformational leadership influence in organizations oriented by projects, *Expert Systems with Applications*, 39(5): 5061–5070.

[11] F. G. Filip (2008); Decision support and control for large-scale complex systems, *Annual Reviews in Control*, 32(1): 61–70.

[12] F.G. Filip, A. Dan Donciulescu, G. Neagu (1998); Decision support for blend monitoring in process industries, *Computers in Industry*, 36(1–2): 13–19.

[13] F. G. Filip, A.M. Suduc & M. Bizoi (2014); DSS in numbers, *Technological and Economic Development of Economy*, 20(1): 154–164.

[14] F.G. Filip, K. Leiviska (2009); Large-Scale Complex Systems, *Springer Handbook of Automation*, 619–638.

[15] L.-H. Lim, K.S. Raman, K.-K. Wei (1994); Interacting effects of GDSS and leadership, *Decision Support Systems*, 12(3): 199–211.

[16] J. Rees, G.J. Koehler (2000); Leadership and group search in group decision support systems, *Decision Support Systems*, 30(1): 73–82.

[17] A. Dan Donciulescu, F. G. Filip (1985); DISPECER-H – A decision supporting system in water resources dispatching, *Annual Review in Automatic Programming*, 12(2): 263–266.

[18] A. Kaklauskas (2015); *Biometric and Intelligent Decision Making Support*, Series: Intelligent Systems Reference Library, Vol. 81, XII. Springer-Verlag, Berlin.

[19] A. Kaklauskas (1999); *Multiple Criteria Decision Support of Building Life Cycle*, Research Report presented for Habilitation. Technika, Vilnius.

[20] L. Kanapeckiene, A. Kaklauskas, E.K. Zavadskas, M. Seniut (2010); Integrated knowledge management model and system for construction projects. *Engineering applications of artificial intelligence*, 23(7): 1200–1215.

[21] N. Lepkova, A. Kaklauskas, E.K. Zavadskas (2008); Modelling of facilities management alternatives, *International journal of environment and pollution*, 35(2–4): 185–204.

[22] http://iti.vgtu.lt/imitacijosmain/simpletable.aspx?sistemid=518

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):678-685, October, 2015.

PCA Encrypted Short Acoustic Data Inculcated in Digital Color Images

S.H. Karamchandani, K.J. Gandhi, S.R. Gosalia,
V.K. Madan, S.N. Merchant, U.B. Desai

Sunil H. Karamchandani*
1. Indian Institute of Technology, Bombay
Mumbai, India.
*Corresponding author: skaramchandani@rediffmail.com

Krutarth J. Gandhi, Siddharth R. Gosalia
D. J. Sanghvi College of Engineering
Mumbai, India.
gandhi.krutarth@yahoo.co.in
siddharth_destiny@hotmail.com

Vinod K. Madan
Kalasalingam University
Krishnankoil (TN), India
klvkmadan@gmail.com

Shabbir N. Merchant
Indian Institute of Technology, Bombay
Mumbai, India
merchant@iitb.ac.in

Uday B. Desai
Indian Institute of Technology, Hyderabad
Hyderabad, India
ubdesai@iith.ac.in

Abstract: We propose develop a generalized algorithm for hiding audio signal using image steganography. The authors suggest transmitting short audio messages camouflaged in digital images using Principal Component Analysis (PCA) as an encryption technique. The quantum of principal components required to represent the audio signal by removing the redundancies is a measure of the magnitude of the Eigen values. The aforementioned technique follows a dual task of encryption and in turn also compresses the audio data, sufficient enough to be buried in the image. A 57Kb audio signal is decipher from the Stego image with a high PSNR of 47.49 and a correspondingly low mse of 3.3266×10^{-6} with an equalized high quality audio output. The consistent and comparable experimental results on application of the proposed method across a series of images demonstrate that PCA based encryption can be adapted as an universal rule for a specific payload and the desired compression ratio.
Keywords: Colour image steganography, principal component analysis, eigen thresholding, Pareto analysis.

1 Introduction

Steganography or Stego in IT parlance, in Greek means "covered writing" and is defined by Markus Kahn [1,2] as the art and science of communicating in a way which hides the existence of the communication. In contrast to Cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a

Copyright © 2006-2015 by CCC Publications

cryptosystem, the goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present. A modern steganographic system should defeat detection even by a machine. It replaces bits of useless or unused data in computer files such as graphics, sound, text, HTML with bits of different invisible information. This hidden information can be plain text, cipher text, sound, and even images. Ideally steganography can be used for any communication channel. However in real life the cover media generally used are multimedia objects such as image, video, and audio files. The reasons include that cover media should be large compared to the size of the secret message, and the methods so far developed have less than one percent of the cover size, and the indeterminacy in the cover is necessary to achieve the necessary security. Large objects without inderminacy like the value π at a very high precision are not suitable. The transmitting data should be plausible as in the modern digital society dependence on audio and image files are so prevalent. It is desirable that a steganography system should have a good embedding capacity, be secure and robust.

Modern steganography uses a number of techniques such as masking and filtering, algorithms and transformations, and least significant bit insertion [3]. In masking and filtering, the information is hidden inside of an image using digital watermarks that include information such as copyright, ownership, or licenses. It adds an attribute to the cover the image thus extends the amount of information presented. In algorithms and transformations technique data is hidden in mathematical functions that are often used in compression algorithms and facilitates to hide the secret message in the data bits in the least significant coefficients. The least significant bit insertion is the most common and popular method of the modern day steganography, and it uses LSB of a picture's pixel information keeping the overall image distortion to a minimum while the message is spaced out over the pixels in the images. This technique works best when the image file is larger than the message file. We propose to inculcate audio information in color images following the encryption of the short audio messages using the Principal Component Analysis (PCA). The age old adage of PCA incepts as an encrypting tool diversifying from its myth as a compression standard.

The framework of the paper is as follows. Literature review of steganography related to audio embedding is detailed in Section 2. The subsequent Section 3 discusses the proposed algorithm elaborating the PCA encryption of audio data followed by the implementation of steganography and steganalysis modules in Section 4. Simulation results are tabled in Section 5 with the conclusions drawn in Section 6.

2 Related Steganography Techniques for Audio Concealment

Majority of the steganography techniques disguise either image or text within an audio or image file. The major contributions concerned with encryption of acoustic files, particularly short audio message in images are described herewith. The authors of [4] discuss a traditional steganography technique which involves substitution of the least significant bit of each pixel of the cover image with the encrypted bits of the audio file. An arbitrary color bit stream of the RGB image is considered as the envelope where the kilo byte audio information is saved. The ciphering of the audio signals is loosely based on the very basic fundamentals of logic design. The encryption is performed [4] using the Boolean operations on the bit 5 and bit 6 of color information of the individual pixels given by (1)

$$h = (b_5 \oplus b_6) \oplus m \tag{1}$$

where "h" represents the encoded bit generated from the X−OR operation of the audio bit "m" with the bits of color information obtained from the image. The encoded bit then replaces the

LSB of the respective color pixels in the cover image. Since the exclusive disjunction of the three logical variables is associative and reversible the audio bit is recovered from (2)

$$m = (b_5 \oplus b_6) \oplus h \tag{2}$$

The process of steganalysis involves decoding of the secret "key" deposited in the header files. The knowledge of the header file exposes the entire stego process making it prune to brute force attacks. The algorithm also puts a cap on the length of the audio file to be embedded in the image. Moreover, the technique proposed in this paper does not give any information regarding the evaluation of the performance parameters. Exploiting the color information The authors [5–8] propose a steganography technique in which Discrete Cosine Transform (DCT) is performed on the RGB planes of the cover image by converting them to grayscale and creating blocks of size 8×8. The least significant bits of the frequency coefficients of B plane are swapped with the bits of the audio file [9]. Manipulation of the color component in the individual blocks leads to a computationally extensive algorithm. In the process the low frequency coefficients are also being manipulated which when converted back to spatial domain, the changes can be easily perceived by the human eye.

The PCA technique incorporated in our proposed algorithm is a method of compression which is camouflaged as an encryption technique for the audio signal. The proposed technique performed in the spatial domain and suggests a robust encryption algorithm, thus overcoming the drawbacks of the existing methods.

3 Proposed Steganography and Steganalysis Model for Short Audio Data

We propose to camouflage short audio signals encrypted by their immanent principal components within the digital color images.The wave or "wav", the most commonly used audio file format accommodates the audio information exclusively in the later 43 bytes. The header files, followed by four bytes representing the length of the file precede the audio data. The proposed model is sub divided in two modules: steganography and steganalysis for encryption and deciphering the data as illustrated in Figure 1

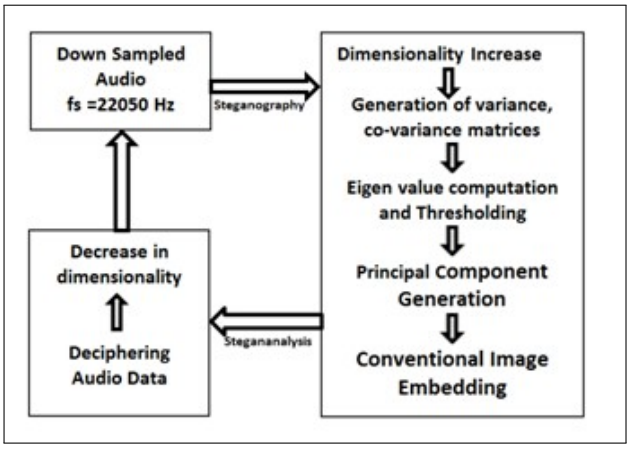


Figure 1: Proposed Model for smuggling audio information in image

3.1 Steganography: Manipulation of the embedded audio

The encryption is based on the Gaussian assumption that an N- dimension data evolves N directions of variance. The one dimension audio signal is trapped in a 2-D array and is represented as (3)

$$Z = WX \tag{3}$$

Where "W" indicates the weight vector required to minimize correlation between the two dimensions. This sets up the basis for the PCA technique which not only provides a good representation of the signal but also reduces the redundancy in the data. The feature similarity is obtained by decorrelating the covariance matrix as (4)

$$z_1^t z_2 = 0 \tag{4}$$

Diagonalization of the covariance matrix suppresses cross-dimensional co-activity. The decorrelation (whitening) of the weights is obtained as a diagonalization problem given in (5)

$$WCOV(X)W^t = I \tag{5}$$

where COV (x) denotes the covariance of the input data and I represents the identity matrix. The solution of the resulting covariance matrix is a function of the eigenvectors and eigenvalues of covariance of X. The decorrelating matrix W contains vectors that normalize the input's variance also called as the principal components while the audio signal gets scaled to a well developed Gaussian curve with unit variance in all dimensions. The decorrelating matrix is used to find the directions of maximal variance in the audio data. The variance of each principal component is reflected in the Eigen values obtained from the as a solution of diagonalization algorithm.The audiosigna lX_e,is obtained in (6) as a scrambled waveform with its redundancies eliminated.

$$X_e = XW \tag{6}$$

The scrambled lX_e is then conventionally embedded in the image using the missionary LSB algorithm.

3.2 Steganalysis: validation of the recovered audio

Steganalysis is art of reconnoitering the hidden information in a medium. The retrieval of speech signal from its redundant components is an essential criterion to discuss the robustness of the system. The generated principal components are used for deciphering the audio data using (7)

$$X = X_e W \tag{7}$$

The audio samples are then retrieved by reducing the dimensionality of the resultant $2-$D array X using the Pareto analysis.

4 Implementation of Proposed Scheme

The dominant frequency band where human speech cognition is most sensitive is around 3 kHz. The epiglottis transmits the vowel sounds at about 100 Hz. The audio message of size 57 Kb is disguised as a "wav" file in a 24-bit ".bmp" format color image of size $256 \times 256 \times 3$. Each sample is recorded in a 16 bit stereo.The audio signal sampled at 44100 samples/sec is down sampled to 22050 Hz, causing the removal of samples with frequencies above 10.025 Khz. The preprocessing eliminates whatever random noise existing in the wav file.

4.1 Proposed algorithm

The algorithm for hiding N audio samples by is structured by selecting the prime principal components. The Eigen vector corresponding to the highest Eigen value is the first principal component. By thresholding the Eigen value, the least important Eigen vectors can be eliminated and a matrix of retained Eigen vectors in the order of significance is formed whose transpose is multiplied with the transpose of audio sample matrix to obtain final matrix. The least significant bit of each pixel of the cover image is replaced by the bits of binary value of each element of the final matrix. This technique reduces the number of bits to be hidden as compared to the actual number of bits to be hidden.

5 Simulation Results

Short hidden message is embodied as an audio signal. The PCA performed on the final matrix contains the original data expressed in terms of retained Eigen vectors that are orthogonal to each other, thus compressing the original data.

5.1 Eigen thresholding

The Pareto analysis is used to determine the total number of Eigen values corresponding to the principal Eigen vectors to be considered for audio retrieval. This process we terms as Eigen thresholding. A Pareto chart in Figure 2 (a) plots the individual variances against the percentage contribution of each Eigen value. The first ten principal components contribute to about 95% information of the audio data. Figure 2 (b) shows the range of the Eigen values obtained from the covariance matrix. From the plot, it is indicated that the final 20% of the Eigen values represent the actual content of information in the signal. A hard threshold and the number of principal components are determined by analyzing the plot of the Eigen values. The audio data is therefore encrypted by compressing it into the fewer principal components. By

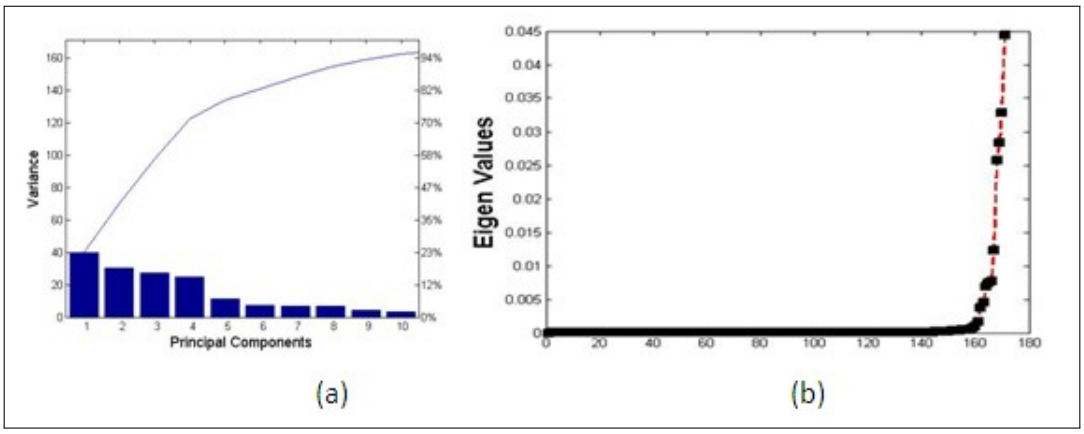


Figure 2: Proposed Model for smuggling audio information in image

thresholding the Eigen value, the principal components with least contribution have been ignored due to redundancy in theinformation. Table 1 illustrates the quantum of principal components and compression ratio obtained for different Eigen thresholds.

The compression ratio between the data to be hidden and original audio signal are calculated as (8)

$$CompressionRatio = \frac{OriginalData - CompressedData}{OriginalData} \tag{8}$$

Table 1: Number of principal components and Compression Ratios achieved for various thresholds of Eigen value for cover image (Barbara)

*E*igen Threshold	*P*rincipal Components Array	*C*ompression Ratio
10^{-2}	171×5	0.9707
10^{-3}	171×11	0.9355
0.8×10^{-3}	171×12	0.9297
10^{-4}	171×31	0.8183
0.75×10^{-4}	171×35	0.7948

The number of principal components exhibits an inverse relation with the Eigen threshold as observed in Table 1, causing the compression ratio to decrease. For threshold value 10−−4, sufficient number of principal components is retained to obtain the retrieved audio message with no distortion. Decrease in the number of principal components cause significant Eigen vectors to be ignored, resulting in misinterpretation of the derived audio. For threshold value greater than 10−−4, the additional principal components do not provide any information of the message and are of lesser significance. Also, the compression ratio will be decreased due to the additional data of lesser significance to be hidden in cover image. Hence the optimum value for thresholding the Eigen values is 10−−4 with ideal compression ratio of 0.8183.

5.2 Performance parameters for steganalysis

Performance parameters such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Normalized Absolute Error (NAE) [10], Maximum Difference (MD) and Structural Content (SC) are used to determine the quality of stego image and are calculated as (9)− (13). Table 2 charts the performance parameters of stego image for various thresholds of Eigen value.

$$MSE = \frac{1}{MN}\sum_{j=1}^{N}\sum_{k=1}^{N}(x_{j,k} - x'_{j,k})^2 \tag{9}$$

$$PSNR = 10\log\frac{255^2}{MSE} \tag{10}$$

$$SC = \frac{\sum_{j=1}^{M}\sum_{k=1}^{N}(x_{j,k})^2}{\sum_{j=1}^{M}\sum_{k=1}^{N}(x'_{j,k})^2} \tag{11}$$

$$MD = \max(\left|(x_{j,k} - x'_{j,k}\right|) \tag{12}$$

$$NAE = \frac{\sum_{j=1}^{M}\sum_{k=1}^{N}\left|(x_{j,k} - x'_{j,k})\right|}{\sum_{j=1}^{M}\sum_{k=1}^{N}\left|(x'_{j,k})\right|} \tag{13}$$

The cover image x of size M × N and the stego image of size x′ of size M × N, and x_jk and are pixel located at j−th row and k−th column of images x and x′ respectively. The maximum difference (MD) is the maximum value of absolute difference between a pixel located at j−th row and k−th column of cover image and Stego image and it is same, equal to 0.0039 for all thresholds of Eigen value. The value of structural content also remains same, equal to unity which implies the cover image and stego image are similar. As the threshold for Eigen values is decreased to retain the significant principal components, the size of final matrix increases thus increasing the number of bits to be replaced in cover image. Hence, the value of MSE and NAE increase causing a decrease in the values of PSNR. The above analysis and performance parameters

Table 2: Performance Evaluation parameters for different thresholds of Eigen value for multiple images

*E*igen *T*hreshold	*I*mage	*M*ean Square *E*rror (MSE)	*P*eak Signal to *N*oise Ratio (PSNR)	*N*ormalized Absolute *E*rror (NAE)
10^{-2}	*Lena*	5.4097×10^{-4}	48.13	2.7513×10^{-4}
	Barbara	5.4645×10^{-4}	48.13	3.3014×10^{-4}
	Cameraman	5.4535×10^{-4}	48.13	2.9832×10^{-4}
10^{-3}	*Lena*	1.1895×10^{-6}	47.86	6.0493×10^{-4}
	Barbara	1.1864×10^{-6}	47.86	7.1679×10^{-4}
	Cameraman	1.1882×10^{-6}	47.86	6.5001×10^{-4}
0.8×10^{-3}	*Lena*	1.2975×10^{-6}	47.83	6.5986×10^{-4}
	Barbara	1.2955×10^{-6}	47.83	7.8267×10^{-4}
	Cameraman	1.27555×10^{-6}	47.83	6.9776×10^{-4}
10^{-4}	*Lena*	3.3462×10^{-6}	47.49	1.7×10^{-3}
	Barbara	3.3404×10^{-6}	47.49	2×10^{-3}
	Cameraman	3.3188×10^{-6}	47.49	1.8×10^{-3}
0.75×10^{-4}	*Lena*	3.7630×10^{-6}	47.45	1.9×10^{-3}
	Barbara	3.7475×10^{-6}	47.45	2.3×10^{-3}
	Cameraman	3.7528×10^{-6}	47.45	2.1×10^{-3}

indicate successful implementation of hiding audio message in an image. The threshold for the Eigen values can be easily visualized by the Pareto chart which provides percentage contribution of the individual Eigen values. Figure 3 shows the stego image adjacent with the recovered audio signal.

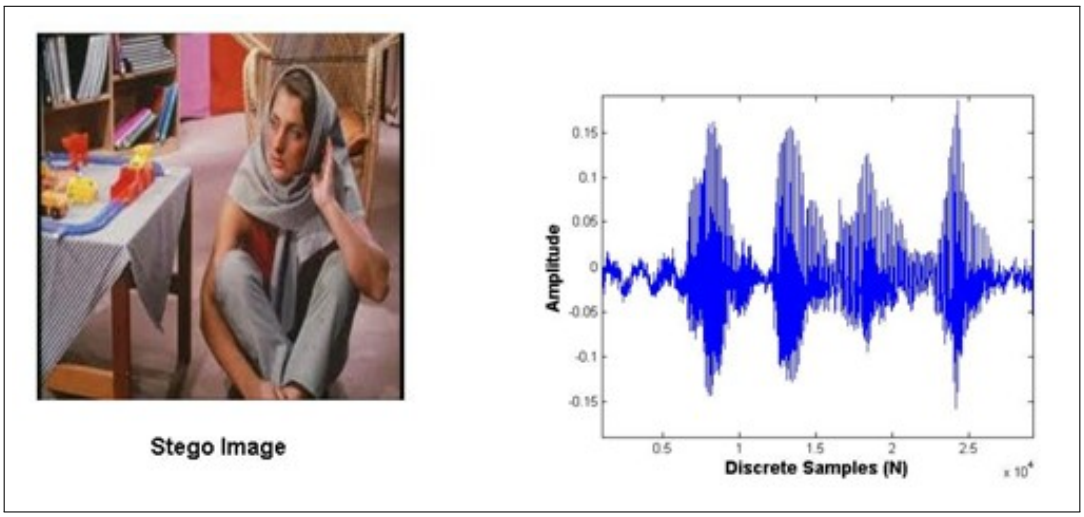


Figure 3: Stego image (Barbara) for Eigen threshold 0.0001 alongside retrieved audio sample

6 Conclusions and Future Works

We have developed a generalized procedure for concealing diminutive information using image steganography. The quantum of the principal components selected is a measure of the magnitude of the Eigen values. A high PSNR of 47.49 with reverse MSE value of 3.3266×10^{-6} is obtained

for the stego image along with a well audible recovered audio signal using the encrypted PCA technique. The results demonstrated in Table 2 illustrates comparable values of the parameters PSNR, NAE and MSE across a set of images concluding that the proposed method can be custom-made universally across all sets of images. For steganography, principal components emerge as a method of encryption while simultaneously performing compression of the audio signal. The future scope would involve adaptive thresholding of the Eigen value for principal component selection with the payload as a constraint.

Bibliography

[1] http://www.jjtc.com/stegdoc/index2.html

[2] Bohme, R. (2010); *Advanced Statistical Steganalysis*, Springer-Verlag, Berlin.

[3] Samarth, K.N.; Poornapragna, M.S. (2013); A Novel Technique of hiding an audio message in an image, *International Conference on Electronics and Communication Engineering*, ISBN: 978-93-83060-04-7, 170-172

[4] Khalil, M. I. (2011); Image Steganography: Hiding Short Audio Messages within Digital Images, *J. Computer Science and Technology*, ISSN: 1860-4749, 11(2): 68-73.

[5] Abd, E. H.; Abdulwahed, H. J.; Mohammed, H. A. (2012); The Use of Discrete Cosine Transformation (DCT), *Information Hiding Process*, J Kerbala University, 10(1):45-51.

[6] Shahana, T. (2013); A Secure DCT Image Steganography based on Public-Key Cryptography, *International Journal of Computer Trends and Technology (IJCTT)*, 4(7): 2039-2043.

[7] Bucerzan, D.; Ratiu, C.; Manolescu, M. J. (2013); SmartSteg: A New Android Based Steganography Application, *International Journal of Computers Communications & Control*, 8(5):681-688.

[8] Singh, K.M.; Chanu, Y.J.; Tuithung, T; (2014), Steganalysis of $\pm k$ Steganography based on Noncausal Linear Predictor, *International Journal of Comuters Communications & Control*, 9(5):623-632.

[9] Othman, S. E. (2012); Hide and Seek: Embedding Audio into RGB 24-bit Color Image Sporadically Using Linked List Concepts, *IOSR Journal of Computer Engineering (IOSRJCE)*, ISSN 2278-0661, 4(1): 37-44.

[10] WAVE and AVI Codec Registries - RFC 2361, Microsoft Corporation (June 1998).

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):686-701, October, 2015.

Adaptive Probe-based Congestion-aware Handover Procedure Using SIP Protocol

R. Libnik, A. Svigelj

Rok Libnik
Telekom Slovenije, d.d.,
Cigaletova 15, 1000 Ljubljana, Slovenia
rok.libnik@telekom.si

Ales Svigelj*
1. Department of Communication Systems,
Jozef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
ales.svigelj@ijs.si
2. Jozef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia
*Corresponding author: ales.svigelj@ijs.si

Abstract: Wireless technologies have evolved very rapidly in recent years. In the future, operators will need to enable users to use communication services independently of access technologies, so they will have to support seamless handovers in heterogeneous networks. In this paper we present a novel adaptive congestion aware Session Initiation Protocol (SIP) based procedure for handover in heterogeneous networks. In the proposed algorithm the handover decision is based in addition to signal strength, also on target network congestion status, which is tested during the conversation. As SIP protocol was used, the proposed procedure is independent of access technologies. For performance evaluation of proposed procedure we developed a purpose built simulation model. The results show that the use of the proposed adaptive procedure significantly improves the QoE of VoIP users, compared to reference scenario, in which only signal strength was used as the trigger for handover decision.
Keywords: Session Initiation Protocol (SIP), seamless handover, heterogeneous networks, performance evaluation, congestion awareness.

1 Introduction

Fixed, nomadic and mobile telecommunications networks, which provide voice and data services, are nowadays converging toward a seamless heterogeneous telecommunication network. Due to many different wireless access technologies, comprising licenced (e.g. GSM/UMTS, HSPA, WiMAX, and LTE) and unlicensed (e.g. WLAN) access, there is a need for a uniform access to converged services. Such services should be independent of the access technologies, providing seamless connectivity and sufficient quality of experience (QoE). In the future networks the wireless access networks will play the key role, as their inherent characteristics of the limited radio bandwidth and channel properties, are of the paramount importance for the provision of appropriate transmission rates and quality of service (QoS).
In addition, WLAN is increasing its popularity, in particular in home/business environment (limited coverage areas). Thus, to enable mobile users to communicate using a variety of different access technologies, terminals have to support several network interfaces. Terminal manufacturers are already producing multi mode terminals and the number of interfaces is bound to increase with technology limits. Operators, on the other hand are starting to offer fixed mobile converged services. With increasing demand from the users to communicate independent of access technologies, operators will need to offer seamless handover between heterogeneous access

Copyright © 2006-2015 by CCC Publications

networks (i.e. vertical handover). In homogeneous networks handover techniques (i.e. horizontal handover) are well studied in the literature and already integrated in mobile networks. The horizontal handover is usually triggered by received signal strength (RSS) only. In the future, this will not be sufficient and other parameters should be taken into account (network congestion status in our case) in handover decision. Thus, new mechanisms need to be developed.
The main contribution of this paper is an advanced adaptive SIP based procedure for congestion aware handover in heterogeneous networks, together with its performance evaluation in simulation environment.
The remainder of the paper is organized as follows. In the following subsections the related research work on mobility management techniques and support for handovers when using the SIP protocol is described. In Section 2 the novel adaptive congestion aware SIP based procedure for handover in heterogeneous networks is presented, while its performance evaluation is presented in Section 3. The paper ends with conclusions in Section 4.

1.1 Mobility management

Mobility management techniques are defined as techniques that support user movement within and between different networks. The handover process can be in general divided into three phases: (i) Handover information gathering phase, (ii) handover decision phase and (iii) handover execution phase [1].
In the first phase (i.e. handover information gathering phase) a mobile node collects not only network information, but also information about the other components of the system such as network properties, mobile devices, access points, and user preferences [1]. The information/parameters typically collected/measured are the following [1] [2] [3] [4] [5]:
- Availability of neighbouring network
- Received Signal Strength (RSS), Signal Noise Ratio (SNR), Carrier to Interference Ratio (CIR), Signal to Interference Ratio (SIR), Bit Error Ratio (BER),
- Delay, jitter
- Throughput,
- Economic price of the usage of the network.
- The Mobile device's state by gathering information about battery status, resources, speed, and service class.
- User preferences information such as budget and services required, preferred network operator.
- Context information.

As seen, some parameters (SNR, delay, jitter, bandwidth, and power consumption) are network/hardware related and cannot be influenced by the user, while others (price, preferred network operator) represent parameters that can be selected/set by the user. Gathered parameters can be grouped also according to the origin of the parameters. They can be provided by the network (i.e. service independent) or can be provided by application/service (i.e. service dependent).
In the second phase (i.e. handover decision phase) the decision for handover is made, based on criteria function, taking into account different information/parameters, which were gathered during the first phase. The second phase is one of the most critical processes during the handover, as in this phase, the decision about time (if and when) and to which network (selecting the best network fulfilling requirements) the handover is made is taken. In a homogeneous network environment the decision about time usually depends on RSS values, while the selection of the network is not an issue since the same networking technology (horizontal handover) is used. In heterogeneous networks the selection of the appropriate network is quite complex, as

many parameters/information obtained from the different information sources (i.e. network, mobile devices, and user preferences) must be evaluated in order to make the best decisions. In this paper we are focusing on vertical handover solutions based on the combination of different parameters. Selection of appropriate parameters and handling of appropriate trigger algorithm in handover decision phase is of a paramount importance, as wrong handover decision can lead into unsatisfied users. This is in particular important when performing handover using real time applications such as Voice over IP (VoIP), where the assurance of appropriate level of Quality of Experience (QoE) in the target access network, represents the main challenge. Operator's backbones usually do not present a bottleneck as they are well maintained and controlled in order to provide an adequate level of QoS.
In the last (third) phase (i.e. handover execution), traffic flow is handed over to the target network. This means that all the traffic is sent using the new connection, while the connection with the old network is terminated. This phase should also guarantee a smooth session transition process.
In this paper we focused in particular on the second phase (i.e. handover decision), proposing new procedure that improves the handover decision and consequently the user experience when performing handover using VoIP applications in heterogeneous networks.
Handover in heterogeneous network can be performed using different protocols and this has been the subject of several studies [1] [5] [6]. At the network layer, Mobile IP (MIP), defined in [8], has been most frequently selected [8] [9] [10] [11] as the protocol for handover. With the modifications presented in [12] it can provide greater support for real time services on a Mobile IPv4 network, by minimizing the period of time when an mobile node (MN) is unable to send or receive IPv4 packets due to delay in the Mobile IPv4 Registration process. In [13] authors proposed an enhancement of Mobile IP (MIP) called MIP with Home Agent Handover (HH-MIP) to enjoy most of the advantages of Route Optimization MIP (ROMIP) but with only a small increase of signalling overhead. The most widely used protocols at the transport layer are TCP and UDP. Both have some limitations for mobility support. However, a new solution called mobile SCTP (mSCTP) has been developed to enable IP addresses to be added, deleted and changed during active SCTP association [14] [15]. For mobility management at the application layer, SIP is usually selected as the most favoured protocol [9] [10] [11] [12] [16] [17] [18]. SIP runs on top of several different transport protocols and is today's most widely used protocol for IP telephony penetrated in both terrestrial and satellite networks [19]. The advantage of using the SIP protocol for handover execution is, that SIP is an application layer protocol, and thus its use does not have a great impact on the network changes needed. The transport independence of SIP means that it does not require great network involvement in handover execution. However, the application usually needs to be improved / customized to support handover and only SIP based application can perform handover. Application level solutions based on tunnelling [20], using SIP only for signalling can overcome this problem. The biggest advantage of using SIP is its wide adoption in real operator environments, since almost all operators that are offering VoIP services are using SIP for signalling. In addition, SIP is used in many operator environments and has been selected as the primary signalling protocol in IMS (IP Multimedia Subsystem) networks. In this paper we focus on solutions that can be deployed easily in a real operator environment. Thus, we decided to focus on the use of SIP for mobility management, which is shortly described in the next subsection.

1.2 SIP mobility

SIP protocol [21] [22] is an application layer signalling protocol for establishing, modifying, and terminating Internet multimedia sessions. These sessions include Internet telephone calls,

multimedia distribution, and multimedia conferences. SIP invitations used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types.
When using SIP protocol for IP telephony in operator's environment, the regulatory and security issues are forcing operators to provide additional functionalities that can affect the architecture of the IP telephony solution [23]. Usually, the Session Border Controller (SBC) is added to their network [24], which can lead to a conflict with SIP architectural principles. SIP based SBCs typically handle both signalling and media, resulting in call flow to be changed in a way that all RTP streams are routed via SBC (see also Figure 4).
With minor modifications, SIP can support four types of mobility. Terminal mobility enables devices to move between subnets and be accessible to other hosts and to continue any ongoing session when they move. Session mobility enables users to maintain a session while moving from one terminal to another. Personal mobility allows users to use the same set of services, even when changing devices or network attachment points. Service mobility enables users to be identified by the same logical address, even if the user is at different terminals.
In this paper we are focusing on terminal mobility, for which two types of mobility management have been defined: pre-call mobility and mid call mobility. SIP protocol supports several applications; however, in this paper we have selected IP telephony as a typical representative of real time application.
In the pre-call mobility scenario MN gets a new IP address prior to the call, thus this operation does not affect the quality of IP telephony service and will therefore not be discussed further.
In the mid call mobility scenario first a call is established between the corresponding node (CN) and the MN that is in the home network. When MN moves to the target network, it gets new IP address and sends SIP re INVITE message to CN and informs it about the location change. The new RTP session is then established. The limitation of this approach is that the SIP server is not informed about the location change. Some solutions have been presented in the literature in which MN informs the SIP server about the location change after sending the SIP re INVITE message [13]. However, in a real operator environment, information about location change needs to be sent to the SIP server prior to starting a new SIP session between MN and CN. This should be done, for example, to support proper charging, since prices can differ between networks.
To overcome the limitation of mid call mobility in [25] we have proposed SIP enhanced mid call scenario (SEMCS)
In SEMCS scenario, a call is established between the CN and the MN that is in the home network After moving to target network, the MN sends the SIP re-INVITE message to the SIP server to inform it about the location change. The SIP server then forwards the SIP re INVITE message to the CN . After the acknowledgement the new RTP session is established using (new) IP address of the target network.
In [25] we proposed message exchange via SIP server and we focused on handover execution based on SNR ratio only. That procedure was upgraded in [26], where we are presenting a novel procedure for handover decision based on congestion detection (CAHP-C). In this paper, we are extending the CAHP-C procedure described in [26] with an adaptive procedure for congestion aware handover.

2 CAHP-A adaptive procedure for congestion aware handover

As stated in the introduction in today's mobile/wireless homogenous networks the trigger for handover is usually based on SNR only. This is sufficient as the whole network uses the same access technology and is under control of one operator offering service. In the future heterogeneous networks, the target access network could be covered by different operator or it is

not even operator's network. The later is usually the case with open WLAN network in a hotel or in a congress centre, where there are several access points (AP) connected to single xDSL connection. On one hand the users can have good signal coverage and very fast (e.g. 52 Mbit/s) connection between MN and AP, while on the other hand, when traffic from all APs is gathered on single xDSL line bandwidth becomes a bottleneck. Thus, the SNR level is not sufficient parameter for handover decision and should not be performed only when SNR level exceeds the threshold, but should be done based also on other parameters (e.g. congestion status) [26]. This is especially important when providing real time applications (IP telephony in our case), where ability to provide sufficient QoE for the user is the most important and where performing handover to congested network could lead to degraded service level.
Our solution is described in [26], where we defined basic congestion aware handover procedure (CAHP-C), which enables efficient handover performance, taking into account also the congestion status of the target network. It should be noted, that receiving signal power measurement remains the prerequisite for handover (i.e. handover cannot take place to a network lacking or with limited signal coverage).
In our study two groups of networks with different characteristics were defined. Networks in the first group are reliable and expensive (e.g. UMTS, HSPA and LTE), while networks in the second one are cheaper or even free of charge and unreliable (e.g. WLAN). In order to present our solution more clearly we selected one representative from each group. The HSPA was selected from the first group of networks and WLAN from the second group. Those two will be used in the rest of this paper. Please note, that our approach can be used between any two networks (e.g. also between different WLAN networks). We assumed that congestion (i.e. QoE degradation) can happen in the WLAN network only. We are not focusing on "who" or "what" is causing the additional delay (e.g. MAC, TCP, routing, low bandwidth on the access link), but only on the fact that if there is a delay, which is not acceptable for IP telephony, we do not use the corresponding access network.
As the proposed CAHP-A procedure is an extension to CAHP-C procedure we provide short description of CAHP-C emphasizing only the main characteristics, as partially depicted in Figure 1, while in depth description of CAHP-C can be found in [26].

2.1 CAHP-C

The CAHP-C procedure is used only if SNR exceeds the predefined threshold [26]. When SNR is below threshold, the proposed procedure is not used. In order to detect possible congestion in the WLAN network before the handover is executed, we proposed Pre-probe algorithm. The congestion is detected with delay testing. In order to monitor congestion (i.e. round trip delay measurement) we defined new SIP message named SIP pre_PROBE , which is sent before the SIP re INVITE message. As characteristics of the WLAN access network can also change during the call, we defined another algorithm named Mid-probe algorithm. Another SIP message called $SIP\ mid\_PROBE$ was defined, which is used to check the congestion status of the WLAN access network when in use. After receiving the responses the MN calculates the average delay D_{pre} (Pre-probe algorithm) or D_{mid} (Mid-probe algorithm) from MN to SBC. We also defined two parameters T_{pre} and T_{mid}. When user is trying to handover to WLAN network, the parameter T_{pre} defines the period, when SIP pre_PROBE messages are sent again if measured delay is above predefined threshold T_d (i.e. 200ms in our case) [26]. When user is already using WLAN network the parameter T_{mid} defines the period when $SIP\ mid\_PROBE$ messages are sent again if the measured delay is below T_d [26]. Those two parameters are the most important in CAHP-C procedure and need to be set carefully as they affect the level of signalling overhead and the speed of detecting possible congestion. The main limitation of CAHP-C procedure was that

T_{pre} and T_{mid} parameters are constant, which means that they do not change if the network characteristics are improved. Thus, we are proposing CAHP-A procedure in which T_{pre} and T_{mid} parameters change adaptively according to current network characteristics (see also Figure 1). The CAHP-A procedure is presented in the next chapter.

2.2 CAHP-A

The SIP *pre_ PROBE* and *SIP mid_ PROBE* messages present additional traffic in the network and can be seen as signalling overhead. As the size of the messages is the same as in RTP traffic, the use of newly defined messages does not have significant effect on backbone traffic. However, such increase of signalling affects the SBC, which needs to handle additional messages. In this paper we are extending the CAHP-C procedure with adaptive calculation of T_{pre} and T_{mid}, as depicted in Figure 1 (bolded blocks).
In order to keep the signalling overhead as low as possible we propose that parameters T_{pre} and T_{mid} change adaptively according to characteristics of network (i.e. measured delay) as the networks differ by the ability to provide sufficient QoE. Some of them get congested likely (e.g. in congress centre) while the other used by a single user (e.g. at home) are not. With the Pre probe algorithm WLAN network is tested prior handover.
In the case that calculated delay D_{pre} is above threshold T_d the parameter T_{pre} change according to difference between D_{pre} and T_d. After handover, the WLAN network is tested with Mid probe algorithm. In case that calculated delay Dmid is below threshold Î¤d the parameter Tmid should change according to difference between D_{mid} and T_d. The calculations happen after the delay measurements (see bolded block in Figure 1).

In order to achieve such dependency we defined equations (1) and (2) which are used for calculation of T_{pre} and T_{mid} respectively:

$$T_{pre} = \begin{cases} N/A; & D_{pre} \leq D_{max} \\ T_{max} \cdot \left(\frac{D_{pre}-D_{min}}{D_{max}-D_{min}} - 1\right); & D_{max} < D_{pre} < 2 \cdot D_{max} - D_{min} \\ T_{max}; & D_{pre} \geq 2 \cdot D_{max} - D_{min} \end{cases} \quad (1)$$

$$T_{mid} = \begin{cases} T_{max}; & D_{mid} \leq D_{min} \\ T_{max} \cdot \left(1 - \frac{D_{mid}-D_{min}}{D_{max}-D_{min}}\right)^{\alpha}; & D_{min} < D_{mid} < D_{max} \\ N/A; & D_{mid} \geq D_{max} \end{cases} \quad (2)$$

where T_{max} represent maximum possible time set for T_{pre} or T_{mid}, D_{min} minimum (i.e. delay of non congested network) and D_{max} maximum delay that does not have effect on QoE (T_d in proposed algorithm). As seen the values for T_{pre} and T_{mid} are function of α and measured D_{pre} or T_{mid}. By selecting different values for α we can define different curves that define T_{pre} and T_{mid}. It is worth noting, that different α can be set for calculating T_{pre} and T_{mid}. Some possibilities are presented in Figure 2 and Figure 3, where parameter α was set to 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8, and 16.

When T_{pre} and T_{mid} are low more SIP probe messages are sent as they are more frequent and the possible degradation of service is detected faster. In such a manner we managed to maintain QoE of the user. When fewer messages are sent (T_{pre} and T_{mid} are high) the degradation of service may not be detected fast enough and degradation of QoE can be expected. However, more frequent sending of SIP *pre_ PROBE* and *SIP mid_ PROBE* messages T_{pre} and T_{mid} are high) increases signalling overhead. From Figure 2 and Figure 3 it can be seen that level of overhead changes also with parameter α. Thus, the α can be seen as the parameter that defines signalling overhead (i.e. the bigger the α, the bigger the signalling overhead). When α is approaching 0

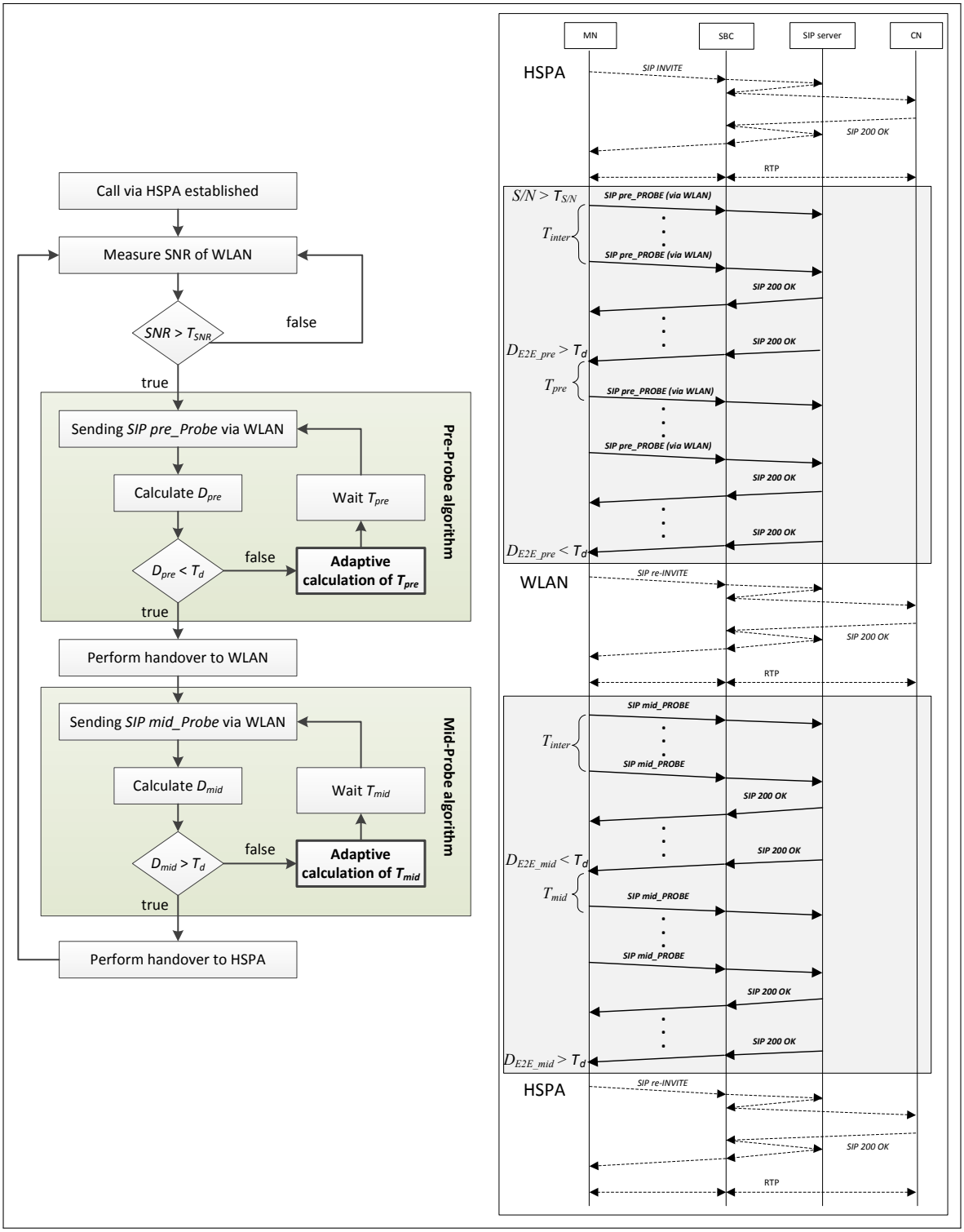


Figure 1: Flow diagram of CAHP-A procedure and message exchange.

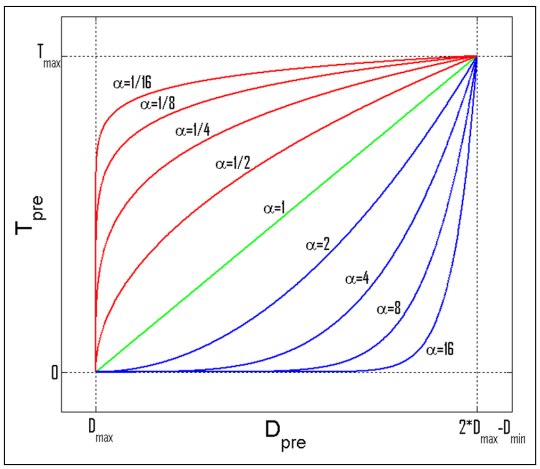


Figure 2: Dependency between SIP D_{pre} and SIP T_{pre} for different values of α

the values for T_{pre} and T_{mid} are approaching T_{max} and when α is approaching infinity the values the values for T_{pre} and T_{mid} are approaching 0 s.
In the proposed handover CAHP-A procedure two assumptions were made: (i) the procedure starts when MN is connected to both current and target networks and (ii) MN is capable of sending SIP messages via WLAN interface while, for RTP the HSPA network is still used. Both functionalities were necessary to perform seamless handover from HSPA to WLAN network. In the case that MN would not be capable of connecting to two different networks at the same time, first HSPA connection should be terminated and only then the establishment of WLAN connection would start. By second functionality the MN sends newly defined and standard signalling SIP messages via WLAN when still connected to HSPA. If MN would not have such capabilities, again HSPA connection should be terminated first and SIP messages could be sent only when WLAN connection would be established. Without those functionalities the connection would be lost several times which would affect QoE of the user.
The SIP *pre_PROBE* and SIP *mid_PROBE* messages are sent to SBC, through which RTP packets are also sent. Thus, the use of the proposed probes in the target networks can provide the real status of the ability of the network to provide an adequate level of QoE for IP telephony. As SIP protocol was used for sending the newly proposed messages, this approach is completely independent of the lower layers (i.e. transport, network and link). Furthermore, it can be used independently of the protocol used in lower layers and easily integrated in the operator's environment. It means that the role of lower layers protocols stays the same (i.e. providing IP connectivity) and that they do not need to be modified. The application uses theirs functionalities (e.g. measurement of signal, establish physical connection, IP connection).
Our proposal is also easily scalable, as operators add additional SBCs to their network, when the number of users and traffic flows increases and existing SBCs can not serve all the signalling and RTP traffic load.

3 Performance evaluation of the CAHP-A procedure

For performance evaluation of the proposed handover procedure presented in section 2 we developed a simulation model of a telecommunication system, which is discussed in [26]. The simulation model comprises two networks, WLAN and HSPA. Two hosts, MN and CN, that are using IP telephony as an application, were also defined. The G.711 codec was used for VoIP.

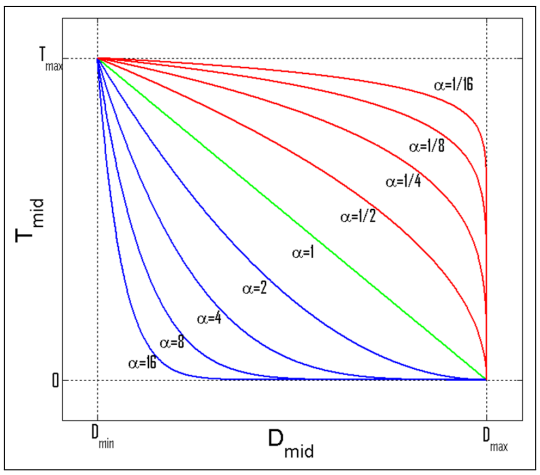


Figure 3: Dependency between D_{mid} and T_{mid} for different values of α

Silence suppression was not used, resulting in a constant RTP traffic stream of 100 packets/s. The network architecture of the simulation model, resembling real operator environment, is presented in Figure 4.

The following assumptions were made in the simulation scenario:

- HSPA network is always available;
- WLAN network has limited coverage, but its access link between Router 1 and IP network representing fixed operator, could become congested, as it aggregates traffic from all WLAN access points (AP);
- The usage of WLAN network is prioritized by the user, which means that MN will always try to perform a handover when this network is available;
- MN is a dual mode handset capable of sending RTP packets and SIP INVITE messages at the same time via different interfaces.

The first two assumptions were made just for simulation scenario in order to validate proposed procedure in environment of two networks, where one is reliable and the second is unreliable. We focused only on access link traffic load, as this is usually critical part of connection from MN to SBC. As described in chapter 2, in proposed procedure the congestion will be detected by measuring delays from MN to SBC.

The simulation model of the communication system was developed using the discrete event, object-oriented modelling simulation tool OPNET Modeler [26]. It has an open source code of commonly used protocols, which is very convenient for performance evaluation of user developed/enhanced mobility management mechanisms [28]. It enables network modelling and simulation for designing new protocols and technologies, together with performance evaluation of existing and newly developed optimized protocols and applications.

OPNET supports SIP telephony but it does not support handover on the application layer, thus some pre-defined process models that incorporate SIP procedures were customized [25]. Beside modification of process's functionalities, we also define the newly proposed SIP *pre_PROBE* and *SIP mid_PROBE messages*, which are used for congestion testing of the WLAN access network. Figure 4 shows the simulation network configuration. The MN is a dual mode terminal, capable of connecting to WLAN and to HSPA network. Both networks are connected via IP networks to the SBC. The CN is an IP phone connected to SBC.

To increase traffic in the WLAN network access link, other clients were added (represented by laptops in Figure 4), which generated additional UDP traffic in the LAN network.

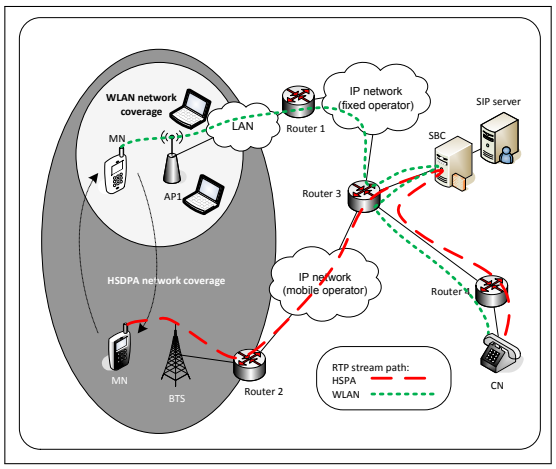


Figure 4: Network architecture of the simulation model.

3.1 Simulation scenario

The proposed CAHP-A procedure was evaluated in simulation model. In order to get statistically representative results, one long call that lasted 28,800 s (i.e. 8 hours) was simulated. However, the results can be easily applied to the scenario with several users that are making calls with the total sum of conversation time 28,800 s.
In defined network architecture WLAN network access link was randomly congested with additional UDP traffic, which was generated by additional VoIP clients. The duration of each "congestion" was distributed exponentially with mean value of 20 s. The time between two congestions was also distributed exponentially with mean value of 10 s. Such distribution enabled us to test the proposed procedure several times in different traffic conditions. The user movements were defined by changing SNR ratio. The SNR values were measured in real environment and imported in the simulation model.

Table 1: Simulation scenarios

Scenario	T_{mid}(s)	T_{pre}(s)	α
R_1	N/A	N/A	N/A
R_2	0	0	N/A
S_1	adaptive	adaptive	1/16
S_2	adaptive	adaptive	1/8
S_3	adaptive	adaptive	1/4
S_4	adaptive	adaptive	1/2
S_5	adaptive	adaptive	1
S_6	adaptive	adaptive	2
S_7	adaptive	adaptive	4
S_8	adaptive	adaptive	8
S_9	adaptive	adaptive	16

In order to evaluate the proposed procedure, we prepared eleven scenarios. Two first scenarios were defined to get reference results for two opposite situations. In the first reference scenario (R_1) the proposed procedure was not used and handover was made based on SNR only. In the second scenario (R_2) parameters T_{pre} and T_{mid} were set to 0 s, which gave us maximal sending frequency of newly defined SIP messages SIP *pre_PROBE* and SIP *mid_PROBE*. We

added additional 9 scenarios (S_1 - S_9) with CAHP-A procedure in which parameters α was set to 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8 and 16. All simulation scenarios are summarised in Table 1.

3.2 Simulation results

In the simulation several results were collected. From the user's point of view the end to end delay should not be significantly affected by performing handovers. We measured end to end delay of IP telephony for each packet. This enabled us to get the cumulative simulation time with end to end delay above 200 ms, which will be presented in results. The cumulative time when HSPA interface was used will be also presented as this affects the cost of communication, which is also very important to the user, as he wants to minimise the usage of expensive network (i.e. HSPA), while still having the appropriate QoE. The signalling increase caused by additional newly proposed messages during handovers was also tracked. If users perform large number of handovers the probe messages will increase signalling traffic and traffic load of the SBC, which is important from the operator's point of view. Thus, the number of overhead messages will be also presented in the results.
The end to end delay distribution for simulation scenarios is presented in Figure 5. For the sake of clarity only delays above 200 ms are presented, as those delays significantly affect QoE of the user. Delays above 400 ms occur in R_1 only, in which cumulative time, with end to end delay above 200 ms, presents 48.7% of all communication time. That means that in such conversation the QoE of the user is highly degraded. It can be seen that in all other scenarios the cumulative conversation time above 200 ms is much smaller (see percentages above histograms in Figure 5), resulting in QoE to be highly improved compared to R_1. The best results are measured in scenarios from S_7 to S_9. It can be seen that results in S_7 to S_9 are even better than in R_2, where T_{pre} and T_{mid} were set to 0 s as in R_2 the signalling overhead itself was causing additional congestion and deteriorate the results.

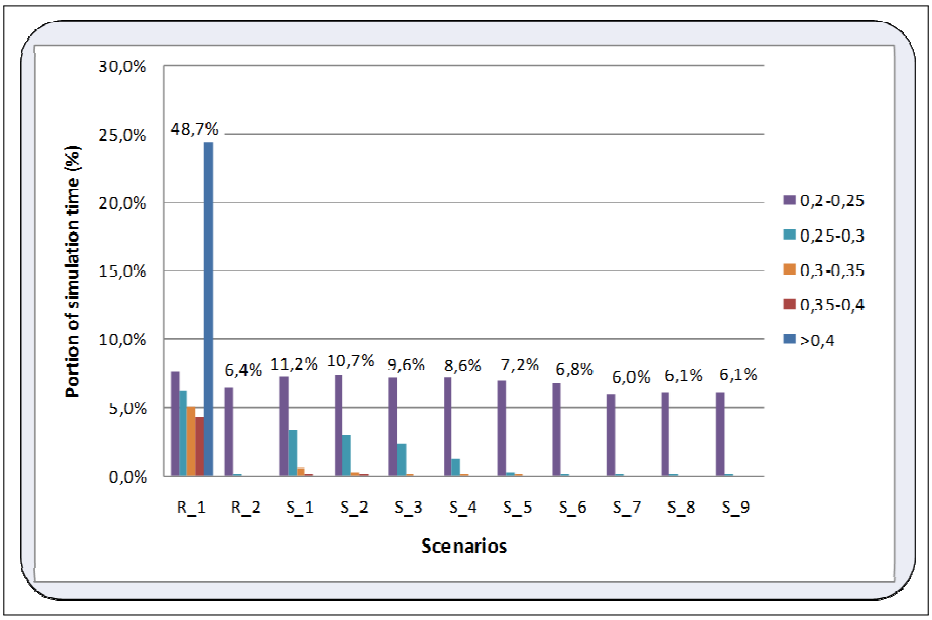


Figure 5: End to end delay distribution

The biggest share of cumulative simulation time above 200 ms is measured in R_1, thus we

took this scenario as a reference for normalizing other scenarios as presented in (1):

$$NSTaT_i = \frac{STaT_i}{STaT_{R1}} \cdot 100\% \tag{3}$$

where $NSTaT\hat{I}\square_i$ normalized simulation time above threshold (200 ms) of scenario i, $STa\hat{I}\square_i$ simulation time above threshold for scenario i, $STa\hat{I}\square T_{R1}$ simulation time with delay above 200 ms for scenario R_1. Normalized values are presented in Table 2.
It can be seen that the $NSTa\hat{I}\square_i$ is decreasing with increase of parameter α in scenarios S_1-S_9. The best results are achieved in S_7, where α was set to 4.
Cumulative conversation time when HSPA interface is used is also presented in Table 2. In R_1 the HSPA was used only for 16.4% of simulation time, as handovers were performed based on SNR only. In all other scenarios (R_2 and S_1 - S_9) the measured HSPA usage is between 52.0% and 59.6%, which give us only 7.6 percentage points of difference.

Table 2: Normalized cumulative simulation time above 200 ms

Scenario	$NSTa\hat{I}\square T_i$	Cum. conv. time on HSPA
R_1	100.0%	16.4%
R_2	13.1%	59.6%
S_1	23.1%	52.0%
S_2	21.9%	52.6%
S_3	19.7%	55.8%
S_4	17.6%	55.0%
S_5	14.7%	57.4%
S_6	13.9%	53.7%
S_7	12.3%	58.1%
S_8	12.6%	59.3%
S_9	12.5%	57.7%

Normalized signalling overhead caused by newly proposed SIP*pre_PROBE* and SIP *mid_PROBE* messages is presented in Figure 6. Maximum number of signalling overhead messages is, as expected, measured in R_2 ($T_{pre} = T_{mid} = 0$ s) in which about 273 thousand overhead messages were sent. This scenario was taken as a reference for normalizing other scenarios as presented in (2):

$$NSO_i = \frac{nSM_i}{nSM_{R2}} \cdot 100\% \tag{4}$$

where NSO_i presents normalized signalling overhead for scenario i, nSM_i number of signalling messages of scenario i, nSM_{R2} number of signaling messages of scenario R_2.
From Figure 6 it can be seen that the signalling overhead is highly decreased in scenarios form S_1 to S_9 (where CAHP-A was used), with lowest overhead in S_1 (only 4.4 % of R_2 signalling traffic). The results show that parameter α defines signalling overhead (i.e. the bigger the α, the bigger the signalling overhead) proofing our theoretical analysis in chapter 2. As in the reference scenario R_1 CAHP-A procedure was not used, no messages were sent, thus we get result of 0.0

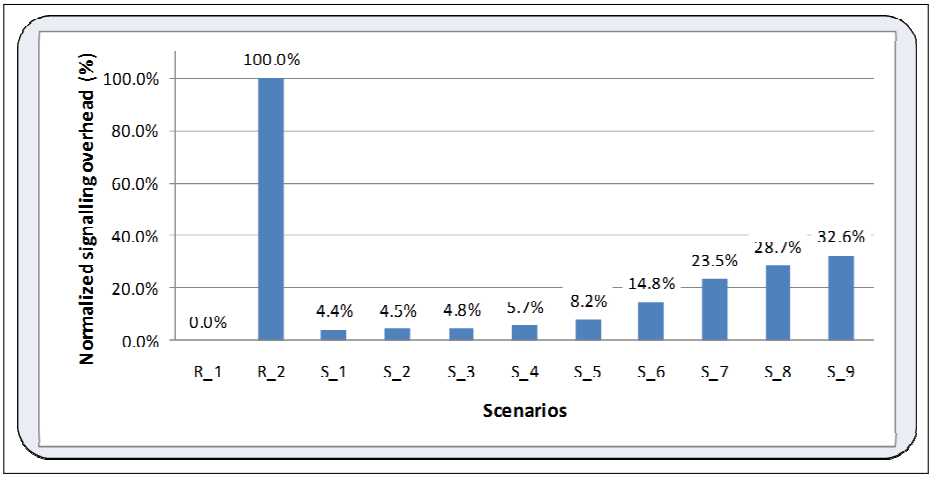


Figure 6: Normalized signalling overhead

3.3 Discussion

Among presented results we are looking for optimal values for parameter α (defining T_{pre} and T_{mid}). As the results presenting simulation time when HSPA was used (Table 2) are not very dependent on T_{pre} and T_{mid} (7.6 percentage points of difference among scenarios), these results are excluded from further analysis. From the results of end to end delay distribution (Figure 5) and normalized signalling overhead (Figure 7) it can be seen, that high signalling overhead results in low end to end delay (i.e. good QoE) and vice versa. Thus, compromise will be needed between user's and operator's objectives.

Users want QoE to be as good as possible. To assure this, the end to end delay should be sufficiently low during VoIP session also when using unreliable network (i.e. WLAN network in or case). In all scenarios we measured end to end delay between 250 and 300 ms, thus end to end delay of 300 ms presents upper limit for sufficient QoE. From results (Figure 5) it can be seen that scenarios R_2 and S_6 - S_9 meet that condition as no packet appeared with delay above 300 ms.

From the operators point of view the signalling overhead needs to be low in order to save SBC resources. The limit for signalling overhead is harder to set, thus we defined two conditions. Our requirement is that signalling overhead is lowered for at least (a) 70% and (b) 80% compared to R_2. Results presented in Figure 6 show that scenarios S_1 to S_8 all meet condition (a) and scenarios S_1 to S_6 meet condition (b).

As seen, scenario S_6 appears in all groups. Based on defined conditions the optimal setting of parameter α is 2. By using those setting, cumulative time with measured end-to end delay above 200 was decreased for 86% (from 48.7% to 6.8%) compared to reference scenario R_1, while signalling overhead was lowered for 85% compared to reference scenario R_2.

The comparison of the results for optimal setting ($\alpha = 2$) with the results obtained for different constant values of T_{pre} and T_{mid} described in [26] shows, that when using CHAP-A, the end-to-end delay is significantly decreased, in particular for the delays higher than 250 ms, while the signalling overhead stays almost the same.

4 Conclusions

In this paper we presented a novel adaptive SIP based procedure for congestion aware handover in heterogeneous networks. With newly proposed Pre-probe and Mid-probe algorithms the handover decision is (in addition to SNR) based also on target network congestion status. In order to analyze and evaluate the proposed procedure several scenarios were prepared. The results show that using the proposed adaptive CAHP-A procedure, the QoE of VoIP users was significantly improved, compared to reference scenario, in which only signal strength was used as the decision for handover. This is achieved by eliminating handovers to unreliable highly congested target network, which could cause degradation of service. With the CAHP-A procedure the unreliable network is tested also after handover. In the case of the detection of congestion the handover back to reliable network is triggered, which prevents the QoE degradation. In the proposed procedure SIP protocol was used for sending the proposed probe messages. As it runs on the application layer, our solution is completely independent of the underlying access technologies and thus applicable easily to next generation wireless systems. Furthermore, it is also independent of the lower layer protocols used. Another advantage of using SIP protocol for messages is that SIP usage is increasing in operators environments. In this paper we have focused only on measurement of end-to-end delay based on transmission of the proposed SIP*pre_PROBE* and SIP*mid_PROBE* messages. In order to further improve the handover decision algorithm, we will in our further work evaluate more parameters (e.g. user profiles, other network parameters, context awareness) to make even more efficient handover decision. In addition, the historic information about particular network can also be used in order to make the handover decision even more efficient, in particular where there are more than two WLAN networks available.

Bibliography

[1] J. M. Barja , C. T. Calafate, J.-C. Cano, P. Manzoni (2011); An overview of vertical handover techniques: Algorithms, protocols and tools, *Computer Communications*, 34(8):985-997

[2] Q.-T. Nguyen-Vuong et al. (2008); A user-centric and context-aware solution to interface management and access network selection in heterogeneous wireless environments, *Computer Networks*, doi:10.1016/j.comnet.2008.09.002

[3] E. Aruna, R.S. Moni (2012); Optimization of Vertical Handoff Decision Algorithm for Wireless Networks, *International Journal of Computers Communications & Control*, 7(2):218-230, DOI: http://dx.doi.org/10.15837/ijccc.2012.2

[4] F. Patriarca, S. Salsano, F. Fedi; Efficient Measurements of IP Level Performance to Drive Interface Selection in Heterogeneous Wireless Networks;

[5] S. Ghahfarokhi, N. Movahedinia (2012); Context gathering and management for centralized context-aware handover in heterogeneous mobile networks. *Turk J Elec Eng & Comp Sci*; 20(6), doi:10.3906/elk-1101-1042

[6] C. Lozano-Garzon, N. Ortiz-Gonzalez, Y. Donoso (2013); A Proactive VHD Algorithm in Heterogeneous Wireless Networks for Critical Services, *International Journal of Computers Communications & Control*, 8(3): 425-431, DOI: http://dx.doi.org/10.15837/ijccc.2013.3

[7] C. Perkins, IP Mobility Support for IPv4, RFC 3344, August 2002.

[8] I. F. Akyildiz, J. Xie, S. Mohanty (2004); A survey of mobility management in next-generation all-IP-based wireless systems, *IEEE Wireless Communications*, 16-28.

[9] H. Fathi, R. Prasad, S. Chakraborty (2005); Mobility management for VoIP in 3G systems: evaluation of low-latency handoff schemes, *IEEE Wireless Communications*, 96-104.

[10] P.M.L. Chan, R.E. Sheriff, Y.F. Hu, P. Conforto, C. Tocci Mobility management incorporating fuzzy logic for heterogeneous a IP environment. *IEEE Communications Magazine* 01/2002; DOI:10.1109/35.968811

[11] S. Mohanty, I. F. Akyildiz (2007); Performance Analysis of Handoff Techniques Based on Mobile IP, TCP-Migrate, and SIP, *IEEE Tranactions on mobile computing*, 6(7):731-747.

[12] K. El Malki (2005); Low Latency Handoffs in Mobile IPv4, INTERNET-DRAFT, October 2005.

[13] J.-Y. Chen , C.-C. Yang, L.-S. Yu (2010); HH-MIP: An Enhancement of Mobile IP by Home Agent Handover. *EURASIP Journal on Wireless Communications and Networking*. 2010. doi:10.1155/2010/653838

[14] L. Ma, F. Yu, V. C. M. Leung, Tejinder Randhawa (2004); A new method to support UMTS/WLAN vertical handover using SCTP, *IEEE Wireless Communications*, 44-51.

[15] S. J. Koh, M. Jeong Lee, Maximilian Riegel, Mary Li Ma, Michael Tuexen (2004); Mobile SCTP for Transport Layer Mobility, *IETF* Internet Draft.

[16] J. Zhang, H. Chan, V. Leung (2007); A SIP-based seamless-handoff (S-SIP) scheme for heterogeneous mobile networks. *IEEE WCNC*.

[17] G. P. Silvana, H. Schulzrinne, SIP and 802.21 for Service Mobility and Pro-active Authentication, *Communication Networks and Services Research Conference*, 2008, doi 10.1109/CNSR.2008.61

[18] N. Banerjee, W. Wu, K. Basu, S. K. Das (2004); Analysis of SIP-based mobility management in 4G wireless networks, *Computer Communications*, 27, 697-707.

[19] Ali M, Liang L, Sun Z, Cruickshank H. (2011); Optimization of SIP Session Setup for VoIP over DVB-RCS Satellite Networks, *Inderscience International Journal of Satellite Communications Policy and management (IJSCPM)*, 1 (1): 55-76. doi: 10.1504/IJSCPM.2011.039741

[20] M. Bonola, S. Salsano, A. Polidoro. UPMT: Universal Per-application Mobility management using Tunnels. *IEEE GLOBECOM* 2009, 30 Nov - 4 Dec 2009, Honolulu, Hawaii.

[21] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: Session Initiation Protocol, *IETF RFC3261*, June 2002.

[22] I. Basicevic, M. Popovic, (2008); Use of SIP Protocol in Development of Telecommunications Services, *The Journal of The Institute of Telecommunications Professionals*, 01/2008, 2.

[23] T. Aljaz, B. Imperl, A. Svigelj (2008); Border gateway function performance requirements for the lawful intercept of voice at IMS architecture. *AEU, Int. j. electron. commun.* (Print), 62(8):610-621, doi: 10.1016/j.aeue.2007.08.006.

[24] J. Hautakorpi, G. Camarillo, R. Penfield, A. Hawrylyshen, M. Bhatia, Requirements from SIP (Session Initiation Protocol) Session Border Control Deployments, Internet-Draft, October 23, 2008, URL: http://www.ietf.org/internet-drafts/draft-ietf-sipping-sbc-funcs-07.txt

[25] R. Libnik, A. Svigelj, G. Kandus (2008); Performance evaluation of SIP based handover in heterogeneous access networks, *WSEAS transactions on communications*, 7(5): 448-458.

[26] R. Libnik, A. Svigelj, G. Kandu (2010); A novel SIP based procedure for congestion aware handover in heterogeneous networks. *Computer Communications*, 33(18): 2176-2184, doi: 10.1016/j.comcom.2010.09.007.

[27] OPNET Technologies (2014); http://www.opnet.com.

[28] R. Libnik, G. Kandus, A. Svigelj (2011); Simulation model for performance evaluation of advancde SIP based mobility management techniques, *International Journal of Communications*, 5(1):26-35.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):702-717, October, 2015.

An Anticipatory Control for a Flexible Manufacturing System Based on the Perception of Mobile Units Using WSNs

J.A. Silva-Faundez, C. Duran-Faundez, P. Melin, C. Aguilera

Joaquin Alder Silva-Faundez, Cristian Duran-Faundez*,
Pedro Melin, Cristhian Aguilera
Departamento de Ingeniería Eléctrica y Electrónica,
Universidad del Bío-Bío
Avda. Collao 1202, Casilla 5-C, Concepción, Chile
E-mail: joaquin.silvaf@ingenieros.com, {crduran, pemelin, cristhia}@ubiobio.cl
*Corresponding author: cduran@ubiobio.cl

Abstract: In this paper, we design and evaluate a control system which, by using as input RSSI measures, allows anticipatory movements of robotic arms decreasing idle times at the CIMUBB Laboratory. Classical Log-Normal model, which relates the strength of a signal received by a node with the distance at which the sender of the signal is, was adopted. The hidden state of the system is determined by the Extended Kalman filter which allows us to estimate the distance and the speed of pallets moving over a closed-loop conveyor belt. From these estimates, remaining time in which the pallet will get to a stopping point near the robot is determined. This information is finally processed by a controller to determine the instant at which the robot must operate and handle the pallet. Both, a Proportional-Integral and a Fuzzy controller, were implemented and evaluated. Results show the feasibility of using wireless signals to accomplish the described goal, with some practical restrictions.
Keywords: Anticipatory control, intelligent manufacturing, wireless sensor networks, computer applications, localization.

1 Introduction

In the last few years, Wireless Sensor Networks (WSNs) [1] have attracted the industrial sector because of their many advantages such as low power consumption, high flexibility and scalability. Indeed, many application cases of WSNs, including factory and building automation, inventory management, efficient irrigation, cattle tracking and many others, nowadays exist [2–6]. Moreover, latest works on new approaches, such as the "active products"and the "ambient intelligence", have considered WSNs as a supporting technology because they represent an efficient way to embed autonomous intelligence, sensing and communication capabilities on multiple objects (see, for example, works in [7–9]). On the other hand, localization and positioning of people, machines, vehicles, and other entities moving in automated environments are highly desired capabilities in modern industries and services. Indeed, these capabilities may allow significant improvements in process control, monitoring, tasks distribution, resources optimization, safety, and others. In certain application cases, these can be achieved thanks to very accurate location systems, such as the Global Positioning System (GPS) [10] and Ubisense [11]. In the matter of WSNs in localization applications, these are interesting for many cases where, given the number of units to identify, limitations of the technologies, and other factors, accurate solutions such as GPS or UWB-based indoor location technologies represent too important energy, economic or structural costs, or when we want to take advantage of the already installed wireless sensing equipment. Here, in order to provide distance estimations, many works have applied the Received Signal Strength Indicator (RSSI), because it is inherently provided by most of the currently applied transceivers in wireless sensor nodes.

Copyright © 2006-2015 by CCC Publications

In [12] we reported our first experiments on the 1-Dimensional (1-D) positioning of pallets moving through a closed loop conveyor belt at the CIMUBB Laboratory [13]. We aimed to minimize the idle time of a robot arm by integrating information about the position of the pallets approaching its Flexible Manufacturing Cell (FMC). To do so, pallets carried WSNs devices periodically sending messages to a receiver node near the robotic arm. RSSI traces were captured and analyzed concluding that, even if this indicator presents high instabilities in closed environments, this metric could be useful in our case study.

In this paper, we design and evaluate a control system which, by using RSSI measures as input, allows anticipated movements of robotic arms decreasing idle times at the CIMUBB Laboratory, continuing works in [12]. In the system, we adopted the classical Log-Normal model which relates the strength of a signal received by a node with the distance at which the sender of the signal is. The hidden state of the system is determined by the Extended Kalman filter which allows us to estimate the distance and the speed of the objects of interest (the pallets). From these estimates, remaining time in which the pallet will get to the stopping position near the robot is determined. This information is finally processed by a controller to determine the instant at which the robot must operate and manipulate the pallet. Both, a Proportional-Integral (PI) controller and a fuzzy controller, were implemented and evaluated. Results show the feasibility of using wireless signals to estimate distances in an indoor real-world manufacturing environment, such as the CIMUBB Laboratory, considering some restrictions.

The remainder of this article is organized as follows: In Section 2 we present the CIMUBB Laboratory, describing its physical configuration. In Section 3 we design the anticipatory control system, describing the developed models, filtering, and controllers to be evaluated. In Section 4 we describe our test implementation, exposing performed tests and results in Section 5. Finally, in Section 6 we conclude and give some future directions.

2 Case study: CIMUBB Laboratory

CIMUBB laboratory is a small scale fully automated Flexible Manufacturing System (FMS) consisting of three flexible manufacturing cells. All of the flexible cells are integrated by means of a closed loop conveyor belt through which several parts travel on pallets. The transportation between the various cells and the conveyor belt is made by preprogrammed robotic arms. Due to the way this system is implemented, most of the time, the robots do not have any information about the position of the pallets to be handled. Indeed, only when a pallet passes over a stop station (where magnetic sensors are located), the system knows the position of a particular pallet. This produces a long-term significant idle time between the time the pallet is identified and stopped until the moment the robot handles it (this is where this work acts). The implementation adopted in this work is similar to the one in [12]. Pallets carrying WSNs devices periodically send messages to a receiver node near the target FMC stop station. RSSI traces are collected and processed by a control system. The system sends orders to the robotic arm, trying to match the time of the pallet's arrival with the end of a robot movement reaching an approaching position that allows it to grasp the pallet with minimal waiting time.

Obviously, this is not the industrial solution. The simplicity of this system should enable us to meet the same goal by simply placing sensors before the stop station. Our current research subjects aim to provide positioning systems and anticipatory control (among others) in more complex production environments. This work is a study that will allow us to determine (as has been done in other work environment) how usable the RSSI is as a metric to estimate distance and speed of moving objects in real-world indoor environments, initially, in a simple cyclic system such as the CIMUBB laboratory.

Regarding the problem of using an RSSI-based location, our previous studies have resulted

in the characterization of electromagnetic (EM) waves allowing to observe the manner in which signals behave in this system. Now, it is necessary to characterize the behavior of the object of interest (i.e., the pallets) at which we want to estimate distances. The behavior of a pallet is given by the mathematical model of its position regarding time. Therefore, it is necessary to know the technical details of the conveyor system. The conveyor is a closed loop conveyor system made of metal. It is supported by pillars that rise the system 1 meter above the building's floor. Circulating pallets moving on the conveyor are dragged by a belt consisting of two plastic chains running in parallel and driven by an AC motor. This rotates at a constant speed giving the pallets a forwarding speed of 0.19[m/s] over straight lines, which decreases to 0.18[m/s] at the corners. activated via a switch which makes it to operate independently from the FMS. The location of the flexible cells, together with the dimensions of the conveyor belt, are shown in Figure 1(a). In this work, we focused on the Machining FMC which is shown in Figure 1(b).

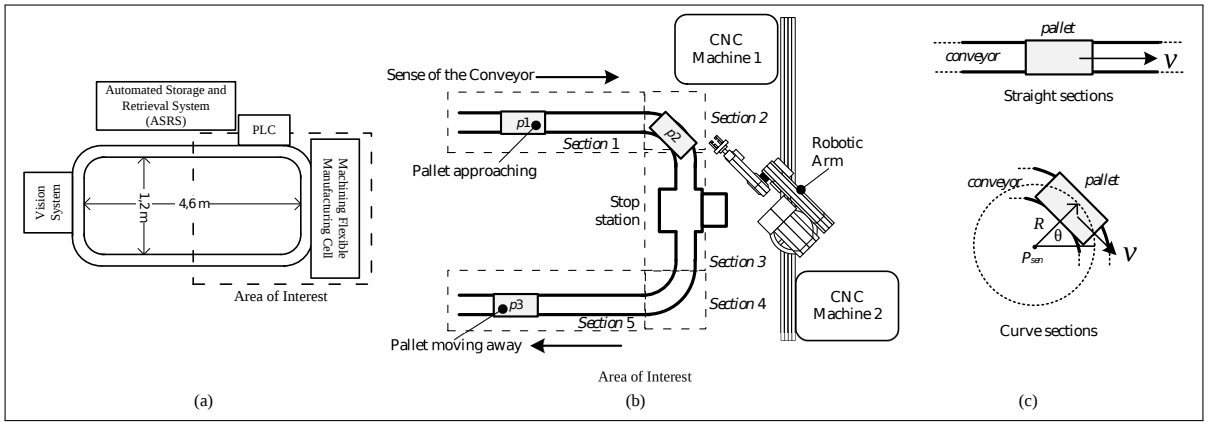


Figure 1: Case study.

Now, knowing the conveyor's geometric structure it is possible to model the path of the pallets within the area of interest. The pallets path, over the area of interest, can be divided into five sections, as shown in Figure 1(b): three of them, in which pallets perform straight movements (sections 1, 3 and 5), and two for curved movements (sections 2 and 4). Thus, the position of a pallet over time can be defined as:

$$\overrightarrow{r}_k = \begin{cases} \overrightarrow{r}_{k-1} + \overrightarrow{V}.\Delta T & \text{during straight movements} \\ R.\left[\left(\cos\left(-\omega.\Delta T + \theta\right) + P_{cenx}\right).\hat{i} + \left(\sin\left(-\omega.\Delta T + \theta\right) + P_{cenx}\right).\hat{j}\right] & \text{during curved movements} \end{cases} \tag{1}$$

were $\theta = \arctan\left(\frac{P_{k-1_y} - P_{ceny}}{P_{k-1_x} - P_{cenx}}\right)$ and $\omega = \frac{\left|\overrightarrow{V}\right|}{R}$. For both of cases, $\overrightarrow{r}$ and $\overrightarrow{V}$ represent the position and velocity vectors of the pallet, respectively. The unit vectors i and j, take the direction of the Cartesian reference system xy. ΔT is an increment in time. Thus, the equations give the new position of the pallet at a determined time increment from its current position (r_{k-1}). Particularly, during curved movements, ω is the angular velocity of the pallet calculated from its rapidity and the radius of the curve (R). The pallet's position is reflected in the angle that has about the center of rotation P_{cen} (see Figure 1(c)).

We established a reference system in the plane xy, with its origin at the bottom left of the conveyor (end of Section 5). From this reference coordinate, various reference points are fixed in order to mark the beginning and the end of each section of the path. Along with these references,

the location of the data collector node is set. This point will serve as a reference point for distance estimation. This node is located 24 centimeters from the stop station.

3 System design

The proposed "*Anticipatory Control System*"(ACS) will be in charge of detecting the approaching of a pallet to the manufacturing cell and estimate the time at which the robot must act in anticipation of its arrival. The system estimates the proximity from available RSSI information supplied by the collector node (a node located near the controlled robotic arm). These values are calculated by the node upon receipt of datagrams sent by the wireless nodes located on the pallets. These act as beacons regularly indicating their presence. The intensity with which the collector gets beacons' signals can be related to the distance at which pallets are through a mathematical model describing their behavior. However, these signals are not free from interferences. Indeed, various factors affect the radio-frequency (RF) waves such as the composition of the surroundings (walls, objects, etc.), the orientation of the transmitters, signals bouncing, etc. From experiments, generated disturbances were observed and analyzed, observing that these phenomena do not completely affect the tendency of the received signal regarding the studied models. Then, it is necessary to develop the estimation system using a mathematical model that approximates the EM waves path loss and an algorithm allowing noise filtering or the estimation considering such alterations. For this task the Extended Kalman filter (EKF) is used as a tool.

Along with estimating the proximity of a pallet, the system handles the operation of the SCORBOT-ER 9 robot manipulator and the control of the FMS's conveyor belt. In addition, we implemented an interface that allows a user to interact with the system. Figure 2 illustrates a general view of the system, exposing the different components and interacting devices.

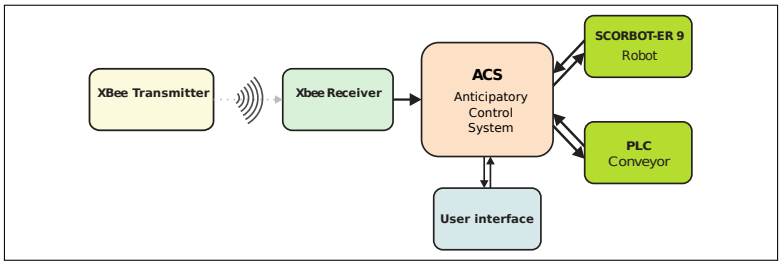


Figure 2: ACS general scheme.

3.1 System model

One of the key aspects on the development of the ACS is to obtain a mathematical model for path loss of the wireless signal, phenomenon related to the distance between the sender and the receiver nodes. Such a system can be modeled considering it as a "gray-box system"because a part of the dynamics of the RF signals is known, existing different prediction models. However, many variables and interactions with the environment are unknown. One of the models used to describe the propagation loss is the "Log-Normal Shadowing", whose equation is given by:

$$PL(d)[dB] = \overline{PL}(d_0)[dB] + 10.\gamma.\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma[dB] \tag{2}$$

were X_σ represents noise and interference as a zero-mean Gaussian random variable with standard deviation σ. In order to find the value of power received by the receiver node, we must substract

this attenuation to the transmission power of the signal Pt, so we have:

$$PL_{rcvd}(d)[dBm] = Pt[dBm] - PL(d)[dB] = -10.\gamma.\log_{10}\left(\frac{d}{d_0}\right) + \left(P_t - \overline{Pl}(d_0)\right) - X_\sigma \quad (3)$$

A relationship between the received strength and the distance between the nodes can be obtained by the propagation model . Along with this variable, it is also necessary to know the speed with which the pallet approaches, thus we must find a relationship between the pallet's speed and its distance to the stop station. An alternative is the derivation of the speed value from distance estimations, however we should consider that this value represents only a measure of the distance change rate and not the actual speed of the pallet. Despite this, both of the values are related, therefore it can be used as an indicator of how quick the pallet moves.

By derivation we obtain the instantaneous speed with which the pallet approaches: $\frac{\delta d}{\delta t} = \nu$, in this work, considered as a constant. This expression is discretized, obtaining the distance for instant $tk + 1$:

$$d(tk + 1) = \nu(tk)T + d(tk) \quad (4)$$

Next, and in order to identify a discrete model parameters, a linear difference equation can by used as $y(t) + a_1y(t-1) + \ldots + a_ny(t-n) = b_1u(t-1) + \ldots + b_mu(t-m)$, where $u(t)$ and $y(t)$ are the input and output, respectively, and a_n and b_m are the system's parameters. From the last expression it is possible to determine the value of the next output solving the difference equation for $y(t)$ and predict the system's output from the system's observed measurements and its parameters [14]. Considering a system of N equations and n unknowns, the estimated output can be written as [14]:

$$\hat{Y}(N) = M(N).\hat{\theta} \quad (5)$$

where the variables have, respectively, the structures: $Y(N) = [y(1) \quad y(2) \quad \ldots \quad y(N)]^T$, $\hat{Y}(N) = [\hat{y}(1) \quad \hat{y}(2) \quad \ldots \quad \hat{y}(N)]^T$, and $M(N) = [m(1) \quad m(2) \quad \ldots \quad m(N)]^T$. In order to estimate the parameters of Equation 5, Least Squares method is used. The predictions made using estimated parameters will have an associated error defined as $E = Y(N) - \hat{Y}(N)$.Then,the Least Squares method reduces this error by minimizing a functional which is a sum of squared errors, i.e., $\min\left(J(\hat{\theta}) = E^T(N).E(N)\right)$, with $J(\hat{\theta}) = \sum_{K=1}^{N} e^2(k)$. In order to minimize the error, we must have $\frac{\partial J(\hat{\theta})}{\partial \hat{\theta}} = 0$. Developing for θ from previous equations, we obtain the following expression for obtaining parameters which minimizes the prediction error that can be used to estimate parameters of the propagation loss model :

$$\hat{\theta} = \left[M(N)^T.M(N)\right]^{-1}.M(N)^T.Y(N) \quad (6)$$

Let us consider the propagation model in Equation 3. This can be developed to adopt the structure of Equation 5, by doing:

$$PL_{rcvd}(d) = -10.\gamma.G(tk) + W \quad (7)$$

where $G(tk) = \log_{10}\left(\frac{d(tk)}{d}\right)$ and $W = \left(Pt - \overline{Pl}(d_0)\right)$. For modeling purposes, the random variable X_σ was considered as zero, obtaining a log-normal model. Finally, expressing Equation 7 in matrix form we have:

$$PL_{rcvd}(d) = \begin{bmatrix} G(tk) & 1 \end{bmatrix}. \begin{bmatrix} -10.\gamma \\ W \end{bmatrix} \quad (8)$$

In this way, the variables are: (1) The system output $Y(N) : PL_{rcvd}(d)$, (2) the measurements vector $M(N) : \begin{bmatrix} G(tk) & 1 \end{bmatrix}$, and (3) the parameters vector $\theta : \begin{bmatrix} -10.\gamma \\ W \end{bmatrix}$

In order to generate useful data for estimations, various experiments were performed. In these, the pallets circulate in the conveyor while the robot takes one of the predetermined positions used in previous experiments. These experiments are performed with each of the three transmitting nodes adopted in this work (3 variants of the Digi's XBee family, explained in Section 4). As an example, Figure 3 illustrates the parameters estimation for a XBee wire antenna transmitter. Table 1 presents estimated parameters for each of the used nodes.

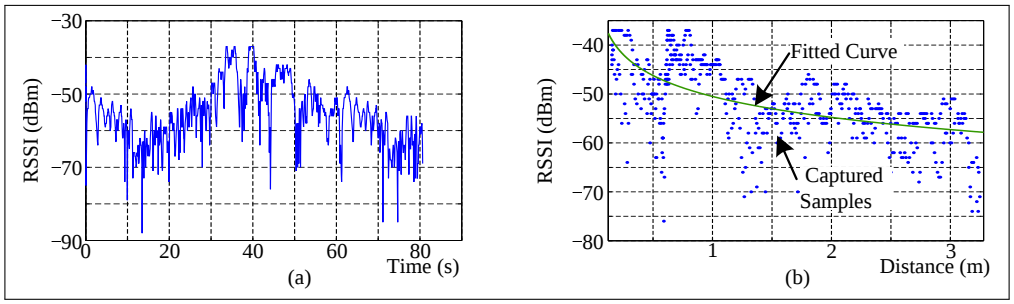


Figure 3: Parameter estimation for a XBee wire antenna transmitter case.

Table 1: Parameters of transmitter nodes.

Transmitter node	Pt	$\overline{Pl}(1.5[m])$	γ
XBee chip antenna	0[dBm]	59.156[dBm]	1.142
XBee wire antenna	0[dBm]	53.0017[dBm]	1.438
XBee Pro wire antenna	16[dBm]	51.376	1.065

The devices sampling time was set to 54[ms], however, the time observed during measurements is not constant. This has a mean deviation of 82[ms] with a standard deviation of 34[ms]. Therefore, the sampling time used in the model will be estimated during the algorithm execution, by the time difference between a RSSI measurement and the previous one. To summarize, the model of the system is given by the following equations:

$$d(tk+1) = v(tk)T + d(tk) \tag{9}$$

$$v(tk+1) = v(tk) \tag{10}$$

$$PL_{rcvd}(d) = -10.\gamma.\log_{10}\left(\frac{d(tk)}{d_0}\right) + \left(Pt - \overline{Pl}(d_0)\right) \tag{11}$$

3.2 Signal filtering

For this work the Extended Kalman Filter (EKF) is used. EKF is a variant of the traditional Kalman filter that addresses the problem of estimating states in nonlinear systems. Let us consider the nonlinear model of the process in a discrete-time state variable representation, i.e., representing the unknown state of the system as $x_k = f(x_{x-1}, u_{k-1}, v_{k-1})$, with known input u_k, and the observable output $z_k = h(x_k, n_k)$. v_k and n_k are random variables representing the process and measurement noises, respectively. It is assumed that the distribution densities are Gaussian and both v_k and n_k are random variables independent to each other, so we can define $p(x_{k-1}|z_{1_k-1}) = N(X_{k-1|k-1}, P_{k-1|k-1})$, $p(v_{k-1}) = N(O, Q)$, and $p(n_k) = N(O, R)$. However, being a non-linear model, distribution densities $p(x_k|z_{1:k-1})$ and $p(x_k|z_{1:k})$ for next instant k may not be Gaussian, therefore, they are approximated to a Gaussian density. At the densities

and distribution of v_{k-1} and n_k, Q and R are the covariance matrices of process disturbance and measurement disturbance covariance, respectively [15]. So, the filter operation may be divided into two stages:

1. **Prediction.** This stage generates *a priori* prediction of the state $x_{k|k-1} = f(x_{k-1}, u_k, 0)$ and the error covariance $P_{k|k-1} = F^x_{k|-1}.P_{k-1}.\left(F^x_{k-1}\right)^T + F^v_{k-1}.Q_{k-1}.\left(F^v_{k-1}\right)^T$, considering all the information available at that time.
2. **Correction.** At this stage, previous estimates are corrected, then generating *a posteriori* estimates of $x_{k|k}$ and $P_{k|k}$. For this, we calculate the Kalman gain as $K_k = P_{k|k-1}.\left(H^x_k\right)^T.\left[H^x_k.P_{k|k-1}.\left(H^x_k\right)^T + H^\eta_k.R_k.\left(H^\eta_k\right)^T\right]^{-1}$. This factor minimizes the error covariance of the new state estimation. So, new estimates are calculated as $x_{k|k} = x_{k|k-1} + K_k.\left(z_k - h(x_{k|k-1}, 0)\right)$ and $P_{k|k} = \left(I - K_k.H^x_k\right).Pk|k-1$.

Matrices F^x_{k-1}, F^v_{k-1}, H^x_k and H^η_k are obtained as Jacobians of the system's function and of the output function, i.e., $F^x_{k-1} = \frac{\partial f}{\partial x}\left(x_{k-1|k-1}, u_{k-1}, 0\right)$, $F^v_{k-1} = \frac{\partial f}{\partial v}\left(x_{k-1|k-1}, u_{k-1}, 0\right)$, $H^x_k = \frac{\partial h}{\partial x}\left(x_{k|k-1}, 0\right)$, and $H^\eta_k = \frac{\partial h}{\partial \eta}\left(x_{k|k-1}, 0\right)$.

The EKF is a recursive algorithm. Therefore, the prediction and correction stages are executed in cycles while the algorithm is running. For more details about an EKF recursive implementation see [16]. On the application of the filter to the developed system it is necessary to obtain matrices F^x_{k-1}, F^v_{k-1}, H^x_k and H^η_k, with respect to the exposed model in Equations 9, 10 and 11. These are defined as:

$$F^x_{k-1} = \begin{vmatrix} \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial d} \\ \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial d} \end{vmatrix} = \begin{vmatrix} \frac{\partial d(tk+1)}{\partial v(tk)} & \frac{\partial d(tk+1)}{\partial d(tk)} \\ \frac{\partial v(tk+1)}{\partial v(tk)} & \frac{\partial v(tk+1)}{\partial d(tk)} \end{vmatrix} = \begin{vmatrix} T & 1 \\ 1 & 0 \end{vmatrix} \quad ; \quad F^v_{k-1} = \frac{\partial f}{\partial v} = \mathbf{I} \quad ;$$

$$H^x_k = \begin{vmatrix} \frac{\partial h}{\partial v} & \frac{\partial h}{\partial d} \end{vmatrix} = \begin{vmatrix} \frac{\partial PL_{rcvd}(tk)}{\partial v(tk)} & \frac{\partial PL_{rcvd}(tk)}{\partial d(tk)} \end{vmatrix} = \begin{vmatrix} 0 & -\frac{10.\gamma}{d(tk).\ln(10)} \end{vmatrix} \quad ; \quad H^\eta_k = \frac{\partial h}{\partial \eta} = \mathbf{I} \qquad (12)$$

To initialize the algorithm it is necessary to designate initial values for the estimated states matrix and the error covariance matrix. In case of $x_{k|k-1}$, it is considered that during the system startup the pallet is far from the stop station and outside the area of interest. The pallet's initial speed is considered as zero. For the case of the $P_{k|k-1}$ matrix, it is initialized to a small value different but close to zero, because if the covariance is zero the filter will keep the last estimated value without innovating.

Matrices of process perturbation covariance and measurement are assumed to be constant although in practice this does not happen like that. Values of matrix Q related to the process, are generally more difficult to obtain, but it is presumed that the variance values of the process are low. Matrix R related to measurements can be predicted from the output observable values. From experimentation, a standard deviation of the measurements less than 0.5[dBm] for the case of the chip and wire antenna XBee nodes and less than 2.5[dBm] in the case of XBee Pro nodes, is observed. However, these values, measured in a quasi-stationary regime (i.e. without movements of the node-of-interest and by controlling as many environmental variables as possible), increase in dynamic conditions reaching a deviation of 5[dBm]. Due to this instability in measurements, an intermediate value for the deviation of 2[dBm] is considered. In this way, variances arbitrarily selected for the process variables and measurements are $\sigma^2_d = \sigma^2_v = 0.002$ and $\sigma^2_{PL_{rcvd}}$. The disturbance covariance matrices are $Q = \begin{vmatrix} 0.002 & 0 \\ 0 & 0.002 \end{vmatrix}$ and $R = 4$.

3.3 Estimating time of arrival

EKF, together with the discrete system model, allows the estimation of the hidden states of the system. For our case, they will be the variables of distance (between the pallet and the stop station) and the rate at which this changes. Estimating the time at which the manipulator robot must act only with distance information may be appropriate under the studied environment which considers a constant movement speed. However, alterations on this parameter would trigger an erroneous reaction of the system. Therefore, it is necessary to determine the instant in which the manipulator operates by considering both the distance and speed of the pallet estimations.

Estimates of distance and speed can be used together to determine the time a pallet will take to get to the target stop station. We name this estimate *time-of-arrival*.

The preceding variable modeling is made from both experimental data and data obtained through the pallet's path simulation.

By Matlab's surface fitting toolbox we obtain the surface that best fits the available data. This is a 2-order polynomial defined as:

$$T_{arrival}(tk) = p_{00} + p_{10}.d(tk) + p_{01}.v(tk) + p_{20}.d^2(tk) + p_{11}.d(tk).v(tk) + p_{02}.v^2(tk) \tag{13}$$

where $T_{arrival}$ is the time-of-arrival and d and v are the pallet's distance and speed respectively. Surface parameters p_{xx} are obtained for each of the wireless modules used in the pallets.

3.4 System controller

The instant at which the manipulator robot must act is determined by the time of arrival of the target pallet. The operation of the robot must be done in the instant in which the time the robot employs in executing the approaching routine equals the time of the pallet's arrival. To achieve the above consideration two controllers were implemented.

The first control is an open-loop PI. This was chosen due to the lack of feedback on the system. Its implementation is performed from the discretization of the PI controller given by $y(tk) = K_c\,(e(tk) - e(tk-1)) + \frac{K_c}{T_i}T_s e(tk) + y(tk-1)$, where $y(k)$ is the output and $e(k)$ is the calculated error between the controller input and a given reference for a time k, and K_p and K_i are the proportional and integral gains, respectively. The implementation is recursively performed using as a reference the time of the robot operation (T_{OP_Robot}) and the time of arrival of the pallet as input ($T_{Arrival}$). Thus the controller was modeled as follows:

$$e(tk) = T_{OP\_Robot} - T_{Arrival} \quad ; \quad a(k) = K_p.e(tk) \quad ; \quad b(k) = K_i.e(k) \tag{14}$$

$$y(k) = a(k) - a(k-1) + b(k) + y(k-1) \tag{15}$$

$$a(k-1) = a(k) \tag{16}$$

$$y(k-1) = y(k) \tag{17}$$

During algorithm's operation, Equation 15 represents an update of the controller output. Equations 16 and 17 are updates of the variables.

The second implemented control was a Fuzzy logic controller that is based on the fuzzy sets theory developed by Lotfi A. Zadeh [17]. This kind of logic intends to represent human logical thinking allowing a system to work with information that is not accurate (e.g., input sentence "the pallet is very close"). The truth degree of a variable x to a "linguistic variable"(e.g., "The pallet is far") is determined by a membership function that takes truth values in the 0 to 1 range. The linguistic variable or label is associated with a fuzzy set which can be defined as: $A = \{x, u_A(x) | x \in Ut\}$, were $u_A(x)$ is the membership function of x and U is the universe of discourse or the exact values that can take the variable [18]. The membership function can

adopt many geometries, such as triangular, trapezoidal, etc. The components of the controller are shown in Figure 4(a), where the *Fuzzification* stage converts crisp inputs or real-world values to a membership degree of a linguistic variable. Later, this variable is evaluated by a set of rules of type: ***IF*** *{set of conditions to meet}* ***Then*** *{set of consequences}*. Finally, the obtained fuzzy result must undergo a reverse process or *Defuzzification* in order to obtain a crisp output or an output with useful values for the process, again, using a membership function for the output values.

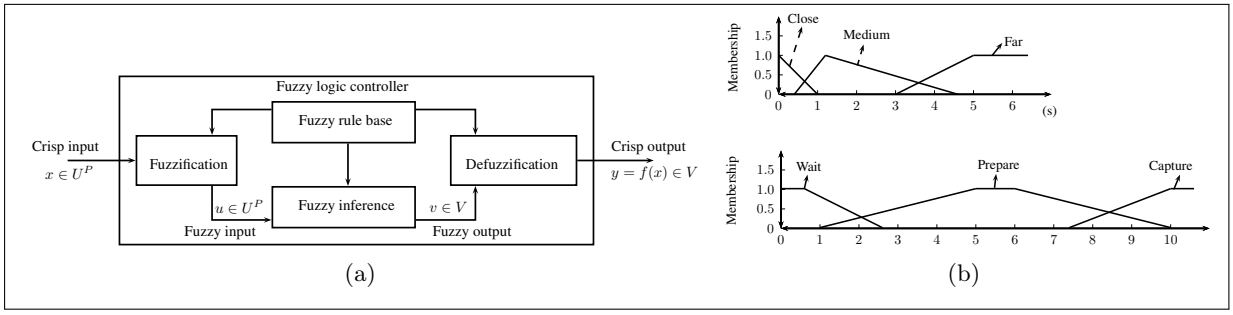


Figure 4: (a) Components of a fuzzy logic controller and (b) the implemented fuzzification (top) and defuzzification (bottom) membership functions.

As in the PI controller, for the fuzzy controller design only one input was considered. This input is the tracking error $e(tk)$ described above. An input range for this variable from 0 to 10 [sec] is established. Besides this, three linguistic variables are defined within the established universe of discourse, "*Far*", "*Medium*"and "*Close*". On the other hand, for the Deffuzification stage, three types of linguistic variables are defined: "*Wait*", "*Prepare*"and "*Capture*". The set universe of discourse corresponds to the range 0-12. Figure 4(b) shows the implemented membership functions.

For the fuzzy inference stage the following set of rules are implemented: (1) **If** $e(tk)$ **is** "*Far*"Then Output **is** "*Wait*", (2) **If** $e(tk)$ **is** "*Medium*"Then Output **is** "*Prepare*", and (3) **If** $e(tk)$ **is** "*Close*"Then Output **is** "*Capture*".

Different controller parameters have been selected from our experience operating the system. The adjustment of membership functions was performed by various assays in the system.

4 Implementation

Pallets moving on the conveyor belt transport wireless beacon nodes periodically sending data packets to a collector node located near the stop station. The collector is directly connected to a Personal Computer (PC) executing the control software. Three types of wireless nodes corresponding to the Digi's XBee family [19] were used in this work. They are: a XBee Chip Antenna, a XBee Wire Antenna, and a XBee Pro Wire Antenna, all from the product's 1 Series. These components are based on the IEEE 802.15.4, and are mainly differentiated by hardware features, such as type of antenna and supported power outputs. In our experiments, the three different XBee modules were tested as beacon nodes. A XBee Chip Antenna module was used as the collector. Nodes main parameters are detailed in Table 2. The rest of the parameters are set to be default parameters recommended by the manufacturer.

Both the robotic arm and the conveyor system are controlled by the ACS software via serial ports. These elements only execute predetermined tasks. The robot has its own operating system. This manages its resources and provides a programming interface for which the user has a set of available functions and operations to work with the robot. Three routines were

Table 2: XBee modules configuration.

	Collector node	Beacon nodes
Device	XBee Chip Antenna	XBee Wire/Chip Antenna - XBee Pro
IEEE 802.15.4 Role	Coordinator	End device
Interface speed	57600[bps]	
Sampling rate	-	54[ms]
Operation mode	API	
Power output	0[dbm]	0[dbm]-16[dbm]
Power source	USB	Batteries

programmed in its controller:
(1) **Standby:** places the robot in one of four positions previously recorded.
(2) **Approaching target:** moves the robot from any position to a position near the point of capture of the pallet (over the stop station).
(3) **Target capture:** allows the robot handling the template carrying the target pallet.

The conveyor system is controlled by a PLC with a resident program. This executes the pallet control services at the stop stations (stop, release, or identify a pallet over a stop station). The services are provided via serial communication software to the ACS. This must extract the relevant information from the data frames sent by the PLC and to generate the necessary commands to perform the required actions.

On the PC, the core of the ACS is a Matlab-based software implementing the filtering and control routines, and serial i/o and user interfacing. The ACS software aims to estimate the proximity of the objects of interest (the pallets) and to determine the instant in which the system must generate the necessary commands to anticipate the arrival of the pallet, using as input only RSSI traces. The software components are summarized in Figure 5a.

The operations sequence cyclically executes the following steps: (1) devices reading, (2) estimation algorithm execution, and (3) device status update. This sequence is repeated until achieving the target pallet capture. The operating cycle of the program is initiated by a user requirement. The software starts the cycle by reading the RSSI value and the transmitting node id, from the assigned serial port (these data are delivered by a C++-based program who actually reads and decodes packets from the XBee collector). Then, the data are filtered before being processed by the estimation algorithm. Once the arriving time is estimated, its value is entered to the system controller. If the estimated time does not equal the reference time, the current system's output state remains and the cycle is repeated. On the contrary, the controller gives the order to start the operations of approaching and capturing of the target pallet.

To summarize, Figure 5b illustrates the implemented algorithm stages.

5 Tests and Results

At the begging of the system's integration stage, parameters are adjusted in order to achieve a proper system's operation. System calibration is performed by contrasting experimentally obtained information and simulations of system's models. For obtaining real data from the system, we registered RSSI values for each node considering different positions of the manipulator robot (as discussed in previous section). With these data we simulate the behavior of the estimation and control algorithm under adjusted system parameters. These parameters are: Reference power $PL(d_0)$, Path loss exponent γ, Standard deviation of the process disturbance σ_{pp}, and Standard

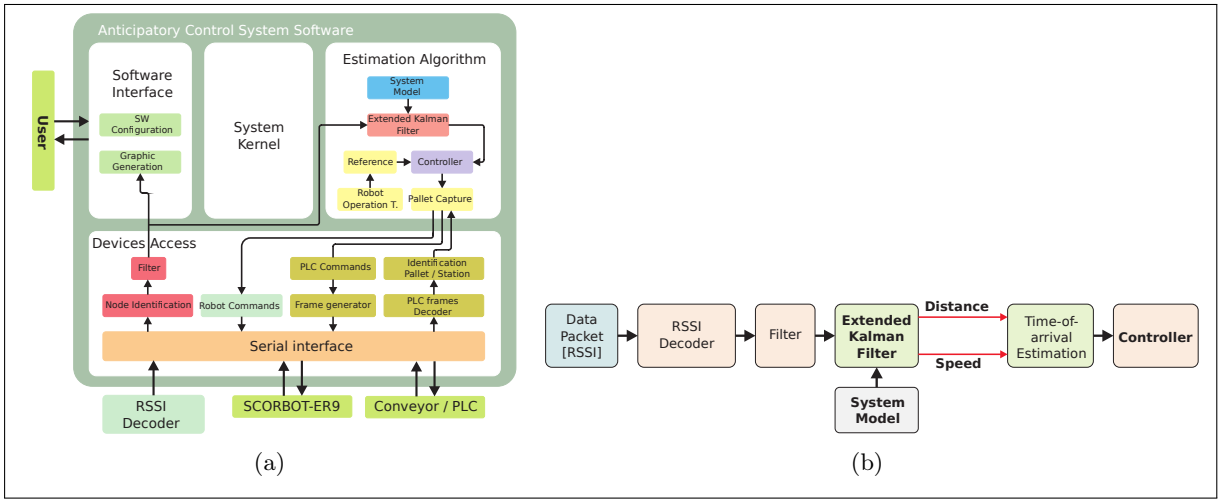


Figure 5: ACS (a) software components and (b) general scheme.

deviation of measurement disturbance σ_{pm}. System parameters calibration was performed by the Greedy algorithm [20]. Table 3 shows the new parameters obtained.

Table 3: Adjusted system parameters.

Node	Pt	$\overline{Pl}(1.5[m])$	γ	σ_{pp}	σ_{pm}
XBee chip antenna	0[dBm]	59.490[dBm]	1.949	0.0034	4.090
XBee wire antenna	0[dBm]	53.838[dBm]	3.186	0.0025	4.818
XBee Pro wire antenna	16[dBm]	52.160[dBm]	2.720	0.0023	6.0

After calibration, we performed an execution test in order to verify the estimation algorithm parameters for each beacon node. During this test, pallets are not stopped at the stop stations. Figure 6(a) shows registered RSSI values for each node. Figure 6(b) and 6(c) show estimations of time-of-arrival and distance for each node.

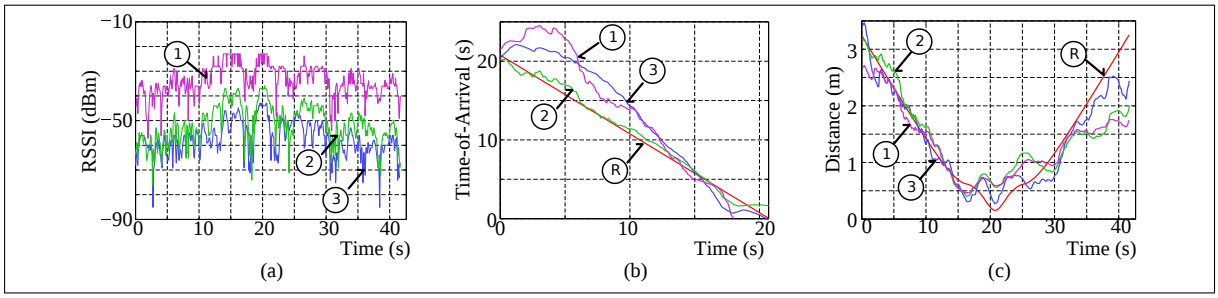


Figure 6: Estimation with calibrated parameters for ① XBee A. Chip, ② XBee A. Wire and ③ XBee Pro A. Wire. In (a), input captured data. In (b) and (c) show Ⓡ real arriving time and distance, respectively, and estimations for the three nodes ① XBee A. Chip, ② XBee A. Wire and ③ XBee Pro A. Wire.

We observe that the best distance estimate is performed on the first half of the workspace represented by the first 21 seconds in Figure 6(c). In Here, the pallet approaches the stop station reaching the shortest distance between the transmitter and the collector nodes (approximately at second 21). Then, the pallet moves away.

Now, in order to evaluate the performance of the ACS, several tests under different operating conditions were conducted. The objectives of these tests are: (1) to compare system performance for different transmitting nodes, (2) to compare performance of the system under different initializations of the error covariance matrix, (3) to measure overall system performance, and (4) to observe the effect of sampling rate variations in the system estimates. To accomplish these objectives, we performed both an individual and a global performance tests, and a test to observe the degradation of the estimation due to sampling rate variations. In each test estimates of distance, time-of-arrival and time of capture of the pallet are recorded. This last value is defined as the time between the arrival of the pallet to the stop station and the time the robot takes to get the catching position. A good result for the capture time is a near zero (seconds) value. The performance metrics of the system are: the Distance Absolute Error Estimation, the Time-of-Arrival Absolute Error Estimation, and the Absolute Capture Time Absolute Error.

In the individual performance test, we compare ACS performance considering the use of different beacon nodes. Firstly, individual beacons are located on a template, then we let them move over the conveyor. Individual tests are performed with the three types of XBee nodes and different initial positions of the robotic arm. In each case, ACS executes a full cycle allowing the robot to capture the template. Along with this, we want to observe the effect of the initialization of the error covariance matrix (P). For this, we executed two versions of the estimation algorithm: (Version A) the original version of the algorithm, initializing the error covariance matrix to a near zero value, modifying its value as tests succeed, and (Version B) a modified algorithm, where the error covariance matrix is initialized with a default value. This value is obtained after successive runs of the original algorithm. During each execution of the experiment the value of the array is initialized to this value. Average results for these tests are shown in Figure 7. In the global

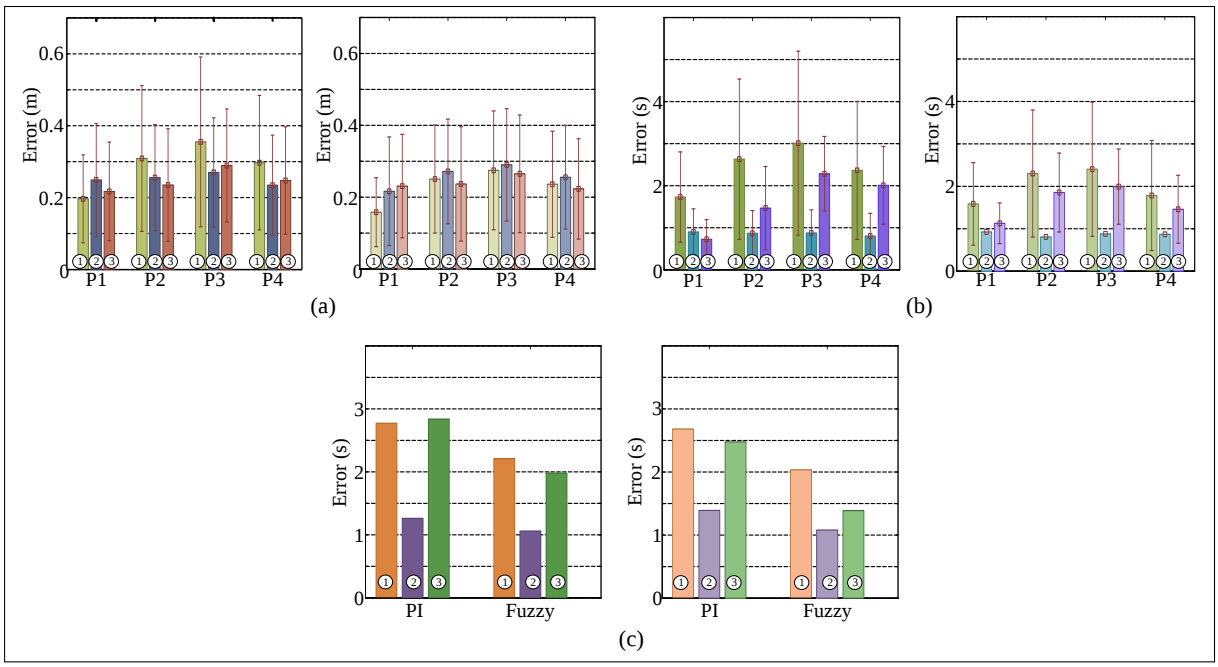


Figure 7: Average results for the individual performance tests with the tree wireless nodes ① XBee Antenna Chip, ② XBee Antenna Wire and ③ XBee Pro Antenna Wire. In (a), Absolute Error of the Estimated Distance with both versions of the filter A (left) and B (right). In (a), Absolute Error of the Estimated Time-of-Arrival with both versions of the filter A (left) and B (right). In (c), Absolute Error of the Capturing Time with both versions of the filter A (left) and B (right).

performance test, we observe the performance of the system under normal operating conditions.

This involves capturing any of the three available pallets on the system from a random position as the starting point for the robotic arm. For this test, we use the three adopted types of nodes as beacons transmitting simultaneously. We compare the performance of each node individually with the performance of the system to a random selection of the target node. The selection of the initial position of the robot is performed at random for each test. As in the previous test, we observe the effect of the P matrix initialization. For each configuration of the experiment 10 tests are performed in a row. Average results for these tests are shown in Figure 8. Finally, we study

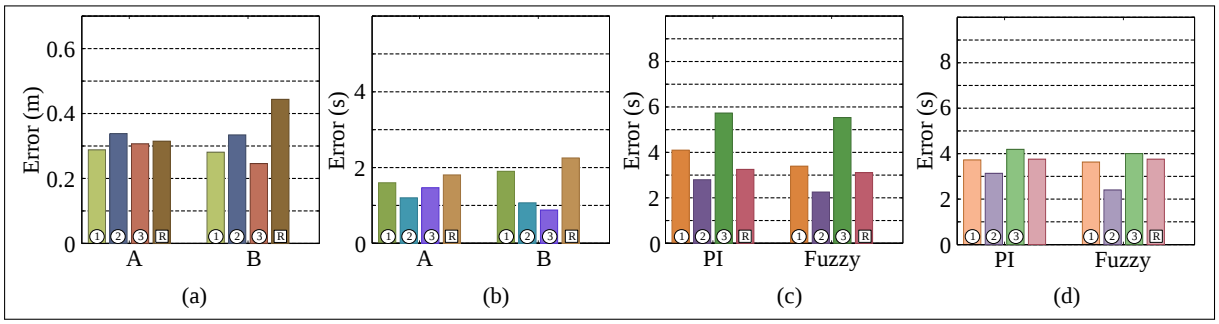


Figure 8: Average results for the global performance tests for the three adopted nodes ① XBee Antenna Chip, ② XBee Antenna Wire and ③ XBee Pro Antenna Wire, and a R random selection for the target node. In (a) and (b), Absolute error of the estimated distance and of the estimated time-of-arrival for both of the filter versions A and B. In (c) and (d), the absolute error of the capturing time for both filter version A and B, respectively.

the degradation of the estimation produced by reductions on the sampling rate. This test is performed by using the same Beacon node (a XBee Chip Antenna) while the manipulator robot adopts a same initial position for each of the experiments. The experiments consider increases 100[ms] in the given sampling rate from an initial value of 100[ms] and a final test value of 1100[ms]. Average results for these tests are shown in Figure 9(a), 9(b), and 9(c).

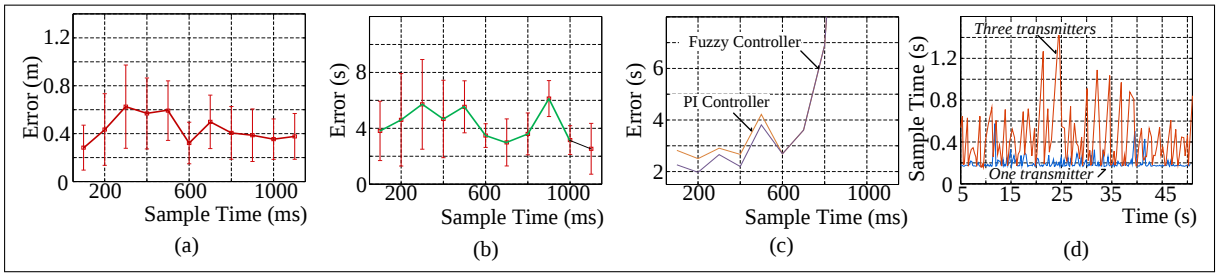


Figure 9: In (a) and (b), the absolute error of the estimated distance and time-of-arrival, respectively. In (c), the absolute capture time shows the absolute error of the capturing time with respect to the sampling time. In (d) the effect on the sampling time of increasing the number of transmitting nodes.

From the results, we can observe a system performance decline when including a larger number of simultaneously operating transmitter nodes (see Figures 7(a) and 8(a)). Note that the operation mode of the transmitting nodes causes dispute over the transmission medium. The information received from these nodes is at random, without an order in the delivery. This causes instability and decrease in the RSSI sample rate, resulting in an increased error in the estimation of distance which grows in a 18.3 %. Figure 9(d) illustrates this effect. The figure shows the sampling rate for XBee Chip Antenna nodes operating independently and the rate obtained in a scenario with 3 nodes transmitting simultaneously.

Regarding the performance of the system, this is maintained within acceptable parameters by using only one node Beacon, observing errors in distance estimation and in the time-of-arrival of

25[cm] and 1.5[s] around, respectively. This allows the system to meet the design requirements by obtaining a capture time of less than 3[s] (2[s] average).

The last performed test allows us to determine the maximum sampling rate supported by the system. This is 600[ms] for the case of an individual node. In case of using a larger number of nodes this rate should allow increasing the number of nodes without significantly affecting the system performance. However, a way to synchronize the senders to the collector node is necessary. A sampling time greater than the determined one can cause the system not to properly determine the area of operation of the controller. This affects the pallet capturing algorithm, causing it not to stop in a timely manner so the pallet flows without stopping at the corresponding stop station and must wait for a new approach to perform the capture. This is observed in Figure 9(c) in which higher values to the determined one produces excessive time increases in pallet capture. Moreover, results show that there is a considerable improvement in the obtained results when initializing the error covariance matrix to a preset value (Version B of the algorithm). In general, we observe a decrease in errors estimates close to 8%. In the case of capture time we observe an improvement of 10% by using the version B of the algorithm. Regarding the performance of the controllers employed in the system, better results are observed when using the fuzzy controller in all of the implemented tests. The fuzzy controller has reduced capture times compared to PI controller by 27% in the individual performance test and 10% in the overall performance test.

Finally, we observe the performance of the system considering the three employed nodes. From these, the best results, considering the best estimate of time-of-arrival and time of capture of the pallet, are obtained by applying the XBee Wire Antenna module. This is observed in Figures 7(b) and 7(c).

6 Conclusions and Future Works

This paper presents the development and implementation of a proposed control scheme which is able to trigger an anticipatory control action. For the above, RSSI measurement from a wireless sensor mounted in a moving pallet are used along with an EFK in order to estimate the speed and distance of the moving pallet relative to a stop station.

The control scheme is proposed for short distance in order to avoid many factors which affect the signals behavior (bouncing, multipath, etc.). Specifically for a 2.25 m radius case the control scheme is able to estimate the time of arrival of the pallet traveling at 0.19 m/s and to generate the necessary control action (for the case presented the movements of a robotic arm) before it reaches the stop point. Due to the lack of a feedback loop for the robotic arm used, a PI and Fuzzy controller are implemented, obtaining a better response by the fuzzy strategy with a 54[ms] sampling time.

Due to RSSI instabilities it is necessary to avoid significant changes in the environment in order to estimate correctly speed and position of the pallet. Future work will focus in the RSSI instabilities compensation.

Acknowledgment

This work was partially supported by the Research Department of the University of Bío-Bío through the project DIUBB 121910 2/R and Conicyt (Fondecyt Project No. 11121657).

Bibliography

[1] I.F. Akyildiz,W. Su, Y. Sankarasubramaniam, E. Cayirci (2002); Wireless Sensor Networks: A Survey, *Computer Networks*, 38(4), 393-422.

[2] K.S. Low,W.N.N. Win, M.J. Er (2005); Wireless Sensor Networks for Industrial Environments, Computational Intelligence for Modelling, Control and Automation, *2005 and Int. Conf. on Intelligent Agents, Web Technologies and Internet Commerce*, 2: 271-276.

[3] V.C. Gungor, G.P. Hancke (2011); Industrial Wireless Sensor Networks, In *Industrial Electronics Handbook*, 2nd ed., CRC Press and IEEE Press.

[4] L. Krishnamurthy et al. (2005); Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the North Sea, in *3rd Int. Conf. on Embedded Networked Sensor Systems*, 64-75.

[5] J. McCulloch, P. McCarthy, S.M. Guru, W. Peng, D. Hugo, A. Terhorst (2008); Wireless sensor network deployment for water use efficiency in irrigation, in *REALWSN'08: Proc. of the workshop on Real-world wireless sensor networks*, ACM, 46-50.

[6] Z. Butler, P. Corke, R. Peterson, D. Rus (2006); From robots to animals: Virtual fences for controlling cattle, *International Journal of Robotics Research*, 25(5-6): 485-508.

[7] A. Zouinkhi, A. Ltifi, M. Ben Gayed, N. Abdelkrim, E. Bajic, E. Rondeau (2010); Simulation of active products cooperation for active security management, *8th Conf. Int. de Modélisation et Simulation (MOSIM'10)*, Hammamet, Tunisie.

[8] D. J. Cook, J. C. Augusto, V. R. Jakkula (2009); Review: Ambient intelligence: Technologies, applications, and opportunities, *Pervasive Mob. Comput.*, DOI=10.1016/j.pmcj.2009.04.001, 5(4): 277-298.

[9] D. Dobre, G. Morel, D. Gouyon (2010); Improving human-system digital interaction for industrial system control: some systems engineering issues, In: *10th IFAC Workshop on Intelligent Manufacturing Systems, IMS'10, Lisbone : Portugal*, 1-7.

[10] S. E. Taylor, T. P. McDonald, M. W. Veal, T. E. Grift (2001); Using GPS to evaluate productivity and performance of forest machine systems, *First Int. Precision Forestry Symposium*, 1-20.

[11] Ubisense. `http://www.ubisense.net/`

[12] C. Duran-Faundez, C. Aguilera-Carrasco, A. S. Norambuena, (2010); Experimenting with RSSI for the perception of moving units in intelligent flexible manufacturing systems, *2010 IEEE Int. Conf. on Industrial Technology, ICIT2010*, 1400-1405.

[13] Laboratorio de Sistemas Automatizados de Producción CIMUBB, Universidad del Bío-Bío, `http://www.ubiobio.cl/cimubb/`

[14] L. Ljung *System Identification Theory for the User*. Prentice Hall PTR. 1999.

[15] G. Welch, G. Bichop (2006); *An introduction to the Kalman Filter. Reference Material*, Univ. of North Carolina at Chapel Hill, Dept. of Computer Science, 1-16.

[16] R. Grover Brown, P.Y. C. Hwang (1997); *Introduction to random signals and applied Kalman Filtering with Matlab: exercises and solution*, 3rd ed.. USA: John Wiley & Sons.

[17] L.A. Zadeh (1965); Fuzzy sets, *Information and Control*, 8 (3): 338-353.

[18] A. M. Ibrahim (1997); *Introduction to Applied Fuzzy Electronics*, Prentice Hall.

[19] Digi International `http://www.digi.com/`

[20] T. H. Cormen, C. E. Leiserson, R.L. Rivest, C. Stein (2009); *Introduction to Algorithms*, (3a. Ed.) Massachusetts: Massachusetts Institute of Technology.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):718-731, October, 2015.

Verification of JADE Agents Using ATL Model Checking

L.F. Stoica, F. Stoica, F.M. Boian

Laura Florentina Stoica*, Florin Stoica
1. Department of Mathematics and Informatics,
Faculty of Science, "Lucian Blaga" University,
5-7 Dr. Ratiu Street, Sibiu, Romania
2. Center of Scientific Research in Informatics and Information Technology
"Lucian Blaga" University
5-7 Dr. Ratiu Street, Sibiu, Romania
laura.cacovean@ulbsibiu.ro, florin.stoica@ulbsibiu.ro
* Corresponding author: laura.cacovean@ulbsibiu.ro

Florian Mircea Boian
Department of Computer Science,
Faculty of Mathematics and Computer Science, "Babes-Bolyai" University,
1 M. Kogalniceanu Street, Cluj Napoca, Romania,
florin@cs.ubbcluj.ro

Abstract:
It is widely accepted that the key to successfully developing a system is to produce a thorough system specification and design. This task requires an appropriate formal method and a suitable tool to determine whether or not an implementation conforms to the specifications. In this paper we present an advanced technique to analyse, design and debug JADE software agents, using Alternating-time Temporal Logic (ATL) which is interpreted over concurrent game structures, considered as natural models for compositions of open systems. In development of the proposed solution, we will use our original ATL model checker. In contrast to previous approaches, our tool permits an interactive or programmatic design of the ATL models as state-transition graphs, and is based on client/server architecture: ATL Designer, the client tool, allows an interactive construction of the concurrent game structures as a directed multi-graphs and the ATL Checker, the core of our tool, represents the server part and is published as Web service.
Keywords: model checking, ATL, agents, JADE, FSM

1 Introduction

Because of competition on the information technology market, the development of the information systems must be achieved with maximum productivity. In many cases, the price paid is the diminution of quality of software products.

The aim of the methods of software engineering is to increase the quality and reliability of software. In particular, the formal methods are focused on reliability and correctness, using a mathematical support.

The reliability of a system can be increased through modelling of the system before the implementation of code, followed by the validation and verification of its correctness.

Validation is accomplished with respect to informal requirements of the system. The designer checks if the model reflects the expected behaviour of the system.

Verification is accomplished with respect to system specifications expressed in a formal manner. A formal specification is an abstract model of the system, expressed in a formal notation.

Verification of a software system involves checking whether the system in question behaves as it was designed to behave. Design validation involves checking whether a system design

Copyright © 2006-2015 by CCC Publications

satisfies the system requirements. Both of these tasks, system verification and design validation can be accomplished thoroughly and reliably using model-based formal methods, such as model checking [1].

Model checking is particularly well-suited for the automated verification of finite-state systems, both for software and for hardware. Main concern of formal methods in general, and model checking in particular, is helping to design correct systems [2]. Detecting and eliminating bugs as early in the design cycle as possible is clearly an economic imperative. For example, the Pentium FDIV bug (a bug in the Intel P5 Pentium floating point unit discovered in 1994) cost Intel Corporation a half billion dollars.

Model checking is a technology widely used for the automated system verification and represents a technique for verifying that finite state systems satisfy specifications expressed in the language of temporal logics.

Alur et al. introduced Alternating-time Temporal Logic (ATL), a more general variety of temporal logic, suitable for specifying requirements of multi-agent systems [3]. ATL is also widely used to reason about strategies in multiplayer games. The semantics of ATL is formalized by defining games such that the satisfaction of an ATL formula corresponds to the existence of a winning strategy.

The model checking problem for ATL is to determine whether a given model satisfies a given ATL formula.

ATL defines "cooperation modalities", of the form $\langle\langle \mathcal{A} \rangle\rangle \varphi$, where $\mathcal{A}$ is a group of agents. The intended interpretation of the ATL formula $\langle\langle \mathcal{A} \rangle\rangle \varphi$ is that the agents $\mathcal{A}$ can cooperate to ensure that φ holds (equivalently, that $\mathcal{A}$ have a winning strategy for φ) [4].

ATL has been implemented in several symbolic tools for the analysis of open systems. In [5] is presented a verification environment called MOCHA for the modular verification of heterogeneous systems.

The input language of MOCHA is a machine readable variant of reactive modules. Reactive modules provide a semantic glue that allows the formal embedding and interaction of components with different characteristics [5].

In [6] is described MCMAS, a symbolic model checker specifically tailored to agent-based specifications and scenarios. MCMAS has been used in a variety of scenarios including web-services, diagnosis, and security. MCMAS takes a dedicated programming language called ISPL (Interpreted Systems Programming Language) as model input language. An ISPL file fully describes a multi-agent system (both the agents and the environment) [6].

Two most common methods of performing model checking are explicit enumeration of states of the model and respectively the use of symbolic methods.

Symbolic model checkers analyse the state space symbolically using Ordered Binary Decision Diagrams (OBDDs), which are data structures for representing Boolean functions and were introduced in [12].

In [13], [14], [15], [16] are presented comparisons between symbolic and explicit model checking of software or hardware systems. For most hardware designs which are based on a clocked-approach and thus are synchronous, the symbolic model checking approach is more appropriate [14]. The explicit-state model-checkers performs better in case of using large states needed to include some information [13], also for nondeterministic, high-level models of hardware protocols and for model checking of concurrent asynchronous software systems [15], [16].

In [7] is presented our tool, a new interactive ATL model checker environment based on algebraic approach. The original implementation of the model checking algorithm is based on Relational Algebra expressions translated into SQL queries. The broad goal of our research was to develop a reliable, easy to maintain, scalable model checker tool to improve applicability of ATL model checking in design of general-purpose computer software.

Taking into account the above considerations, in our tool we are using an explicit-state model technique. Thus, in contrast to previous approaches, our tool is using oriented multi-graphs to represent concurrent game structures over which is interpreted the ATL specification language. The core of our ATL model checker is the ATL compiler which translates a formula φ of a given ATL model to set of nodes over which formula φ is satisfied. The implementation of the model checking algorithm is based on Java code generated by ANTLR (Another Tool for Language Recognition) using an original ATL grammar and provides error-handling for eventual lexical/syntax errors in formula to be analysed. We found that our ATL model checker tool scale well, and can handle even very large problem sizes efficiently, mainly because it is based on a client/server architecture and take advantage of a high performance database server for implementation of the ATL model checker algorithm.

In this paper, using components of our tool, we will show how ATL model checking technology can be used for automated verification of multi-agent systems, developed with JADE.

One of the main drawbacks of employing ATL logic in the automated verification of multi-agent systems using previous approaches consists in necessity of translate the programs written in specific modelling languages to the programming language used in the real implementation.

Our approach eliminates this problem by allowing a transparent building of the ATL model at runtime, using the native language of JADE agents (Java).

Using Java version of the ATL Library - a component of our ATL model checker, used for development of custom applications with large ATL models - into JADE agents is inserted the code necessary to build in a transparent manner the ATL model which will be verified at runtime.

The paper is organized as follows. In section 2 we present the definition of the concurrent game structure, the ATL syntax and the ATL semantics. In section 3 is outlined the architecture of our ATL model checker and is made a brief analysis of its performance. In section 4 is presented the JADE FSMBehaviour and formal models used to build an equivalent concurrent game structure. These concepts are applied in section 5 where ATL Library is used to verify the design of the JADE agents having FSM - driven behaviours. Conclusions are presented in section 6.

2 Alternating-Time Temporal Logic

The ATL logic was designed for specifying requirements of open systems. An open system interacts with its environment and its behaviour depends on the state of the system as well as the behaviour of the environment. In the following we will describe a computational model appropriate to describe compositions of open systems, called concurrent game structure (CGS).

2.1 The concurrent game structure

A *concurrent game structure* is defined as a tuple $S = \langle \Lambda, Q, \Gamma, \gamma, M, d, \delta \rangle$ with the following components:

- a nonempty finite set of all agents $\Lambda = \{1, ..., k\}$;
- Q denotes the finite *set of states* ;
- Γ denotes the finite *set of propositions* (or *observables*);
- $\gamma : Q \to 2^{\Gamma}$ is called the *labelling* (or *observation*) *function*, defined as follows: for each state $q \in Q$, $\gamma(q) \subseteq \Gamma$ is the set of propositions *true* at q;
- M represents a nonempty finite *set of moves*;

- the *alternative moves function* $d : \Lambda \times Q \to 2^M$ associates for each player $a \in \{1, ..., k\}$ and each state $q \in Q$ the set of available moves of agent a at state q. In the following, the set $d(a, q)$ will be denoted by $d_a(q)$. For each state $q \in Q$, a tuple $\langle j_1, ..., j_k \rangle$ such that $j_a \in d_a(q)$ for each player $a \in \Lambda$, represents a *move vector* at q.

- the *transition function* $\delta(q, \langle j_1, ..., j_k \rangle)$, associates to each state $q \in Q$ and each move vector $\langle j_1, ..., j_k \rangle$ at q the state that results from state q if every player $a \in \{1, ..., k\}$ chooses move j_a.

A *computation* of S is an infinite sequence $\lambda = q_0, q_1, ...$ such that q_{i+1} is the successor of q_i, $\forall i \geq 0$. A *q-computation* is a computation starting at state q. For a computation λ and a position $i \geq 0$, we denote by $\lambda[i]$, $\lambda[0, i]$, and $\lambda[i, \infty]$ the i-th state of λ, the finite prefix $q_0, q_1, ..., q_i$ of λ, and the infinite suffix $q_i, q_{i+1}, ...$ of λ, respectively [3].

2.2 Syntax of ATL

The ATL operator $\langle\langle\rangle\rangle$ is *a path quantifier*, parameterized by sets of agents from Λ. The operators $\bigcirc$ ('next'), $\Box$ ('always'), $\Diamond$ ('future') and U ('until') are *temporal operators*. A formula $\langle\langle\mathcal{A}\rangle\rangle\ \varphi$ expresses that the team $\mathcal{A}$ has a collective strategy to enforce φ.

The temporal logic ATL is defined with respect to a finite set of agents Λ and a finite set Γ of propositions. An ATL formula has one of the following forms:

1. p, where $p \in \Gamma$;
2. $\neg\varphi$ or $\varphi_1 \vee \varphi_2$ where φ, φ_1 and φ_2 are ATL formulas;
3. $\langle\langle\mathcal{A}\rangle\rangle \bigcirc \varphi$, $\langle\langle\mathcal{A}\rangle\rangle\Box\varphi$, $\langle\langle\mathcal{A}\rangle\rangle\Diamond\varphi$ or $\langle\langle\mathcal{A}\rangle\rangle\varphi_1 U \varphi_2$, where $\mathcal{A} \subseteq \Lambda$ is a set of players, and φ, φ_1 and φ_2 are ATL formulas.

Other boolean operators can be defined from $\neg$ and $\vee$ in the usual way. The ATL formula $\langle\langle\mathcal{A}\rangle\rangle\Diamond\varphi$ is equivalent with $\langle\langle\mathcal{A}\rangle\rangle\ true\ U\ \varphi$.

2.3 Semantics of ATL

Consider a game structure $S = \langle \Lambda, Q, \Gamma, \gamma, M, d, \delta \rangle$ with $\Lambda = \{1, ..., k\}$ the set of players. We denote by

$$D_a = \bigcup_{q \in Q} d_a(q) \tag{1}$$

the set of available moves of agent a within the game structure S.

A *strategy* for player $a \in \Lambda$ is a function $f_a : Q^+ \to D_a$ that maps every nonempty finite state sequence $\lambda = q_0 q_1 ... q_n$, $n \geq 0$, to a move of agent a denoted by $f_a(\lambda) \in D_a \subseteq M$. Thus, the strategy f_a determines for every finite prefix λ of a computation a move $f_a(\lambda)$ for player a in the last state of λ.

Given a set $\mathcal{A} \subseteq \{1, ..., k\}$ of players, the set of all strategies of agents from $\mathcal{A}$ is denoted by $\mathcal{F}_\mathcal{A} = \{f_a | a \in \mathcal{A}\}$. The *outcome* of $\mathcal{F}_\mathcal{A}$ is defined as $out_{\mathcal{F}_\mathcal{A}} : Q \to \mathcal{P}(Q^+)$, where $out_{\mathcal{F}_\mathcal{A}}(q)$ represents *q-computations* that the players from $\mathcal{A}$ are enforcing when they follow the strategies from $\mathcal{F}_\mathcal{A}$. In the following, for $out_{\mathcal{F}_\mathcal{A}}(q)$ we will use the notation $out(q, \mathcal{F}_\mathcal{A})$. A computation $\lambda = q_0, q_1, q_2, ...$ is in $out(q, \mathcal{F}_\mathcal{A})$ if $q_0 = q$ and for all positions $i \geq 0$, there is a move vector $\langle j_1, ..., j_k \rangle$ at state q_i such that [3]:

- $j_a = f_a(\lambda[0, i])$ for all players $a \in \mathcal{A}$, and

- $\delta(q_i, j_1, ..., j_k) = q_{i+1}$.

For a game structure S, we write $q \models \varphi$ to indicate that the formula φ is satisfied in the state q of the structure S.

For each state q of S, the satisfaction relation $\models$ is defined inductively as follows:

- for $p \in \Gamma$, $q \models p \Leftrightarrow p \in \gamma(q)$
- $q \models \neg\varphi \Leftrightarrow q \not\models \varphi$
- $q \models \varphi_1 \vee \varphi_2 \Leftrightarrow q \models \varphi_1$ or $q \models \varphi_2$
- $q \models \langle\langle\mathcal{A}\rangle\rangle \bigcirc \varphi \Leftrightarrow$ there exists a set $\mathcal{F}_\mathcal{A}$ of strategies, such that for all computations $\lambda \in out(q, \mathcal{F}_\mathcal{A})$, we have $\lambda[1] \models \varphi$ (the formula φ is satisfied in the successor of q within computation λ).
- $q \models \langle\langle\mathcal{A}\rangle\rangle\Box\varphi \Leftrightarrow$ there exists a set $\mathcal{F}_\mathcal{A}$ of strategies, such that for all computations $\lambda \in out(q, \mathcal{F}_\mathcal{A})$, and all positions $i \geq 0$, we have $\lambda[i] \models \varphi$ (the formula φ is satisfied in all states of computation λ).
- $q \models \langle\langle\mathcal{A}\rangle\rangle\varphi_1 U \varphi_2 \Leftrightarrow$ there exists a set $\mathcal{F}_\mathcal{A}$ of strategies, such that for all computations $\lambda \in out(q, \mathcal{F}_\mathcal{A})$, there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$, we have $\lambda[j] \models \varphi_1$.
- $q \models \langle\langle\mathcal{A}\rangle\rangle\Diamond\varphi \Leftrightarrow$ there exists a set $\mathcal{F}_\mathcal{A}$ of strategies, such that for all computations $\lambda \in out(q, \mathcal{F}_\mathcal{A})$, there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi$.

3 Architecture, scalability and performance of the ATL model checker

Our ATL model checker tool contains the following packages:

- *ATL Compiler* - the core of our tool, embedded into a Web Service (ATL Checker);
- *ATL Designer* - the GUI client application used for interactive construction of the ATL models as directed multi-graphs;
- *ATL Library* - used for development of custom applications with large ATL models. Versions of this library are provided for *C#* and *Java*.

The software can be downloaded from http://use-it.ro (binaries and examples of use).

ATL Designer [9] implements the Tic-Tac-Toe (TTT) game, using an algorithm which looks for infallible conditional plans to achieve a winning strategy that can be defined via ATL formulae. The game is played by two opponents with a turn-based modality on a 3×3 board. The two players take turns to put pieces on the board. A single piece is put for each turn and a piece once put does not move. A player wins the game by first lining three of his or her pieces in a straight line, no matter horizontal, vertical or diagonal. A user can play TTT game against the computer and the ATL model checking algorithm is used to achieve a winning strategy for the computer.

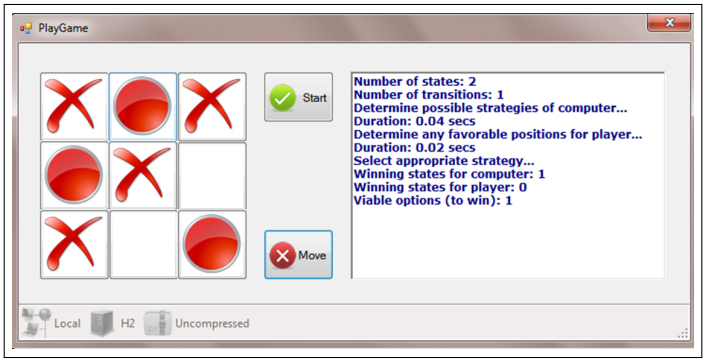


Figure 1: The computer has found a winning strategy and won a game against the human user

In the following we evaluate the effectiveness of our approach in designing and implementing an ATL model checker and we report some experimental results.

For a set $\mathcal{A}$ of agents, the implementation of most ATL operators in the model checking algorithm [3] implies the computation of function $Pre(\mathcal{A}, \Theta)$, where $\Theta \subseteq Q$. The value returned by $Pre(\mathcal{A}, \Theta)$ represents the set of states from which agents $\mathcal{A}$ can enforce the system into some state in Θ in one move.

In [7] we made a implementation of the function $Pre()$ using SQL statements, ready to be executed on a high-speed database server. Our approach is to use SQL and its massive scalability features in verification of large real-world systems.

In order to analyze their impact in the performance of the ATL model checker, were used three different database servers necessary to determine the winning strategy in the Tic-Tac-Toe game, namely *MySql 5.5*, *H2 1.3* and respectively *Microsoft SQL Server 2008* using an *Intel Core I5*, *2.5 GHz*, *4Gb RAM*.

Benchmark results are presented in figure 2. Results from [18] showed that both SMV (a symbolic CTL model checker) and $SPIN$ (a well-known explicit-state LTL model checker tool) were able to find an optimal strategy for a player in less than one second, on a 3×3 board. As we can see from figure 2, our ATL model checker tool is not as fast as the CTL/LTL tools, but we must take into consideration that an ATL model is more expressive (with ATL we can quantify over the individual powers of one player or a cooperating team of players, ATL models capture various notions of synchronous and asynchronous interaction between open systems, etc.). But our tool achieves substantial speedup over an implementation of the Tic-Tac-Toe in the Reactive Modules Language (RML) of the ATL model checker MOCHA [17], [7].

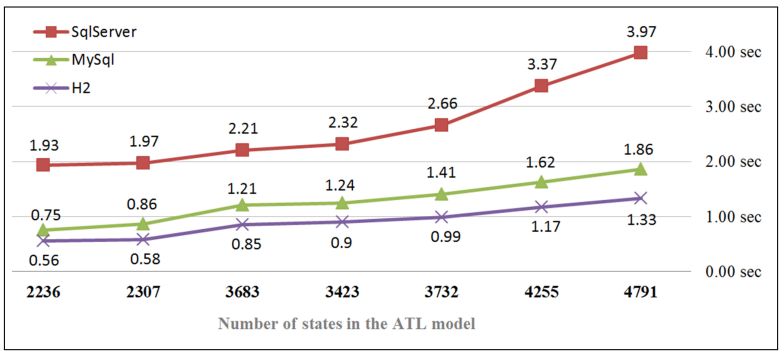


Figure 2: The performance of ATL model checker related to database server used

4 Using ATL model checking in agent-based systems

In the following we will show how our tool can be used for applying the ATL technology in the field of agent-based applications.

The domain of software agents being relatively recent and in a continuous development, there is no a standard definition, unanimous accepted definition for an agent.

However, the autonomy is the central property of agent concept, others properties having a different importance, regarding of concrete considered applications. We will consider that an agent is a computational system situated in a runtime environment and capable of autonomous actions in that environment, in order to accomplish his planned objectives.

Because ATL includes notions of agents, their abilities and strategies (conditional plans) explicitly in its models, ATL is appropriate for planning, especially in multi-agent systems [8]. ATL models generalize turn-based transition trees from game theory and thus it is not difficult to encode a game in the formalism of concurrent game structures, by imposing that only one agent makes a move at any given time step.

Automated verification of a multi-agent system by ATL model checking is the formal process through which a given specification expressed by an ATL formula and representing a desired behavioural property is verified to hold for the ATL model of that system.

In the following, ATL Library will be used to detect errors in the design, specification and implementation of an agent developed in JADE.

For the beginning we present an ATL model suited for FSM (Finite State Machine) - driven behaviour of a JADE agent. This model will help us to elaborate the mapping rules between ATL and JADE concepts. ATL Library will be used to validate the design of JADE agents having *FSM-behaviours*, in other words, to see that no incorrect scenarios arise as a consequence of a bad design.

4.1 JADE agents with FSM behaviours

JADE is a middleware that facilitates the development of multi-agent systems and applications conforming to FIPA standards for intelligent agents [10].

The Agent class represents a common base class for user defined agents. Therefore, from the programmer's point of view, a JADE agent is simply an instance of a user defined Java class that extends the base Agent class.

The computational model of an agent is multitask, where tasks (or behaviours) are executed concurrently. A scheduler, internal to the base Agent class and hidden to the programmer, automatically manages the scheduling of behaviours.

A behaviour represents a task that an agent can carry out and is implemented as an object of a class that extends the standard JADE class jade.core.behaviours.Behaviour. In order to make an agent execute the task implemented by a behaviour object it is sufficient to add the behaviour to the agent by means of the *addBehaviour()* method of the Agent class.

Each class extending Behaviour must implement the *action()* method, that actually defines the operations to be performed when the behaviour is in execution, and the *done()* method (returns a boolean value), that specifies whether or not a behaviour has completed and have to be removed from the pool of behaviours which an agent is carrying out. Scheduling of behaviours in an agent is not pre-emptive (as for Java threads) but cooperative. This means that when a behaviour is scheduled for execution its *action()* method is called and runs until it returns. The termination value of a behaviour is returned by his *onEnd()* method [11]. The path of execution of the agent thread is showed in the following pseudocode:

```
void AgentLifeCycle() {
  setup();
  while (true) {
    if (was called doDelete()) {
      takeDown();
      return;
    }
    Behaviour b =
    getNextActiveBehaviourFromSchedulingQueue();
    b. action();
    if (b.done() returns true) {
      removeBehaviourFromTheSchedulingQueue (b);
      int terminationValueOfTheBehaviour =  b.onEnd();
    }
  }
}
```

Behaviours work just like co-operative threads, but there is no stack to be saved. Therefore, the whole computation state must be maintained in instance variables of the Behaviour and its associated Agent.

Following this idiom, agent behaviours can be described as finite state machines, keeping their whole state in their instance variables. When dealing with complex agent behaviours, using explicit state variables can be cumbersome; so JADE also supports a compositional technique to build more complex behaviours out of simpler ones. The JADE abstract class CompositeBehaviour provides the possibility of combining simple behaviours together (children) to create complex behaviours. The actual operations performed by executing this behaviour are defined inside its children while the composite behaviour deals with execution planning. The scheduling policy must be defined by subclasses of CompositeBehaviour.

The FSMBehaviour is such a subclass that executes its children according to a Finite State Machine (FSM) defined by the user. More in details each child represents the activity to be performed within a state of the FSM and the user can define the transitions between the states of the FSM. When the child corresponding to state S_i completes, its termination value (as returned by the *onEnd()* method) is used to select the transition to fire and a new state S_j is reached. At next round the child corresponding to S_j will be executed. Some of the children of an FSMBehaviour can be registered as final states. The FSMBehaviour terminates after the completion of one of these children.

The following methods are needed in order to properly define a FSMBehaviour:

- `public void registerFirstState (Behaviour state, java.lang.String name)`
  Is used to register a single Behaviour *state* as the initial state of the FSM with the name *name*.

- `public void registerLastState (Behaviour state, java.lang.String name)`
  Is called to register one or more Behaviours as the final states of the FSM.

- `public void registerState(Behaviour state, java.lang.String name)`
  Register one or more Behaviours as the intermediate states of the FSM.

- `public void registerTransition (java.lang.String s1, java.lang.String s2, int event)`
  For the state $s1$ of the FSM, register the transition to the state $s2$, fired by terminating event of the state $s1$ (the value of terminating event is returned by *onEnd()* method, called when leaving the state $s1$ - sub-behaviour $s1$ has completed).

- `public void registerDefaultTransition (java.lang.String s1, java.lang.String s2)`

This method is useful in order to register a default transition from a state to another state independently on the termination event of the source state.

4.2 A formal model of the FSMBehaviour

In the following we present a model for FSM-driven behaviour of a JADE agent, implemented by FSMBehaviour class. This model will help us to elaborate the mapping rules between ATL and JADE concepts.

A JADE finite state machine is a tuple $FSM = (Q_{FSM}, \Pi, \pi, q_0, F, t, \delta_{FSM})$ where:

- Q_{FSM} is a finite, non-empty *set of states*;
- Π denotes the finite *set of state names*;
- $\pi : Q_{FSM} \to \Pi$ is called the *labelling function*, defined as follows: for each state $q \in Q_{FSM}$, $\pi(q) \in \Pi$ is the *name of state* q;
- q_0 is an element of Q_{FSM}, the *initial state*;
- $F \subseteq Q_{FSM}$ is the *set of final states*;
- $t : Q_{FSM} \to 2^{\mathbb{Z} \cup \{default\}}$ is called the *terminating function*, where for each state $q \in Q_{FSM}$, $t(q) \subseteq \mathbb{Z} \cup \{default\}$ represents the set of admissible termination codes of the state q;
- The transition function $\delta_{FSM}(q, j)$, associates to each state $q \in Q_{FSM}$ and each termination code j of q the state that results from state q if the child behaviour associated with the state q returns at finish the value j.

The behaviour of an FSM is more easily understood when this is represented graphically in the form of a state transition diagram. The control states are represented by circles, and the transition rules are specified as directed edges. Each transition from a state q is labelled by termination code of q that triggers the transition. The arc without a source state denotes then initial state of the system (state q_0).

During one reaction of the FSM, one transition is triggered, chosen from the set of admissible transitions (outgoing transitions from the current state), so that label of transition matches the terminating code of the current state. The FSM goes to the destination state of the triggered transition.

If terminating code of the current state $q \notin F$ is not explicit associated with an admissible transition, then:

- if exist the admissible transition labelled with *default*, this transition (called *implicit transition*) will be triggered;
- else FSM goes in an inconsistent state.

In case if FSM arrive in a state $q \in F$, after completeness of activities from that state, execution of finite state machine is stopped.

4.3 ATL model of the FSMBehaviour

For a JADE finite state machine defined in section 4.2, the equivalent concurrent game structure $S = \langle \Lambda, Q, \Gamma, \gamma, M, d, \delta \rangle$ is defined as follows:

- There is only one agent, i.e. $\Lambda = \{1\}$;

- The set of states is $Q = Q_{FSM}$;
- The finite *set of propositions* is defined by $\Gamma = \Pi \cup \{*FINAL*\}$;
- The *labelling function* $\gamma : Q \to 2^{\Gamma}$ is defined as follows:

$$\gamma(q) = \begin{cases} \pi(q) for\ q \in Q \backslash F; \\ \pi(q) \cup \{*FINAL*\} for\ q \in F. \end{cases}$$

- The nonempty finite *set of moves* M contains all admissible termination codes, i.e.:

$$M = \bigcup_{q \in Q} t(q)$$

- The *alternative moves function* $d : \Lambda \times Q \to 2^M$ is defined by $d(1, q) = t(q)\ \forall q \in Q$
- The *transition function* δ is defines as follows: $\delta(q, \langle j \rangle) = \delta_{FSM}(q, j)\ \forall q \in Q\ and\ \forall j \in t(q)$.

4.4 Using ATL for verification of the FSM - driven behaviour of a JADE agent

Discrepancies between actual and expected results are called *conformance failures* and may indicate any of the following: implementation bug, modelling error, specification error or design error.

Because testing and simulation can give us only confidence in the implementation of a software system, but cannot prove that all bugs have been found, we will use a formal method, the ATL model checking, for detecting and eliminating bugs in the design of a FSM - driven behaviour of a JADE agent.

Design validation using ATL involves checking whether a system design satisfies the system requirements expressed by ATL formulas.

For a given JADE FSMBehaviour, the ATL model checking is done in two steps:

1. For the beginning, the corresponding ATL is constructed following rules described in section 4.3
2. Then, a given specification (ATL formula) representing a desired behavioural property is verified to hold for the model obtained at step 1.

Using ATL Library [9] to perform ATL model checking, we can detect error states (the states of the model where the ATL formula does not hold) and then we can correct the given model or design.

5 Using ATL Library to verify the design of JADE agents

Using the Java version of our ATL Library, the standard methods of JADE FSMBehaviour have been overwritten such that building of the ATL model to be done in parallel with the definition of the FSM.

The new class $ATL\_FSMBehaviour$ extends the functionality of the standard JADE class FSMBehaviour by adding ATL model checking capability:

```
public class MyAgent extends Agent {
  // State names
  private static final String STATE_A = "a";
  private static final String STATE_B = "b";
  private static final String STATE_C = "c";
  private static final String STATE_D = "d";
  private static final String STATE_E = "e";
  private boolean wellDefined = false;

  protected void setup() {
    ATL_FSMBehaviour fsm = new  ATL_FSMBehaviour(this);
    fsm.registerFirstState(new RandomGenerator(2), STATE_A);
    fsm.registerState(new NamePrinter(), STATE_B);
    fsm.registerState(new NamePrinter(), STATE_C);
    fsm.registerState(new NamePrinter(), STATE_D);
    fsm.registerLastState(new NamePrinter(), STATE_E);
    fsm.registerTransition(STATE_A, STATE_B, 0);
    fsm.registerTransition(STATE_A, STATE_E, 1);
    fsm.registerTransition(STATE_B, STATE_C, 0);
    fsm.registerTransition(STATE_C, STATE_D, 0);
    fsm.registerTransition(STATE_D, STATE_B, 0);
    wellDefined = fsm.checkFSM();
    System.out.print("Execution: ");
    addBehaviour(fsm);
    }
  }
```

In the above example, the child behaviour *NamePrinter* displays only the name of the parent state. The *RandomGenerator* behaviour allows FSM to randomly select the transition to fire from the parent state (the terminating event of the parent state is chosen randomly from the admissible values). The *checkFSM()* make calls in ATL Library to perform verification of the defined FSM.

In our example, the ATL formula checked is:

$$\langle\langle \mathcal{A} \rangle\rangle \Diamond \ (*FINAL*) \tag{2}$$

Thus, we verify that in every computation the agent will reach a final state.

In figure 3 is presented the underlying ATL model of the FSMBehaviour described in **MyAgent class** and loaded in ATL Designer:

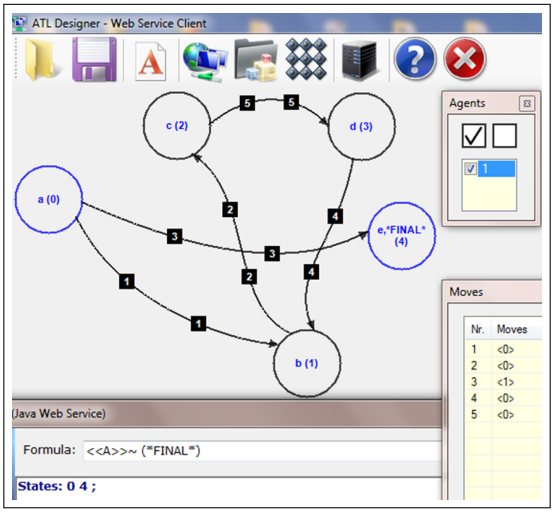


Figure 3: Checking the ATL model in ATL Designer

As we can see from the figure 3, the desired behavioural property expressed by ATL formula (2) does not hold for the entire model. The only states in which the formula holds are STATE_A (0) and STATE_E (4). This can lead to unexpected results at runtime, as we can see from the figure 4:

INFO:
Agent container Container-4@Server1
is ready.

States: 0 4 ;
The ATL formula is satisfied in the following states: 0 4
The FSM is not well defined!
Execution: ae
The end

Figure 4a: The final state is reached

INFO:
Agent container Container-1@Server1
is ready.

States: 0 4 ;
The ATL formula is satisfied in the following states: 0 4
The FSM is not well defined!
Execution:abcdbcdbcdbcdbcdbcdbcdbcdb
cdbcdbcdbcdbcdbcdbcdbcdbcdbcdbcdbcd
bcdbcdbcdbcdbcdbcdbcdbcd....

Figure 4b: The final state is not reached, there is an infinite loop!

Figure 4: Unexpected results at runtime

In the figure 4 is presented a FSM with a correct design, satisfying the specification (2):

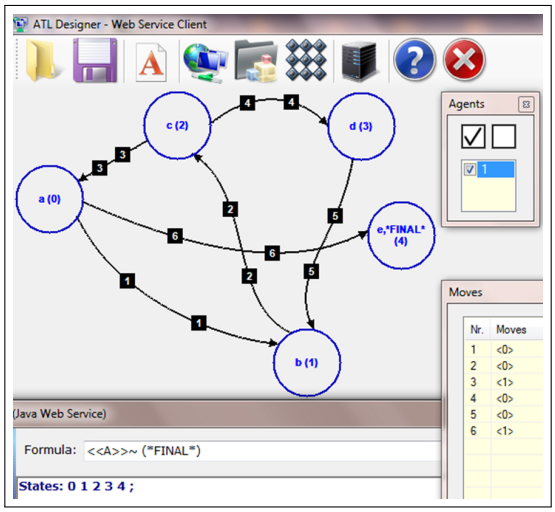


Figure 5: An revised ATL model in which the ATL formula (2) holds

In figure 6a. is presented the revised version of the JADE FSMBehaviour, so that the corresponding ATL model which was built automatically satisfies the specification (2). Figure 6b. shows some execution scenarios of the new JADE agent. Obviously, in all cases the final state will be reached and agent will terminate properly.

```
fsm.registerFirstState(
      new RandomGenerator(2), STATE_A);
fsm.registerState(new NamePrinter(), STATE_B);
fsm.registerState(
      new RandomGenerator(2), STATE_C);
fsm.registerState(new NamePrinter(), STATE_D);
fsm.registerLastState(
      new NamePrinter(), STATE_E);
fsm.registerTransition(STATE_A, STATE_B, 0);
fsm.registerTransition(STATE_A, STATE_E, 1);
fsm.registerTransition(STATE_B, STATE_C, 0);
fsm.registerTransition(STATE_C, STATE_D, 0);
fsm.registerTransition(
      STATE_C, STATE_A, 1);
fsm.registerTransition(STATE_D, STATE_B, 0);
```

Figure 6a: The revised code of the JADE agent

```
States: 0 1 2 3 4 ;
The ATL formula is satisfied in
the following states: 0 1 2 3 4
The FSM is well defined.
Execution: abcdbcae
The end
Execution: abcdbcabcdbcdbcae
The end
Execution: ae
The end
Execution: abcabcdbcdbcdbc
dbcabcdbcabcae
The end
```

Figure 6b: The verification of the ATL model ensure that in all scenarios the agent will behave correctly

6 Conclusions

In this paper Alternating-time Temporal Logic was used for the automated verification of software systems based on JADE agents with FSM - driven behaviours. Our solution is based on Java version of ATL Library, a component of our ATL model checker tool.

The proposed method can be easily adapted to any software system (written in Java or *C#*) capable to exhibit an ATL model (in the form of a concurrent game structure). The original classes (agents) can be extended with specific functionality in order to create an ATL model on-the-fly, while executing the regular code of the application. The ATL model thus constructed is verified against specifications expressed by ATL formulas, at debugging time. After the validation phase, the model checking functionality will be disabled in the release build of the software system.

Our ATL model checker tool is based on client/server architecture to address the time constraints in verification of large models. By using a database-based technology in the core of the ATL model checker, our tool provides a good foundation for further improvement of its performance and scalability.

On the client side, ATL Designer allows an interactive graphical specification of an ATL model as a directed multi-graph. Programmatic construction of a large ATL model is made using ATL Library, available for *C#* and Java languages by now, but we intend to provide interfaces for other popular programming languages.

The server component of our tool (ATL Checker) was published as a Web service, exposing its functionality through standard interfaces. By using Web Services a client can perform model checking without the need of configuring servers.

All components of the new ATL model checker tool can be downloaded from `http://use-it.ro`.

Further investigation on improving performance in verification of ATL formulas will be done through parallelized SQL queries using the horizontal scalability features such as Parallel Pipelined Table Functions (PTF) provided by Oracle databases.

Bibliography

[1] K.Y. Rozier (2011); Survey: Linear Temporal Logic Symbolic Model Checking, *Computer Science Review*, vol. 5, Issue 2: 163-203.

[2] J. Barnat, L. Brim, P. Rockai (2010); Scalable Shared Memory LTL Model Checking, *International Journal on Software Tools for Technology Transfer (STTT)*, 12(2): 139-153.

[3] R. Alur, T.A. Henzinger, O. Kupferman (2002); *Alternating-time temporal logic, Journal of the ACM*, 49(5): 672-713.

[4] M. Kacprzak, W.Penczek (2005); Fully symbolic Unbounded Model Checking for Alternating-time Temporal Logic, *Journal Autonomous Agents and Multi-Agent System*, 11(1): 69-89.

[5] R. Alur et al (1998); Mocha: modularity in model checking, in Proc. Of CAV 98, vol. 1427 of *Lect. Notes in Comp. Sci., Springer-Verlag*: 521-525.

[6] A. Lomuscio, F.Raimondi (2006); Mcmas: A model checker for multi-agent systems, in Proc. of TACAS 06, vol. 3920 of *Lect. Notes in Comp. Sci., Springer-Verlag*: 450-454.

[7] F. Stoica, L.F. Cacovean (2014); Implementing an ATL Model Checker tool using Relational Algebra concepts, in *Proceeding The 22th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*: 361-366.

[8] W. Van der Hoek, M. Wooldridge (2002); Tractable multiagent planning for epistemic goals, in *Proceedings of AAMAS-02, ACM Press*: 1167-1174.

[9] L.F. Stoica, F. Stoica ; WebCheck - ATL/CTL model checker tool, http://use-it.ro

[10] Java Agent Development Framework (JADE), http://jade.tilab.com/

[11] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa (2013); JADE programmer's guide, http://jade.tilab.com

[12] R. E. Bryant (1986); Graph-based algorithms for boolean function manipulation, *IEEE Transactions on Computers*, C-35(8): 677-691.

[13] C. Eisner, D. Peled (2002); Comparing Symbolic and Explicit Model Checking of a Software System, In *Proc. SPIN Workshop on Model Checking of Software*, volume 2318 of LNCS, Volume 55: 230-239.

[14] F. Lerda, N. Sinha, M. Theobald (2003); Symbolic Model Checking of Software, *Electronic Notes in Theoretical Computer Science*, 89(3): 480-498.

[15] B. Bingham, J. Bingham, F. M. de Paula,; J. Erickson, G. Singh, M. Reitblatt (2010); Industrial Strength Distributed Explicit State Model Checking, *Proc.of the 2010 Ninth International Workshop on Parallel and Distributed Methods in Verification*, and *Second Int. Workshop on High Performance Computational Systems Biology (PDMC-HIBI '10)*, *IEEE Computer Society*, Washington, DC, USA, 28-36.

[16] A. J. Hu (1995); Techniques for Efficient Formal Verification Using Binary Decision Diagrams, *PhD thesis*, Stanford University.

[17] J. Ruan (2008); Reasoning about Time, Action and Knowledge in Multi-Agent Systems, *Ph.D. Thesis*, University of Liverpool, http://ac.jiruan.net/thesis/.

[18] D. Owen, T. Menzies (2003); Lurch: a Lightweight Alternative to Model Checking, In *Software Engineering and Knowledge Engineering (SEKE)*: 158-165.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):732-745, October, 2015.

A Modified Membrane-Inspired Algorithm Based on Particle Swarm Optimization for Mobile Robot Path Planning

X.Y. Wang, G.X. Zhang, J.B. Zhao, H.N. Rong, F. Ipate, R. Lefticaru

Xueyuan Wang
1. School of Electrical Engineering
Southwest Jiaotong University
Chengdu 610031, P.R. China
2. School of Information Engineering
Southwest University of Science and Technology
MianYang 621010, P.R.China
121053406@qq.com

Gexiang Zhang*, Junbo Zhao, Haina Rong
School of Electrical Engineering
Southwest Jiaotong University
Chengdu 610031, P.R. China
gexiangzhang@gmail.com, junbozhao55589@gmail.com, ronghaina@126.com
*Corresponding author: gexiangzhang@gmail.com

Florentin Ipate, Raluca Lefticaru
Faculty of Mathematics and Computer Science
University of Bucharest
Academiei 14, Bucharest, Romania
florentin.ipate@ifsoft.ro, raluca.lefticaru@fmi.unibuc.ro

Abstract: To solve the multi-objective mobile robot path planning in a dangerous environment with dynamic obstacles, this paper proposes a modified membrane-inspired algorithm based on particle swarm optimization (mMPSO), which combines membrane systems with particle swarm optimization. In mMPSO, a dynamic double one-level membrane structure is introduced to arrange the particles with various dimensions and perform the communications between particles in different membranes; a point repair algorithm is presented to change an infeasible path into a feasible path; a smoothness algorithm is proposed to remove the redundant information of a feasible path; inspired by the idea of tightening the fishing line, a moving direction adjustment for each node of a path is introduced to enhance the algorithm performance. Extensive experiments conducted in different environments with three kinds of grid models and five kinds of obstacles show the effectiveness and practicality of mMPSO.
Keywords: Membrane computing, evolutionary membrane computing, particle swarm optimization, variable dimensions, mobile robot path planning, membrane systems.

1 Introduction

As a branch of natural computing, membrane computing(MC), initiated by Păun in 1998, aims to abstract distributed and parallel computing models, also called P systems or membrane systems, from the compartmentalized structure and interactions of living cells [1, 2]. There are three main research directions in this area [3]: theoretical study including computing models and their computing power and efficiency; applications such as modeling biological processes and approximately solving engineering optimization problems [4]; software and hardware realization. In the past seventeen years, much attention has been paid to the theoretical aspects, but the applications are worth further discussing, especially for solving real-world engineering problems.

Copyright © 2006-2015 by CCC Publications

Evolutionary algorithms (EAs) are a class of probabilistic search methods with many advantages such as flexibility, convenient application and robustness. While MC can provide flexible evolution rules and parallel-distributed framework [5], which is very beneficial to produce the membrane-inspired evolutionary algorithms (MIEAs). Until now, different kinds of MIEAs have been proposed. In [6], a certain number of nested membrane structures in the skin membrane were combined with EAs for multi-objective optimization problems. A novel MIEA, called QEPS, combining quantum-inspired evolutionary algorithms with P systems to solve image processing problems and knapsack problems were proposed in [7, 8]. Particle swarm optimization (PSO) [9] with one-level membrane structure (OLMS) was used to solve broadcasting problems of P systems and radio spectrum allocation, respectively [10, 13]. In [11], Xiao et al. applied the bio-inspired algorithm based on membrane computing for engineering design problems.

The investigations verify the usefulness of the introduction of P systems into EAs to solve various real-world applications. To the best of our knowledge, there is not any work focusing on the use of a MIEA to solve mobile robot path planning problems (MR3P), which is one of very important real-world applications.

In this paper, a modified membrane-inspired algorithm based on particle swarm optimization (mMPSO) is proposed to solve MR3P. The main contributions of this paper can be summarized as follows: (1) In this study, the solving process of MR3P is considered as a dimension-reducing optimization procedure and therefore a PSO with variable dimensions (vPSO) is introduced into mMPSO, and further a dynamic double OLMS (D-OLMS) with membrane division and dissolution is presented to combine with vPSO to arrange the particles and execute the communications between regions delimited by membranes. (2) Mobile robot path planning is a multi-objective optimization problem. This study considers three objectives, *distance*, *safety* and *smoothness*, instead of a single objective (path length) [15–17] or bi-objectives (path length and risk degree) as previously considered in the literature [18, 21]. A point repair algorithm and a smoothness approach are presented to effectively trade-off multiple objectives and speed-up the mMPSO convergence. (3) Inspired by the idea of tightening the fishing line, a moving direction adjustment for each node of a path is introduced to enhance the algorithm performance, together with the point repair algorithm. (4) Extensive experiments are carried out by considering various environments with different grid models and different obstacles to verify the effectiveness and practicality of mMPSO.

The rest of this paper is organized as follows. Section 2 describes MR3P. In Section 3, we present mMPSO for solving MR3P. Section 4 discusses parameter setting and provides experimental results. Conclusions are drawn in Section 5.

2 Mobile Robot Path Planning Problems

This section gives a brief description of MR3P and then summarizes the related works.

2.1 Problem Statement

MR3P aims to find a reasonable collision-free path for a mobile robot from the starting position to the target position through an environment containing static or dynamic obstacles. It is proved that MR3P is an NP-complete problem [19]. Mobile robots are very useful for the dangerous or hostile environments that humans are not able to reach. So, in recent years, mobile robotics field is a very hot and also challenging research area. As one of important research themes in the mobile robotics field, MR3P, launched at the 1960s, has become an attractive area.

Generally, the criterion for planning a mobile path has to consider many factors, such as the shortest distance, safety degree, smoothness, the lowest energy cost and minimum time, based on the characteristics of a special robot with the minimal turning radius, acceleration and the limited velocity and the features of the environment, such as the distances between obstacles, the shapes of obstacles, the occurrence probabilities of dynamic obstacles. Thus, the optimization of a mobile robot path in a static or dynamic environment is very complicated. To realize the real time robot path planning in a dynamic environment or the consideration that a robot can dynamically track the motion target, at least three aspects should be carefully considered together due to their interactions among each other.

(I) An efficient and effective optimization approach is very important for planning a good mobile robot path. Aiming to solve MR3P in a dynamic environment, this study proposes mMPSO.

(II) A simple and good objective function is very important for planning a good mobile robot path. In this study, the objective function aims to minimize the path length, to maximize the smoothness and the distances between a robot and the obstacles or dangerous sources, and can be expressed as

$$f = K_d \cdot Dis + K_f \cdot S + K_s \cdot SD \tag{1}$$

where K_d, K_f, K_s are the weighing factors of path length, smoothness and safety degree, respectively. The detailed description of path length, smoothness and safety degree are as follows:

1. ***Path length***: Path length Dis is the sum of distances between n nodes from the starting point to the end point and can be described as

$$Dis = \sum_{i=0}^{n-1} L\left(i, i+1\right) \tag{2}$$

where $L(i, i+1) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ is the distance between nodes i and $i+1$, where x_i and x_{i+1} are the x-axis values of nodes i and $i+1$; y_i and y_{i+1} are the y-axis values of nodes i and $i+1$, respectively.

2. ***Smoothness***: Smoothness refers to the sum of the reflection angles formed by any three neighboring nodes of a path. As usually calculating directly the smoothness is a time-consuming process, this study uses an indirect approach, i.e., it uses the ratio S_c of the number of deflection angles less than the given expected value to the total number of deflection angles and the ratio S_p of the number of path segments more than the number of the segments in the path with the smallest number of path segments in a group to the total number of path segments to evaluate the smoothness of a path. Smoothness can be calculated by using the following formula:

$$S = \alpha \cdot S_c + \beta \cdot S_p \tag{3}$$

where $S_c = 1 - \frac{DA_l}{N_f - 1}$; $S_p = 1 - \frac{S_{min}}{N_f}$, where N_f is the total number of path segments; DA_l is the number of deflection angles greater than the expected value; S_{min} is the number of the segments in the path with the smallest number of path segments in a group; α and β are two weighting coefficients.

3. ***Safety degree***: Safety degree (SD) is the sum of deviation degrees C_i $(i = 1, 2, \ldots, N)$ between any segment in a path and its nearest obstacle. SD is defined as

$$SD = \sum_{i=1}^{n-1} C_i = \begin{cases} 0, & d_i \geq \lambda \\ \sum_{i=1}^{n-1} e^{\lambda - d_i}, & d_i < \lambda \end{cases} \tag{4}$$

where d_i is the minimal distance between the ith segment and its nearest obstacle; λ is the threshold of safty degree.

(III) The establishment of an environment model is the foundation of MR3P and decides the environment feature (static or dynamic), and how to choose an evaluation method and an optimization approach to implement the path planning for a mobile robot. There are three main environment models: vector (obstacles represented by polygons), grid (occupancy cell) and graph (Voronoi diagram or visibility graph). As compared with vector and graph, grid has the advantages of simple and flexible. This study uses a grid environment. There are two ways of representing a grid-based environment. One is a X-Y coordinates plane [15] and the other is an orderly numbered grid, which has been widely used. We adopt the latter approach, in which a square environment is evenly divided into a certain number of squares, i.e., the x-axis and y-axis are divided equally into m parts, thus, we get $m \times m$ grids, where one or more grids are used to represent the obstacles. An example of the 7×7 grids is shown in Fig. 1(a), where the grid map is encoded by using Matlab and the grey shadow grids represent obstacles.

The mapping relations between coordinates (x, y) and the serial number p beginning from one can be identified by the following formula:

$$p = fix(y/SoG) \cdot NoC + fix(x/SoG) \tag{5}$$

where NoC is the number of columns; SoG is the size of a grid; the function $fix(t)$ rounds t to its nearest integer towards zero.

2.2 Related work

Since the pioneering work of Lozano-Pérez [12], a number of algorithms for solving the path planning problems have been reported in the past thirty years. These algorithms can be generally divided into two main classes: classical [14] and heuristic [19]. Although the classic approaches can be used to solve this problem, they may suffer from some drawbacks, such as easily falling into local minima, high complexity in high dimensions. In order to overcome these problems of classic methods, heuristic algorithms have been developed.

The representative heuristic approaches for solving MR3P are neural networks, genetic algorithms [15], ant colony optimization, fuzzy logic [16], simulated annealing [17], PSO [21], probabilistic road maps, rapidly exploring random trees, etc. Although heuristic methods do not guarantee to find an optimal solution, they may be faster and may have higher efficiency than classical methods [19]. The studies in [20, 21] have shown that the interest in PSO-based meta heuristics algorithms is growing in mobile robotics. Particularly interesting is the work in [20, 21] for solving the static or dynamic MR3P. According to the reports in the literature, the dimensions of the search space are set to a fixed value and remain constant throughout the entire optimization process in almost all of PSO-based algorithms for solving MR3P; consequently, the solving ability of the algorithms is limited to a single individual's dimension and the algorithm cannot find the optimal solutions. In MR3P, the dimensions of the search space decide the number of nodes of the optimal path. High dimensions may result in the decrease of the searching efficiency, while low dimensions may cause the case in which it is impossible to get barrier-free paths. In order to find a proper dimension for a path and improve the search efficiency to self-adapt the dynamic environment with randomly appearing or disappearing obstacles, mMPSO with variable dimensions is introduced to solve MR3P and will be presented in the next section.

3 mMPSO for MR3P

This study considers a grid-based environment, in which moving obstacles or dangerous sources may appear or disappear. To adapt the hostile environment, mMPSO uses a variable dimension PSO and a dynamic membrane structure with membrane division and dissolution. To improve the mMPSO performance such as effectiveness, efficiency and extensibility, we introduce a point repair algorithm, a smoothness approach and a moving direction adjustment technique. In what follows, we first present the variable dimensions and then describe the point repair algorithm and the smoothness approach. Finally, we summarize the mMPSO algorithm.

3.1 Variable Dimensions

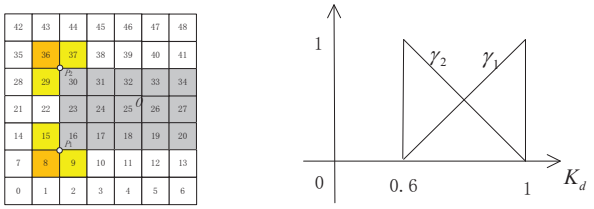
In mMPSO, each particle represents a feasible path, instead of an infeasible path in other heuristic approaches such as genetic algorithms. If the dimension of each particle is fixed, the search efficiency is often low due to the following reasons: (1) Population initialization, i.e., obtaining a population of initially feasible paths through randomly searching each node row by row, is time-consuming, especially for large grids or complex environments with circuitous route phenomenon. (2) The search process of the algorithm with fixed dimensions is also time-consuming, as compared with the dimension-reducing methods, because the variable dimensions in this study consider the removal of redundant information at each iteration. (3) Due to the complex or hostile environment, the optimization algorithms with fixed dimensions have low efficiency and poor adaptability.

To overcome these shortcomings, a set of high-dimension particles are needed at the beginning and the dimension of the best solution, i.e., the optimal path, is usually quite low. Thus, the dimensions of each particle in mMPSO are considered to be variable. In mMPSO for solving MR3P, the initial population P of particles (feasible paths) is classified into several subpopulations P_{min},...,P_{max}, where P_{min} is the subpopulation with lower dimensions, which represent shorter paths, that pass around fewer obstacles, and P_{max} is the subpopulation with higher dimensions, which denote longer paths, passing around more obstacles. At the beginning, the population size S_{min} of P_{min} may be similar to the one S_{max} of P_{max}, and the particles in P_{max} may search a feasible path through passing around external obstacles, while the particles in P_{min} may go to the contrary case. As the algorithm goes forward, S_{min} will increase and S_{max} will decrease. In general, a particle with low dimensions produces a shorter path, while a particle with high dimensions corresponds to a longer path. However, there are still some exceptions. But in mMPSO, the point repair algorithm, the smoothness approach and the moving direction adjustment technique can rectify the exceptions. The implementation of variable dimensions motivates the dynamic membrane structure of mMPSO.

3.2 Point Repair Algorithm

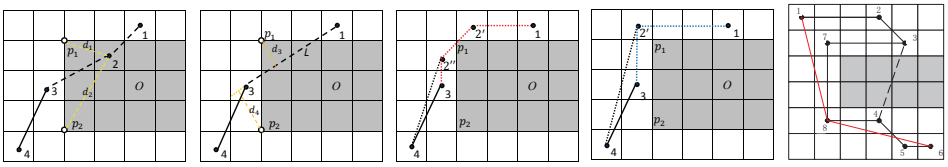
In the process of searching the optimal path, some nodes may move into obstacles and some path segments may cross obstacles, which results in infeasible paths and it is necessary to repair them. This study introduces a point repair algorithm to change the infeasible paths into feasible paths. We first define some special grids in the environment model. In Fig. 1(a), p_1, which is surrounded by the three peripheral grids, 8, 9 and 15, and p_2, which is surrounded by the three peripheral grids, 29, 36 and 37, are the vertexes of obstacle O. All the peripheral grids have two kinds of coefficients, γ_1 and γ_2, which are randomly selected and are controlled by the weighing factors, K_d, K_s and K_f in (1). The coefficient γ_1 related to lateral grids $\{9, 15, 29, 37\}$ is mainly controlled by K_d. The coefficient γ_2 related to the diagonal grids $\{8, 36\}$ is mainly controlled by K_s and K_f, where $K_d + K_s + K_f = 1$, $0.6 \leq K_d \leq 1$, $0 \leq \gamma_1 \leq 1$, $0 \leq \gamma_2 \leq 1$, $\gamma_1+\gamma_2 = 1$. The

relationship between γ_1 and γ_2 is shown in Fig. 1(b). For example, if $K_d = 0.8$, $K_s = K_f = 0.1$, we obtain $\gamma_1 = 0.5$ and $\gamma_2 = 0.5$. If $K_d = 1$, $K_s = K_f = 0$, we get $\gamma_1 = 1$ and $\gamma_2 = 0$.



(a) (b)

Figure 1: Definition of grids



(a) (b) (c) (d) (e)

Figure 2: An example for point repair and smoothness algorithms

Two types of infeasible paths are shown in Fig. 2(a)-(b), where d_1 is the point to point distance between node 2 and p_1; d_2 is the point to point distance between node 2 and p_2; d_3 is the point to line distance between p_1 and L; d_4 is the point to line distance between p_2 and L; d_1, d_2, d_3 and d_4 decide which peripheral grids will be selected. In Fig. 2(a), the node 2 in the path $\{1, 2, 3, 4\}$ (Type 1) must be repaired. In Fig. 2(b), the path segment L crossing the obstacle (Type 2) must be broken. The point repair process is as follows.

Step 1: Evaluate a path found. If it is feasible, we skip the repair process, otherwise, we perform the repair process.

Step 2: Judge the type of an infeasible path, Types 1 or 2.

Step 3: If the infeasible path is Type 2, go to Step 4, otherwise, conduct the following steps:

(a) Calculate the distance between the infeasible point p_0 and the vertex p_i of the obstacle, then get the value(s) of d_i, $i = 1$ (the obstacle is in the corner) or $i = 1, 2$ (the obstacle is located at the edge of the map) or $i = 1, 2, 3, 4$ (the obstacle is located in the middle of the map). Next, sort d_i in an increasing order.

(b) Select unused p_i by using the corresponding smallest value in d_i and get the peripheral grids $p^j_{g\_i}, j = 1, 2, 3$, and randomly select unused $p^j_{g\_i}$ by using γ_1 and γ_2.

(c) Replace the value of the infeasible node with selected $p^j_{g\_i}$.

(d) Judge the path again. If it is feasible, this process terminates, otherwise, go to step 2.

Step 4: Type 2 is repaired according to the following steps:

(a) Calculate the distance between the infeasible path segment L and the vertex p_i of the obstacle, get the value(s) of d'_i, $i = 1$ (the obstacle is in the corner) or $i = 1, 2$ (the obstacle is located at the edge of the map) or $i = 1, 2, 3, 4$ (the obstacle is located in the middle of the map). Next, sort d'_i in an increasing order.

(b) Select unused p_i by using the corresponding smallest value in d'_i and get the peripheral grids $p^j_{g\_i}, j = 1, 2, 3$, and randomly select unused $p^j_{g\_i}$ by using γ_1 and γ_2.

(c) Insert the selected $p^j_{g\_i}$ between the two nodes of the infeasible path segment and get two new path segments $path_{s\_1}$ and $path_{s\_2}$.

(d) Judge each of the two paths $path_{s\_1}$ and $path_{s\_2}$. If one of them is not feasible, go to step 4, otherwise, the repair process terminates.

There are three cases of infeasible paths shown in Fig. 2(e) (dash line) and Fig. 2(a)-(b). We use the introduced point repair algorithm to process the three cases and obtain the corresponding results shown in Fig. 2(e) (solid line) and Fig. 2(c)-(d), respectively. E.g., the path segment $\{3, 4\}$ across the obstacle O in the infeasible path represented by the nodes $\{1, 2, 3, 4, 5, 6\}$ in Fig. 2(e) should be modified. If $\gamma_1 \gg \gamma_2$, the path segment $\{3, 4\}$ is replaced by two path segments $\{3, 8\}$ and $\{8, 4\}$ in the first modification, but the modified path segment $\{3, 8\}$ is still infeasible and must be modified further. In the second modification, the segment $\{3, 8\}$ is replaced by the path segments $\{3, 7\}$ and $\{7, 8\}$. Thus, all the path segments are feasible and the new path is $\{1, 2, 3, 7, 8, 4, 5, 6\}$. In Fig. 2(c)-(d), the new feasible path is $\{1, 2', 2'', 3, 4\}$ or $\{1, 2', 3, 4\}$ under the condition $\gamma_1 = 0, \gamma_2 = 1$ or $\gamma_1 = 1, \gamma_2 = 0$.

3.3 Smoothness Algorithm

The smoothness algorithm is used to get rid of those redundant nodes of a feasible path. The smoothness process is described as follows:

Step 1: Sort the nodes in a path from the starting node to the goal node and get a sequence $n_i, i = 1, 2, \ldots, m$, where m is the dimension of a particle; n_1 and n_m are the starting and goal nodes, respectively.

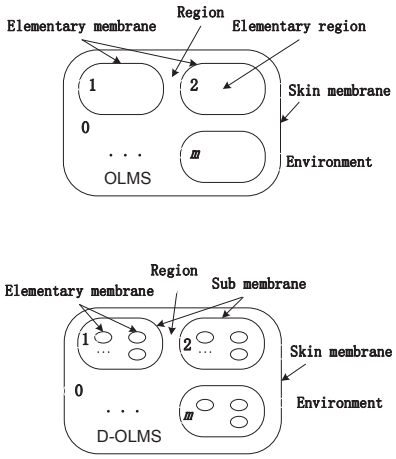
Step 2: Judge the path segment L_{ij} between n_i and n_j (at the beginning, $i = 1, j = 3$), if L_{ij} is infeasible, insert the nodes i and $j - 1$ into the node set P_f of the smoothed path, i.e., $P_f = \{i, j - 1\}$ and let $i = j - 1$ and j increase 1, the algorithm continues to judge the path segment L_{ij}, otherwise, let j increase 1, continue to judge the feasibility of the path segment L_{ij} till it is infeasible, insert the nodes i and $j - 1$ into the node set P_f of the smoothed path and let $i = j - 1$. Repeat this step till $j = m$.

As shown in Fig. 2(e), we use the introduced smoothness algorithm to remove the redundant nodes in the path $\{1, 2, 3, 7, 8, 4, 5, 6\}$ and obtain the smoothed path $\{1, 8, 6\}$. Similarly, the smoothed paths $\{1, 2', 2'', 4\}$ and $\{1, 2', 4\}$ come from the paths $\{1, 2', 2'', 3, 4\}$ and $\{1, 2', 3, 4\}$, respectively, as shown in Fig. 2(c)-(d).

3.4 mMPSO

In mMPSO, a dynamic membrane structure (specifically, OLMS alternates with D-OLMS shown in Fig. 3(a)) is introduced to arrange a population of particles, each of which is a feasible path for a mobile robot, and specify various rules, such as membrane division, transformation and communication-like rules, and membrane dissolution. The dimension of each particle is variable in the process of evolution. The point repair algorithm described above is used to change infeasible paths into feasible ones. The repair process may increase the dimensions of each particle. The smoothness algorithm presented in this section is applied to remove the redundant nodes of a path and the process may decrease the dimensions of each particle. In addition, a moving direction adjustment technique is presented to accelerate the algorithm convergence. The pseudocode algorithm of mMPSO is shown in Fig. 3(b), where each step is described in detail as follows:

Step 1: An OLMS $[[\,]_1, \ldots, [\,]_m]_0$ composed of a skin membrane denoted by 0 and $m + 1$ regions inside the skin membrane is constructed. The way in which the parameter m is chosen will be discussed in Section 4.



(a)

Begin:
$t \leftarrow 1$
(1) Initialize membrane structure
(2) Initialize populations
(3) Produce new particles in each elementary membrane
(4) **While** (not termination condition) **do**
(5) Divide each elementary membrane
(6) Evaluate every particle
(7) Find $P_{best}^{id}(t)$
(8) Find local best particle $G_{ij}^{b}(t)$
(9) Execute communication rules (a)
(10) Find global best particle $G_{d}^{b}(t)$
(11) Execute communication rules (b)
(12) Update particle's velocity $V(t)$
(13) Update particle's position $X(t)$
(14) Execute point repair algorithm
(15) Execute smoothness algorithm
(16) Adjust each particle's moving direction
(17) Dissolve elementary membrane
$t \leftarrow t+1$
End
End

(b)

Figure 3: Membrane structures and the pseudocode algorithm for mMPSO

Step 2: A particle swarm X with m particles in a D-dimensional search space is randomly generated and each particle is put inside an elementary membrane in OLMS, where D represents the number of nodes in a feasible path; $X = \{x_1, x_2, \ldots, x_m\}$, where x_i is an arbitrary individual in X and denotes a feasible path, $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$.

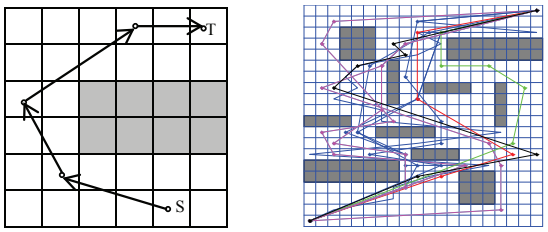
Step 3: In this step, a moving direction adjustment technique is introduced to produce n particles inside each elementary membrane. To be specific, we modify the velocity of the particle inside each elementary membrane to generate a new particle by using (6) and (7),

$$V(g+1) = \rho_1 \cdot V_r(g) + \rho_2 \cdot V_f(g) \tag{6}$$

$$X(g+1) = X(g) + V(g+1) \tag{7}$$

where ρ_1 and ρ_2 are the inertia weighting factors and usually are set to larger values for exploring the global solutions; $V_r(g)$ is the randomly produced velocity of the gth particle (at the beginning for each elementary membrane, g=0); $V_f(g)$ is the adjusted velocity of the gth particle by using the idea of tightening fishing line and the moving directions of each node in the gth particle is shown in Fig. 4(a); $V(g+1)$ is the velocity of the $(g+1)$th particle; $X(g)$ and $X(g+1)$ are positions of the gth and $(g+1)$th particles. Inspired by the idea of tightening fishing line, we consider a feasible path as a fishing line and tighten the line from the target node, thus, each node except for the target one in the path will show a moving direction toward the next node (the target node is the first one). The moving directions of all the nodes in the path construct the velocity $V_f(g)$. This step is repeated for n times to produce n new particles for each elementary membrane and used together with the point repair algorithm and smoothness algorithm. The dimensions of new particles may be greater than or less than D. Thus, the swarm will have $m \times n$ particles in total. Fig. 4(b) shows an example that one particle with 20 dimensions (in thick line) inside a certain elementary membrane produces 10 particles with 6–10 dimensions (in thin line). Compared to the random approach, the production of the new particles can remove redundant nodes of a path and has better adaptability in hostile environment, especially in the circuitous route environment.

Step 4: The maximal number of iterations is used as the termination condition.



(a) (b)

Figure 4: An example of the generation of new particles and direction of each dimension of the individual

Step 5: This step first classifies the n particles inside the ith elementary membrane into k_i clusters according to the dimension of each particle and then divides the ith elementary membrane into k_i membranes, each of which contains the particles with the same dimension, $i = 1, 2, \ldots, m$. Thus, OLMS becomes D-OLMS.

Step 6: Each particle is evaluated by using (1) and assigned a fitness value.

Step 7: Find $P_{best}^{id}(t)$, which is the best solution of each particle in its history with respect to the fitness values.

Step 8: Find the locally best solution $G_{ij}^{b}(t)$ in the jth elementary membrane inside the ith membrane, $i = 1, 2, \ldots, m, j = 1, 2 \ldots, k_i$, in terms of fitness values.

Step 9: Perform communication rules (a), which first send all the locally best solutions $G_{ij}^{b}(t)$ $(j = 1, 2 \ldots, k_i)$ out into the ith submembrane, $i = 1, 2, \ldots, m$, and further send out into the skin membrane.

Step 10: Find the globally best solution $G_{d}^{b}(t)$ by comparing $G_{ij}^{b}(t)$ with the same dimension d, $d \in \{1, 2, \ldots, D\}$, $i = 1, 2, \ldots, m, j = 1, 2 \ldots, k_i$.

Step 11: Perform communication rules (b), which send $G_{d}^{b}(t)$ back into the elementary membrane containing d-dimension particles across certain submembrane.

Step 12: Update the velocities of the d-dimension particles using (8).

$$\begin{aligned} V_{id}(t+1) = \delta_1 \cdot \Big(\rho \cdot V_{id}(t) + c_1 \cdot r_1 \cdot \big(P_{best}^{id}(t) - X_{id}(t)\big) + c_2 \cdot r_2 \cdot \Big(G_{ij}^{b}(t) - X_{id}(t)\Big) \\ + c_3 \cdot r_3 \cdot \big(G_{d}^{b}(t) - X_{id}(t)\big)\Big) + \delta_2 \cdot V_{id}^{f}(t) \end{aligned} \tag{8}$$

where $V_{id}(t)$ and $V_{id}(t+1)$ are the velocities of the particle at generation t and $t+1$, respectively; $P_{best}^{id}(t)$ is the best solution of the particle at generation t; $X_{id}(t)$ is the position the particle at generation t; $G_{ij}^{b}(t)$ is the locally best solution with the same dimension d at generation t; $G_{d}^{b}(t)$ is the globally best solution with the same dimension d at generation t; $V_{id}^{f}(t)$ is the adjusted velocity of the particle at generation t; δ_1 and δ_2 are proportion coefficients; ρ is an inertia weighting factor; r_1 , r_2 and r_3 represent the functions that generate independently random numbers, which are uniformly distributed between 0 and 1; c_1, c_2 and c_3 are acceleration coefficients.

Step 13: Update the positions of the d-dimension particles using (9).

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{9}$$

where $X_{id}(t)$ and $X_{id}(t+1)$ are the positions of the particle at generation t and $t+1$, respectively; $V_{id}(t+1)$ is the velocity of the particle at generation $t+1$.

Step 14: Execute point repair algorithm for each particle.

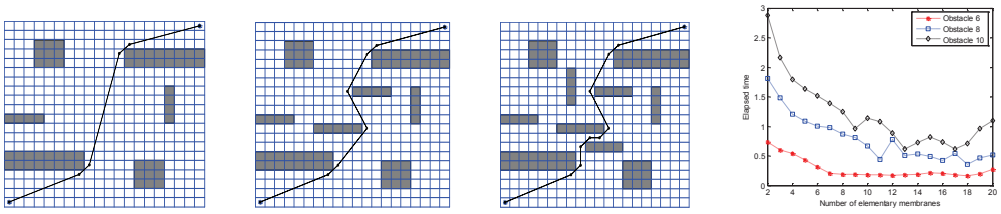
Step 15: Execute smoothness algorithm for each particle.

Step 16: Adjust the moving direction of each particle by using the moving direction adjustment technique.

Step 17: This step dissolves all the elementary membranes and releases their particles into their corresponding submembranes. Thus, D-OLMS becomes the original OLMS.

4 Experimental Results

The mMPSO performance is tested by using MR3P. We first discuss how to set the number m of elementary membranes in OLMS by using 20×20 grid model environment with 6, 8 and 10 obstacles, respectively. Then, 16×16 grid model environment with 9 static obstacles are applied to compare mMPSO with its counterpart vPSO and GA [15]. Subsequently, the complex environments, 32×32 and 64×64 grid model environments with 20 static obstacles, are applied to further test the mMPSO performance. In these experiments, one dynamic obstacle representing a moving obstacle or a dangerous source occurring suddenly is employed to analyze the mMPSO behavior.

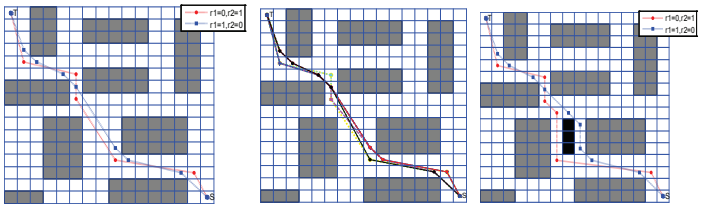


(a) 6 obstacles (b) 8 obstacles (c) 10 obstacles (d) Results

Figure 5: The near shortest paths in three environments (20×20 grid with obstacles 6, 8 and 10, respectively) and experiment results

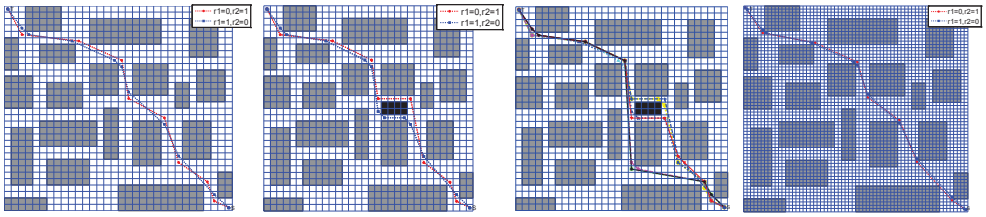
4.1 Parameter setting

In this subsection, we use 20×20 grid model environment with three kinds of obstacles shown in Fig. 5 to discuss how to choose m. Fig. 5(a)-(c) have 6, 8, 10 static obstacles (shaded areas), respectively. In the following experiments, the population size is set to 100; In (8), $c_1 = c_2 = c_3 = G_{size}/V_{\max}$, where $G_{size} = 0.08$ is the size of a grid and $V_{\max}$ is the maximal distance allowing a node to move in a step; the proportion coefficients $\delta_1 = 0.65$, $\delta_2 = 0.35$; ρ is defined as a variable, which varies from 0.246 to 0.157 along the logarithm function $\log_{10}(y)$. In (6), ρ_1 and ρ_2 are set to random values between 0.4 and 0.6. In (3), $\alpha = 0.6$, $\beta = 0.4$. In (4), λ is set to the robot radius.



(a) (b) (c)

Figure 6: Experimental results of mMPSO in the environments 16×16 grids, $O_s = 9$ and $O_d = 1$.



(a) 32×32 (b) 32×32 (c) 32×32 (d) 64×64

Figure 7: Experimental results of mMPSO in the environments 32×32, 64×64 grids, $O_s = 20$ and $O_d = 1$.

In what follows, m varies from 2 to 20 by the interval 2, thus, we first generate m particles in *Step 2* and in *Step 3* for the first $m-1$ elementary membranes, we produce $round(100/m)$ particles and $100-(m-1)*round(100/m)$ particles for the mth elementary membrane, where $round(.)$ is a function for rounding its element towards nearest integer. In the experiments, if a given near-optimal solution is reached, mMPSO stops. Because the optimal solution to MR3P is usually unknown, we set $K_d = 1, K_s = K_f = 0$ in (1) and independently perform mMPSO for 30 times, where the terminal condition is such that the maximal number of iterations is set to 2000, in order to find the near-optimal solution. Fig. 5(a)-(c) show the near shortest paths of the model environment with different obstacles, 6, 8 and 10, respectively. The mMPSO performance for each of the 19 cases is evaluated by using the mean of the elapsed time in 30 independent runs. The experimental results are shown in Fig. 5(d), where the elapsed time for three environments first decreases and then increases as the value of m goes up. These experimental results indicate that m could be assigned as 13 by considering the three environments.

4.2 MR3P Experimental Results

To investigate the mMPSO performance, this subsection uses three grid models, 16×16, 32×32 and 64×64, to carry out the experiments and considers five environments: 16×16 with 9 static obstacles ($O_s = 9$), 16×16 with $O_s = 9$ and one dynamic obstacle ($O_d = 1$), 32×32 with $O_s = 20$, 32×32 with $O_s = 20$ and $O_d = 1$, 64×64 with $O_s = 20$. The place for the possible occurrence of the dynamic obstacle is set to the near center, which is very likely to block the feasible paths. The model with 16×16 grids is applied to compare mMPSO with its counterpart PSO (vPSO) and GA in [15]. The models with 32×32 and 64×64 grids are used to further discuss the mMPSO performance in different complex environments. The setting of the parameters in mMPSO except for K_d, K_s, K_f is the same as in Subsection 4.1. m=13. The termination condition is designated as the maximal number 2000 of iterations. All the experiments are run on the PC with CPU 1.7GHz, 512MB RAM, and the software platform MATLAB7.4 and Windows XP OS.

We first use the model with 16×16 grids to compare mMPSO with vPSO (when $m = 1$, mMPSO becomes vPSO) and GA in [15]. We consider three cases for K_d, K_s, K_f as follows:

(1) Case 1: $K_d = 1, K_s = K_f = 0, \gamma_1 = 1, \gamma_2 = 0$;

(2) Case 2: $K_d = 0.6, K_s = K_f = 0.2, \gamma_1 = 0, \gamma_2 = 1$;

(3) Case 3: $K_d = 0.8, K_s + K_f = 0.2, \gamma_1 + \gamma_2 = 1$.

The experimental results of mMPSO are shown in Fig. 6. In Fig. 6(a), the blue line is the best path in Case 1 considering only one objective, path length, and the red line is the best result in Case 2 by trading-off safety and smoothness. Fig. 6(b) illustrates 8 near optimal paths (8 colors) through balancing the path length, safety and smoothness. The paths in Fig. 6(c)

Table 1: Comparisons of three methods in the environment(Fig. 6 (a))

Method	NoO	NoNO	NoI	Fv	Gn	St
GA[15]	9	78	13	24.68	16	1.68
vPSO	83	108	0	24.95	65	2.97
mMPSO	94	239	0	24.26	27	0.84

Table 2: Comparisons of three methods in the environment(Fig. 6 (c))

Method	NoO	NoNO	NoI	Fv	Gn	St
GA[15]	32	68	0	24.71	12	0.69
vPSO	81	103	0	28.56	73	3.12
mMPSO	92	235	0	27.43	34	0.97

are obtained by considering one dynamic obstacle and the blue line is the best path in Case 1 considering only one objective, path length, and the red line is the best result in Case 2 by trading-off safety and smoothness.

To draw a comparison with GA in [15] and vPSO, let K_d=1 and the experiment is executed for 100 independent runs. Tables 1 and 2 show the experimental results of GA, vPSO and mMPSO for the environments with static obstacles and the environments with static and dynamic obstacles. In Tables 1-3 , NoO, NoNO, NoI, Fv, Gn, St represents the number of optimal solutions, the number of near optimal solutions, the number of infeasible solutions and the fitness value in 100 trials, the average generations for finding the optimal solution and the mean of the elapsed time (s) in each trial, respectively.

As it can be clearly seen from Table 1 and Fig. 6, mMPSO finds much more optimal paths and near optimal paths, while it spends smaller computing time than GA. There are some infeasible solutions in GA, while there is not any infeasible solution in vPSO and mMPSO because the point repair algorithm have repaired the infeasible path. On the other hand, vPSO also finds more optimal paths and near optimal solutions than GA, but the elapsed time is far larger than GA. mMPSO is better than vPSO with respect to optimal and near optimal solutions and the elapsed time, which indicates the advantage of the combination of a membrane system with PSO. Tables 2 shows similar conclusions to those in Tables 1.

To further analyze the mMPSO performance in more complex environments, more experiments are conducted in the environments with 32×32 and 64×64 grids containing 20 or 21 obstacles, as shown in Fig. 7 (a-d). The environment with 32×32 grids and 20 static obstacles are shown in Fig. 7 (a). Fig. 7(b)-(c) show the environment with 20 static obstacles and one dynamic obstacle. In Fig. 7 (c), the three objectives, path length, smoothness and safety, are considered. The parameters of mMPSO are the same as above except for the population size

Table 3: Experimental results of mMPSO in different environments in Fig. 7

Environment	NoO	NoNo	Fv	Gn	St
32×32, $O_s = 20$, $O_d = 0$	86	242	28.79	36	1.72
32×32, $O_s = 20$, $O_d = 1$	82	225	31.53	45	1.93
64×64,$O_s = 20$, $O_d = 0$	83	247	28.14	59	2.68

150 and $m = 15$. All the tests are executed for 100 independent runs. Table 3 shows the results.

It can be seen from Tables 1-3 that the optimal solutions of mMPSO drop from 94 to 83, the elapsed time rises from 0.84 to 2.68 and the average generations vary from 27 to 59 as the number of grids increases from 16×16 to 64×64 and the static obstacles go up from 9 to 20. The elapsed time and average generations increase a little with the dynamic obstacle. To sum up, as the number of model grids increases by 4^n $(n = 1, 2, 3 \ldots)$ and the static obstacles double, the increase of the elapsed time is quite small, instead of an exponential increase. mMPSO maintains good search capability to find the optimal solution in both static and dynamic environments, which indicates mMPSO has good adaptability to MR3P under complex environments.

5 Conclusions

This paper discusses a feasible combination of membrane systems and PSO to solve MR3P. The outstanding novelty is to justify the introduced dynamic membrane structure, which proves to be suitable for solving MR3P with variable dimensions. mMPSO uses the alternation of OLMS and D-OLMS to integrate a PSO with variable dimensions, point repair algorithm, smoothness algorithm and moving direction adjustment. A large number of experiments are carried out on several MR3P with various environments and the results show that mMPSO can achieve much better solutions than its counterparts PSO and GA, as reported in the literature.

This paper considers only the planar (two dimensions) environments. Following this work, some issues need to be further investigated, such as how to extend mMPSO to three dimensional spaces, how to use mMPSO to solve more difficult path planning problems (mobile robots follow the tracks of moving targets in a hostile environments), how to combine mMPSO with numerical P systems to control mobile robots and how to apply the idea of variable dimension PSO to solve more engineer application problems.

Acknowledgment

The work of XW, GZ, JZ and HR was supported by the National Natural Science Foundation of China (61170016, 61373047), the Program for New Century Excellent Talents in University (NCET-11-0715) and SWJTU supported project (SWJTU12CX008); The work of FI and RL was supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE- 2011-3-0688.

Bibliography

[1] G. Păun, G. Rozenberg, A. Salomaa (eds.) (2010); The Oxford handbook of membrane computing, Oxford University Press.

[2] G. Păun (2007); Tracing some open problems in membrane computing, *Rom J Inform Sci Tech*, ISSN 1453-8245, 10(4): 303–314.

[3] G. Zhang, J. Cheng, M. Gheorghe, Q. Meng (2013); A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, *Appl Soft Comput*, ISSN 1568-4946, 13(3):1528-1542.

[4] T.Y. Nishida (2004); An application of P system: a new algorithm for NP-complete optimization problems. *Proc 8th WMCSCI*, 109–112.

[5] G.X. Zhang, M. Gheorghe, L.Q. Pan,; M.J. Pérez-Jiménez (2014); Evolutionary membrane computing: a comprehensive survey and new results, *Inform Sci*, ISSN: 0020-0255, 279: 528-551.

[6] L. Huang, X. He, N. Wang, Y. Xie (2007); P systems based multi-objective optimization algorithm, *Prog Nat Sci*, ISSN 1002-0071, 17(4): 458-465.

[7] G.X. Zhang, M. Gheorghe, Y. Li (2012); A membrane algorithm with quantum-inspired subalgorithms and its application to image processing, *Nat Comput*, ISSN 1567-7818, 11(3): 701-717.

[8] G.X. Zhang, J.X. Cheng, M. Gheorghe (2014); Dynamic behavior analysis of membrane-inspired evolutionary algorithms, *International Journal of Computers Communications & Control*, ISSN 1841-9836, 9(2): 227-242.

[9] J. Kennedy, R. Eberhart (1995); Particle swarm optimization, *Proc ICNN*, 4: 1942-1948.

[10] G.X. Zhang, F. Zhou, X.L. Huang, J.X. Cheng, M. Gheorghe, F. Ipate, R. Lefticaru (2012); A novel membrane algorithm based on particle swarm optimization for solving broadcasting problems, *J Univers Comput Sci*, ISSN 0948-6968 18(13): 1821-1841.

[11] J. Xiao, Y. Huang, Z. Cheng, J. He, Y. Niu (2014); A hybrid membrane evolutionary algorithm for solving constrained optimization problems, *Optik*, ISSN 0030-4026, 125(2): 897-902.

[12] T. Lozano-Pérez,M.A. Wesley (1979); An algorithm for planning collision-free paths among polyhedral obstacles, *Commun ACM*, ISSN 0001-0782, (22)10: 560–570.

[13] H. Gao, J. Cao (2012); Membrane quantum particle swarm optimisation for cognitive radio spectrum allocation, *Int J Comput Appl Tech*, ISSN 0952-8091, 43(4): 359-365.

[14] Y.K. Hwang, N. Ahuja (1992); Gross motion planning-A survey, *ACM Comp Surv*, 24: 219-291.

[15] A. Tuncer, M. Yildirim (2012); Dynamic path planning of mobile robots with improved genetic algorithm, *Comput Electr Eng*, ISSN 0045-7906, 38: 1564-1572.

[16] M.A. Garcia, O. Montiel (2009); Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Appl Soft Comput*, ISSN 1568-4946, 9(3): 1102-1110.

[17] Z. Qidan, Y.J. Yang, Z.Y. Xing (2006); Robot path planning sased on artificial potential field approach with simulated annealing, *Proc ISDA*, 622-627.

[18] Y. Zhang, D.W. Gong (2013); Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neurocomputing*, ISSN 0925-2312, 103: 172-185.

[19] E. Masehian, D. Sedighizadeh (2007); Classic and heuristic approaches in robot motion planning-a chronological review, *Proc. WASET*, 101-106.

[20] W.F. Xu, C. Li, B. Liang (2008); The cartesian path planning of free-floating space robot using particle swarm optimization, *Int J Adv Rob Syst*, ISSN 1729-8806, 5: 301-310.

[21] D.W. Gong, J.H. Zhang (2011); Multi-objective particle swarm optimization for robot path planning in environment with danger sources, *J Comput*, 6(8): 1554-1561.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):746-759, October, 2015.

Multi-attribute Collaborative Filtering Recommendation

C. Yu, Y. Luo, K. Liu

Changrui Yu
School of Information Management and Engineering,
Shanghai University of Finance and Economics,
Shanghai 200433, China
yucr@sjtu.edu.cn

Yan Luo*
SHU-UTS SILC Business School,
Shanghai University,
20 Chengzhong RD, Shanghai 201800, China
*Corresponding author: luoyan@shu.edu.cn

Kecheng Liu
Informatics Research Centre, Henley Business School,
University of Reading,
Reading, RG6 3XA, U.K.
k.liu@reading.ac.uk

Abstract: Currently researchers in the field of personalized recommendations bear little consideration on users' interest differences in resource attributes although resource attribute is usually one of the most important factors in determining user preferences. To solve this problem, the paper builds an evaluation model of user interest based on resource multi-attributes, proposes a modified Pearson-Compatibility multi-attribute group decision-making algorithm, and introduces an algorithm to solve the recommendation problem of k-neighbor similar users. This study addresses the issues on preference differences of similar users, incomplete values, and advanced converge of the algorithm, and realizes multi-attribute collaborative filtering. The algorithm is proved to be effective by an experiment of collaborative recommendation among multi-users in a virtual environment. The experimental results show that the algorithm has a high accuracy on predicting target users' attribute preferences and has a strong anti-interference ability on deviation and incomplete values.
Keywords: personalized recommendation, group decision-making, multi-attribute, collaborative filtering, Pearson-Compatibility.

1 Introduction

A recommender system aims to generate meaningful recommendations to users for items or products that might interest them [1]. In many markets, consumers face a wealth of products and information from which they can make choices. To alleviate this problem, many web sites attempt to help users by incorporating a recommender system that provides users with a list of items and/or web pages that are likely to interest them. There are real-world operations of industrial recommender systems, such as the recommendations for books on Amazon, or movies on Netflix.

As one of the most successful approaches to building recommender systems, collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users [2]. The developers of one of the first recommender systems, Tapestry [3] coined the phrase "collaborative filtering (CF)", which has been widely adopted regardless of the facts that recommenders may not explicitly collaborate

Copyright © 2006-2015 by CCC Publications

with recipients and recommendations may suggest particularly interesting items, in addition to indicating those that should be filtered out [4]. The fundamental assumption of CF is that if users X and Y rate n items similarly, or have similar behaviors (e.g., buying, watching, and listening), and hence will rate or act on other items similarly [5].

Studies in literature [3] and [4] have shown that users' interest in a product or service is affected by user topic preferences, content preferences, user habits, public evaluation and other factors, and that these factors are decided by the different attributes of items. For example, users liking a new movie may be caused by one or more attributes of the movie, such as the director, star, theme, content, style, public comment and so forth. Thus, in the application of collaborative filtering algorithm, it is necessary to use a multi-attribute analysis model, in which the user rating to an item is based on a different perspective (attributes) to describe their interest preferences.

Although user and resource clustering based on the attribute information has been widely discussed in the multi-attribute collaborative filtering literatures, the recommend method is still traditional [6]. Such methods can only obtain a set of potential interest items of target users, but the reasons of such a recommendation are not given to the target user. In addition, the present studies scarcely consider the characteristics differences of similar users interested in the item attributes, which can lead to recommendation deviation [6]. For example, in a traditional way, user B is the most similar one to the target user A because A and B have the same degree of interest in the same film. However, if the film properties they prefer are completely different, it will lead to recommendation deviation when we give greater weight to B.

Based on our previous research, we propose that multi-attribute collaborative filtering can be treated as a group decision making process. By building the rating matrix of target items for the similar users, we remove the user who has a large attribute preference difference to target user from the nearest user set, and save the problem of recommendations deviation. In addition, we can analyze the user's interest performance from the view of item attributes and give the descriptions for the recommendation. Accordingly, this paper proposes a modified Pearson-Compatibility multi-attribute group decision-making algorithm and introduces the algorithm to solve the recommendation problem of k-neighbor similar users.

This paper has extended our previous research significantly with much more details of the theoretical model, such as the characters of the multi-attribute evaluation, the selection of the nearest neighbor of target user, and the group decision-making model of personalized recommendation. More importantly, we have enriched and refined the collaborative filtering recommendation algorithm which is the core of this paper.

This paper is organized as follows. We review recommender systems and multi-attribute utility theory. We then introduce the establishment of user interest model using applied ontology method to describe user profile and illustrate the algorithm in specific steps. Finally, we do an experiment and conclude with findings and discussions.

2 Descriptions of Basic Model

2.1 User rating matrix

A user's comment on a certain item is usually an integration of multi-attribute comments made from different angles [6]. Suppose an item is shown as follows:

$$P = \{a_1, a_2, a_3, \ldots a_n\}$$

Based on the revised rating model, the paper establishes the user rating matrix. Suppose the user set is denoted as $U = \{U_1, U_2, \ldots U_p\}$ and the user U_j rating for item P_i is denoted as

$A(U_j, P_i)$:

$$A(U_j,P_i)=\begin{array}{c} \begin{matrix} a_1 & a_2 & a_3 & & a_{n-1} & a_n \end{matrix} \\ \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \dots & \omega_{1n-1} & \omega_{1n} \\ \omega_{21} & \omega_{22} & \omega_{23} & \dots & \omega_{2n-1} & \omega_{2n} \\ \omega_{31} & \omega_{32} & \omega_{33} & \dots & \omega_{3n-1} & \omega_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega_{(n-1)1} & \omega_{(n-1)2} & \omega_{(n-1)3} & \dots & \omega_{(n-1)(n-1)} & \omega_{(n-1)n} \\ \omega_{n1} & \omega_{n2} & \omega_{n3} & \dots & \omega_{n(n-1)} & \omega_{nn} \end{bmatrix} \end{array} \begin{matrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_{n-1} \\ a_n \end{matrix}$$

Where ω_{xy} is the importance of attribute a_x of product P_i in comparison with attribute a_y for user U_j. Here we use the 1-9 scale Paired comparison method to analyze the compared importance level of each attribute of the product that a user evaluates [7]. The rating matrix of an item is mainly acquired through user scoring, or acquired through user behavior analysis, or acquired with the approaches of Web semantic digging and fuzzy mathematics [6].

The present user rating system, such as Movielens, only asks a user to make a synthetic rating for the movie he or she watched and give a quantitative scoring between 1 and 5. This approach is not accurate in identifying the similarity in the preference of two users. For example, if two users are interested in the same movie. When they rate it, they give it the same score. However, the angles of their preference for the movie are totally different. A user may like the star and the style. The other user prefers the theme and the content of the movie. Therefore, we propose to build a multi-attribute rating system to evaluate a product, i.e., a product has many attributes. When evaluating the product, a user mainly gives the preference ratings of each product attribute.

Another notable character of the multi-attribute evaluation is the absolute sparsity of the judgment matrix. This is caused by users who only know a few attributes of the product. This matrix is like the incomplete value judgment matrix of group-decision making, which easily lead to obvious deviations for the prediction results. So we should take some adjusted measures to solve the problem.

2.2 User Interest Model

Suppose user U_j has rated several items and the rating matrix set is $AS = \{A(U_j, P_1), A(U_j, P_2), \dots A(U_j, P_t)\}$, where $A(U_j, P_i)(i = 1, 2, \dots t)$ is user U_j rating matrix for product i (i.e., P_i). This paper applies the rating matrix set to establishing the user interest model [6]. The specific steps are as follows.

1. Calculating the feature weight vector of each rating matrix, and then acquire the feature weight vector set

$$\begin{aligned} VS =& \{V_{U_j}^{P_1}(w_1, w_2, w_3, \dots w_{size(A(U_j,P_1))}), V_{U_j}^{P_2}(w_1, w_2, w_3, \dots w_{size(A(U_j,P_2))}) \dots \\ & V_{U_j}^{P_t}(w_1, w_2, w_3, \dots w_{size(A(U_j,P_t))})\}. \end{aligned}$$

   Where $V_{U_j}^{P_i}(w_1, w_2, w_3, \dots w_{size(A(U_j,P_i))})$ denotes the feature weight vector of the user rating matrix $A(U_j, P_i)(i = 1, 2, \dots t)$ and $size(A(U_j, P_i))$ denotes the length of the feature weight vector.

2. According to the category of each attribute, calculate the user interest weights of the relavant attribute in the related resource category. Referring to the methods proposed

by [8] and [9], we propose the following formula for calculating the degree of the user interest.

$$Va(U_j, a_y, n) = \frac{\sum_{k=1}^{n} A(U_j, P_k) \times V_{U_j}^{P_k}(w_y)}{n} \quad (1)$$

Where $Va(U_j, a_y, n)$ denotes the degree to which user U_j is interested in attribute a_y. n is the number of the items which has attribute a_y and user U_j have rated. $A(U_j, P_k) \times V_{U_j}^{P_k}(w_y)$ $(k = 1, 2, 3, \ldots n)$ denotes the degree of user U_j interest in attribute a_y of product P_k, which indicates how user U_j preference on item P_k is mostly determined by attribute a_y.

2.3 Selection of the Nearest Neighbor of Target User

We define the target user as the online user which requires evaluations and preliminary recommendations. The set of nearest neighbors is composed of the users who have the most similar interest and preference to item with the target user.

Traditionally, researchers use k-Nearest Neighbor (KNN) algorithm and Pearson correlation-based similarity formula to do cluster analysis on the target user and the similar users, according to their similar interest and preference. Through these methods, the similar user set with different group standards can be obtained. In the process of collaborative filtering recommendation with group decision-making method, we consider the characteristics of the target user preference and search the similar user sets. The specific steps are as follows.

Step 1. Search the user set which has similar interest distribution with target user U_T, i.e., to obtain the intersection set of the interest distribution of all users and the target user. Two users can rate the same attributes in a number of categories and have similar interest weights. This approach can be applied to obtain the initial Nearest Neighbor set:

$$IU = \{(U_1, \Omega_1), (U_2, \Omega_2), (U_3, \Omega_3) \ldots, (U_w, \Omega_w)\}$$

Where w is the number of users totally in the initial Nearest Neighbor set. u_k denotes the k^{th} user whose interest set is S_k. Ω_k denotes the interest intersection between user U_k and the target user U_T.

Step 2. Use Pearson correlation-based similarity formula to calculate the similarity between target user U_T and a random user U_k. When the degree of similarity reaches the threshold, u_k can be divided into S_{U_T} which is the similar user set of U_T. Finally, $S_{u_T} = \{u_1, u_2, \ldots u_s\}$ is obtained, i.e., s similar users meet the threshold. The set of the interest intersection between the target user u_T and s similar users that meet the threshold is $Se_{u_T} = \Omega_1 \cup \Omega_2 \cup \Omega_3 \ldots\ldots \cup \Omega_s$.

3 Collaborative Filtering Recommendation Algorithm Based on Multi-attribute Group Decision-making

3.1 Group Decision-making Model of Personalized Recommendation

After acquiring a similar user set S_{U_T}, we need predict and recommend the items that target user U_T has not commented yet. Suppose the item set is $Source = \{P_1, P_2, S_3, \ldots P_n\}$ and P_k $(k = 1, 2, \ldots n.)$ is the item that the target user has not commented yet. The traditional collaborative filtering recommendation algorithm is applied to calculate the overall evaluation value of the item $A(U_T, P_k)$ given by similar users and then obtain the initial recommended source $S_{Initial}$. Suppose a random item $P_k \in S_{Initial}$ has n attributes denoted as a set $Sp_k =$

$\{a_1, a_2, \ldots a_n\}$, the set satisfies the condition $Sp_k \subseteq Se_{u_T}$. The comment matrix is denoted as $A^1, A^2, A^3, \ldots A^p$, which means similar users in S_{u_T} commend P_k. We suppose p users are similar with the target user and have made comments to the item. The algorithm proposed in this paper requires that $p >= 3$. When $p < 3$, refer to the article [10].

After obtaining the evaluation matrix $A^1, A^2, A^3, \ldots A^p$, it vital for this research to use the information of comments to get preference matrix of the target users. The collaborative filtering recommendation of multi-attribute similar users is a typical group decision problem, in which many similar users make their decisions independently without any discussions and then the computer synthesizes their opinions to make recommendations to the target user. The result requires that all similar users' comprehensive evaluation matrixes and comprehensive characteristics weight vector $\{w_1, w_2, w_3, \ldots w_n\}$ be calculated. w_i denotes the comprehensive preference of the similar users to attribute a_i, which is compared with other attributes of the product that has not been recommended. According to comprehensive characteristics weight vector, we can know which attribute determines the users' interest on a specific item. Applying the semantic analysis method and the semi-structured description language to the attributes, we can make better recommendation to users.

A recommendation model can be transferred to group decision-making model in order to solve a problem. However, the traditional group decision-making algorithm still need be improved when applied since there exists differences between similar users and decision-making expert. The differences mainly exist in the following aspects:

1. It is hard to use the weight vectors to measure the influence of similar users in the recommendation processes mainly because of the complexity of user preferences. According to the traditional method, two users have a high similarity in their interest preferences. However, it is hard to identify the deviation existing in the preferences of the two users on a specific item.

2. There are a large number of incomplete values in user comment information because some users may make no comments on the unfamiliar attributes to ensure the accuracy of evaluations. In this situation, the application of some traditional group decision-making algorithms, such as weighted average method or weighted least squared logarithm method may result in a great deviation of the final result.

To solve the two problems mentioned above, this paper makes improvements in group decision-making compatibility test algorithm and builds a collaborative correction algorithm based on Pearson-Compatibility model. The core of this algorithm is to simulate the expert group-decision making process, discover the common opinions of most experts via data analysis, and revise the opinions of each expert to realize the final consistent compatibility. Moreover, to solve the problem of premature convergence, this paper adopts a mode of multi-user collaborative correction. In addition to the usual algorithms, when it calculates the value of each user's impact, this paper uses the common opinion of all users rather than target user's opinion as a measurable index. The user whose opinion is more similar to the common opinion would be given a higher weight, which is different to traditional method.

3.2 The Related Definitions

Firstly, we introduce the calculation of the value of user impact weight. This value is an important indicator to measure the degree of evaluation information consistency between a user and the others [6]. The user matrix with higher group evaluation consistency will get higher

weight. Vise versa. This paper adopts the concept of user rating similarity [11, 12]. We turn all the similar $n \times n$ user rating matrixes into $n^2 \times 1$ one dimensional vector. The user U_k judgment matrix A^k could be denoted as $V^k = \{\omega_{11}^k, \omega_{12}^k, \omega_{13}^k \ldots \omega_{1n}^k, \omega_{21}^k, \omega_{22}^k, \omega_{23}^k \ldots \omega_{2n}^k \ldots \omega_{n1}^k \ldots\ldots \omega_{nn}^k\}$. Pearson similarity formula to calculate the rating matrix between the user U_k and the user U_l is show as follows:

$$Si(A^k, A^l) = \frac{\sum_{i=1}^{n^2} (V^k(i) - \overline{V^k}) \times (V^l(i) - \overline{V^l})}{\sqrt{\sum_{u=1}^{n^2} [V^k(i) - \overline{V^k}]^2} \times \sqrt{\sum_{u=1}^{n^2} [V^l(i) - \overline{V^l}]^2}} \tag{2}$$

$\overline{V^k}$ is the average value of all elements of user U_k rating matrix.

$$\overline{V^k} = \frac{\omega_{11}^k + \omega_{12}^k + \ldots + \omega_{nn}^k}{n^2}.$$

The similarity between user k and other users could be calculated as follows:

$$Si_k = \sum_{l=1, l \neq k}^{p} Si(A^k, A^l)/(p-1).$$

where p denotes the number of users. We propose a formula $D_k = 1 - Si_k$ as the approximate measure of variance, which indicates the deviation degree of evaluation matrix. The approximate influence weight of user U_k is shown as follows:

$$\theta_k = \frac{(1 - \max\{Si_l, l = 1, 2, \ldots p\})^2}{D_k^2} \tag{3}$$

After acquiring the similar user influence weight, we suppose the group integrated approximate evaluation matrix of p users is A^*, and the value of each element ω_{ij}^* in matrix A^* is as following:

$$\omega_{ij}^* = \sum_{k=1}^{p} \theta_k \times \omega_{ij}^k / \sum_{k=1}^{p} \theta_k \tag{4}$$

Matrix A^* is not a positive reciprocal matrix. Suppose X is a positive reciprocal matrix composed of x_{ij}. This paper uses the least square method to modify X and propose the following formula:

$$F(X) = \min \sum_{i=1}^{n} \sum_{j=1}^{n} (x_{ij} - \omega_{ij}^*)^2$$

$$s.t \begin{cases} x_{ij} \times x_{ji} = 1 \\ x_{ij} > 0 \qquad\qquad (i, j = 1, 2, \ldots n) \end{cases} \tag{5}$$

The definition of compatibility and comprehensive compatibility are as follows:

Definition 1. Suppose X is the group user comprehensive evaluation matrix obtained by using the method of the least squares. Then the judgment matrix compatibility between user k and the other users is as follows:

$$S(A^k, X) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\omega_{ij}^{(k)} \times x_{ij}}{\max((\omega_{ij}^{(k)})^2, (x_{ij})^2)}}{n^2 - \alpha} \tag{6}$$

Although paper [13] has defined expert judgment matrix compatibility in usual cases, it does not consider the incomplete value. Formula (6) is a modified approach to solve the problem. Firstly, the block that the user does not rate is processed and given the value 0. Then α is used to indicate the number of 0. The aim of this approach is to eliminate the influence of user judgment matrix on compatibility indicator.

Definition 2. Suppose $A^{1\prime}$, $A^{2\prime} \ldots A^{p\prime}$ are the compatibility correction of matrixes of p users' judgment matrixes. Then we get the comprehensive consistency indicator $\bar{S}$, as follows:

$$\bar{S} = \frac{\sum_{k=1}^{p} S(A, A^{k\prime})}{p} \tag{7}$$

Readers can refer to the simulation result of article [7]. When $S(A, B) >= 0.8$, the two evaluation matrixes is considered nearly compatible. When $\bar{S} \geq 0.8$, evaluation matrixes of all p similar users is considered compatible.

3.3 Collaborative Correction Algorithm Based On Pearson-Compatibility

Pearson-Compatibility model could be used to simulate the process of experts doing group discussions and finally making group decisions. The model mainly consists of two indicator calculation formulae: Pearson similarity calculation and Compatibility test. Pearson similarity calculation formula is mainly used to calculate the information reservation degree of user rating matrix after the matrix has been revised. And Compatibility test is used to test degree of consistency of user rating matrix after matrixes has been revised. The mainly function of Pearson-Compatibility model is to build the associations between two variational indicators. Based on this association, Compatibility correction algorithm will choose the best revised matrix in each iteration process, which could make the user matrixes unanimous and most similar to the true value.

Referring to [14], based on the related method [15, 16], this paper proposes a Pearson-Compatibility model as follows:

$$\Omega^{(t)} = \overline{Si^{(t)}}^{1-\beta} \times \overline{S^{(t)}}^{1+\beta} \tag{8}$$

Where $\Omega^{(t)}$ denotes the calculation result of Pearson-Compatibility model after t times iteration. With all the continuous revision of user matrixes, $\overline{Si^{(t)}}$ decreases from 1 to 0, and $\overline{S^{(t)}}$ increases from 0 to 1. So each time of modification generates a better revised scheme which minimizes the change of $\overline{Si^{(t)}}$ while maximizes the change of $\overline{S^{(t)}}$. This indicates that all the experts acquires a better compatibility with the least possible original information lose. This method is similar to the real decision processes [16] and can provide an effective way to approach the true value. In Pearson-Compatibility model, the best revised scheme in each iteration process is the one that maximizes $\Omega^{(t)}$. β is used to control the marginal diminishing rate and the marginal increasing rate of $\overline{Si^{(t)}}$ and $\overline{S^{(t)}}$. The efficiency of this algorithm is low when the consistency of the evaluation matrixes is low. Moreover, it is possible that the algorithm converges in advance, i.e., $\Omega^{(t)}$ reaches the max value at the beginning of calculation. At that time, if the value of β increases, the efficiency of the algorithm can be improved and simultaneously solve the problem of pre-mature convergence to some degree.

Regarding the designation and realization of compatibility correction algorithm based on Pearson-Compatibility, readers could refer to [17, 18]. In order to solve the problem of pre-mature convergence, this paper introduces a multi-user collaborative correction mechanism which

is based on Simulated Annealing Algorithm (SAA). The following experiments in this paper testify that this method can solve the problem of pre-mature convergence in advance, and further improves the accuracy of the experiment result. This algorithm is described as follows:

Step1: Suppose u is the iteration times of compatibility test. zs is the number of users whose evaluation matrix could not be revised further in each step, which is given the original value 0. Suppose original user matrixes of all p users are $A^{1(0)}, A^{2(0)} \ldots\ldots A^{p(0)}$. Then after u times compatibility correction iteration, p users evaluation matrixes are: $A^{1(u)}, A^{2(u)} \ldots\ldots A^{p(u)}$

Step2: Use formula (2) and (3) to calculate the influence weights of the evaluation matrixes of p users.

Step3: Use formula (4) and (5) to calculate the user comprehensive rating matrix $X^{(u)}$ after the u^{th} compatibility correction.

Step4: Use formula (6) to calculate the Compatibility between the k^{th} user $A^{k(u)}$ and $X^{(u)}$. Then apply formula (7) to calculate the comprehensive consistency degree of $\bar{S}^{(u)}$.

Step5: When $zs = p$ and $\bar{S}^{(u)} < 0.8$, go to Step7. When $zs \neq p$ and $\bar{S}^{(u)} < 0.8$, go to Step6. When $zs \neq p$ and $\bar{S}^{(u)} \geq 0.8$, we believe p users comprehensive consistency could pass the test, go on with Step8.

Step6: Referring to [17], we can revise $X^{(u)}$ and $A^{k(u)}$. Suppose the deviation matrix of $A^{k(u)}$ is $D^{k(u)} = \{e_{ij}^{k(u)}\}$, where $e_{ij}^{k(u)} = x_{ij}^{(u)} - \omega_{ij}^{k(u)}$. Suppose $e_{wr}^{k(u)} = \max(|x_{ij}^{(u)} - \omega_{ij}^{k(u)}|)$ is the maximum deviation item after u times iteration, where $\omega_{ij}^{k(u)}$ is nonzero term. Let $u = u + 1$ and begin a new round of revision. The revised rating matrix is:

$$A^{k(u)} = \begin{cases} x_{ij}^{(u-1)} - e_{ij}^{k(u-1)} \times 0.95^{\mu}, & \text{when } i = w, j = r \\ 1/[x_{ij}^{(u-1)} - e_{ij}^{k(u-1)} \times 0.95^{\mu}], & \text{when } i = r, j = w \\ x_{ij}^{(u-1)} - e_{ij}^{k(u-1)}, & \text{when } i \neq w, i \neq r, j \neq r, j \neq w \end{cases} \tag{9}$$

where u is the step length adjusting factor. This paper gives a limitation between 0 and 10 to u. Thus, the range of 0.95^{μ} is [0.5987, 1]. The larger u is, the faster the rate of convergence of rating matrix of user k will be. In order to prevent an oversized adjustment of a single user matrix and thus lose the information in the initial rating matrix, the maximum adjustment extent is set as 0.5987.

If u is given an appropriate value in each iteration, the calculating speed and the accuracy of the algorithm result will be highly improved. However, the existing algorithm is not effective enough. This paper proposes an approach of building a multi-user collaborative filtering model based on simulated annealing algorithm (SAA), i.e., applying SAA to revise the maximum deviation item in each iteration and calculate the optimal correction factor μ^k for each user $k(k = 1, 2 \ldots p)$.

In the simulated annealing algorithm (SAA), each point s of the search space is analogous to a state of some physical system. The goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy. According to the principle of Metropolis, the probability of solid particle at temperature T changed from the disordered state to the steady state is $\exp(-\frac{\Delta E}{k \times T})$. ΔE is the internal energy change quantity when the solid temperature change to T and k means the Boltzm-ann constant. When we apply SAA method to solve the

combinational optimization problem, internal energy E means the value of target function and temperature T means the control factor. Then we can get the simulated annealing algorithm.

In the SAA, we suppose the initial energy value of target function in the model is E and control factor is T. Then we can control the attenuation amplitude in each iteration and use the random function to simulate the random motion of particle. And we calculate the objective function difference as ΔE, and according to Metropolis principle to choose the result. When T decrease to the critical value or the model solution could not be better, we stop this algorithm and get the approximate optimal result.

The SAA applied to the multi-user collaborative correction algorithm could be described as following:

Step6.1: Suppose it is the u^{th} iteration. Let formula (8) be the energy function. Suppose $c = 0$ is the number of iterations for calculating μ^k, and $sc = 0$ is the verdict factor to judge whether to stop the iteration.

Step6.2: For user $A^{k(u)}, (k = 1, 2, \dots p)$, apply formula (9) to confirm the maximal deviation item and to calculate the evaluation set $A_c^{1(u)}, A_c^{2(u)}, A_c^{3(u)}, \dots A_c^{p(u)}$. Randomly choose the initial state set $[\mu_c^1, \mu_c^2, \dots \mu_c^p]$ between 0 and 1.

Step6.3: Run Matlab software to generate p random numbers $\Delta_1, \Delta_2, \dots \Delta_p$, which are between 0 and 1. Suppose $\mu_c^k = \Delta_k$ ($k = 1, 2 \dots p$, and Δ_k in each iteration is different). Let $c = c + 1$ and calculate the energy function $\Omega_c^{(u)}$. Suppose T is the temperature schedule. Its initial value $T_0 = 100$. The attenuation function is $T_c = T_{c-1} \times 0.95$. Suppose the counting variable is rt and $rt = rt + 1$. Let $\Delta f = \Omega_c^{(u)} - \Omega_{c-1}^{(u)}$. Then we make the following judgment:

When $\Delta f > 0$, $\mu_c^1, \mu_c^2, \dots \mu_c^p$ is accepted as the new state solution and go on with **Step4**.

When $\Delta f < 0$, there are two possibilities. If $e^{(\frac{\Delta f}{T_c})} > Random[0, 1]$ holds, $\mu_c^1, \mu_c^2, \dots \mu_c^p$ is accepted as the new state solution and go on with **Step4**. Otherwise, let $\mu_c^k = \mu_{c-1}^k$, $sc = sc + 1$.

Step6.4: When $sc > 10$ or $T_c < 0.01$, which means we reach the end condition and export the result $\mu_c^1, \mu_c^2, \dots \mu_c^p$. Otherwise, go on with **Step2** after the value of u is obtained.

Step7: Popup dialog prompt whether to set a new threshold S_{xy} which means the user acceptable level. If the user choose to reset, zs value would be zero. And we will let the value of Pearson in each loop iteration compared with S_{xy} and get the value of zs. Then we execute Step5. If the user choose not reset, we execute **Step8.**

Step8: Export the calculation result $X^{(u)}$, $\overline{S}^{(u)}$ and p user final evaluation matrixes $A^{k(u)}, (k = 1, 2, \dots p)$.

We have testify the algorithm is effective in the previous studies. If you are interested in the test process, you can refer to article [16] and [17].

4 Experimentation

We build an experiment environment to execute our algorithm at current conditions to validate the effectiveness of this algorithm [6]. The environment is described as follows:

We adopt ontology and the relevant methods in order to design and develop the movie information database. Jena 2.6.2 is applied to store the movie information in RDF format and ARQ-2.2 is used to manage the movie information. We have imported 300 movies which involve 10 categories. A semantic analysis of each movie is conducted to get key words and form the initial attribute set. Then the synonyms and the similar words in the initial set are combined. Take some topical words as the characteristic attributes and use them to represent

these movies. Finally, 15 attributive categories and 282 concrete attributes are extracted. Then an online multi-attribute rating system based on the movie database and a collaborative filtering recommendation system based on group-decision making are designed and developed.

The concrete process that tests the algorithm is as follows [6]:

1. Select four evaluated movies in which $G(u, p)$ is comparatively big and use them as the testify set. They respectively include 6, 7, 8 and 9 attributes. Then, use the target user evaluation matrixes which are further used as the real weight vectors to calculate the user interest vectors for each movie.

2. Based on the user-evaluated movies set (excluding the 4 movies in the test set), apply the methods in sections 2.3 and 2.4 to searching the most similar user set for the target user (i.e., the similar interest distributions).

Take Movie 1 with 6 attributes as an example. The real interest vectors are S = [3.7288, 2.7053, 1.9627, 0.4657, 0.3293, 0.3293]. The total score of this movie is 4.5 which indicates that the target user has a high preference to this movie. Moreover, the preference is mainly determined by the first three attributes. Totally, 9 similar users have evaluated this movie. Firstly, the traditional collaborative filtering algorithm is applied to obtaining the weighted average of the total score of this movie and gets the result 3.94. We are not sure whether the target users have interests in this movie. Thus, we need use the similar user evaluation matrixes to make judgments. The evaluation matrixes of six similar users are listed as follows:

$$A = \begin{bmatrix} 1 & 2 & 2 & 5 & 7 & 9 \\ 1/2 & 1 & 1 & 6 & 5 & 6 \\ 1/2 & 1 & 1 & 7 & 7 & 6 \\ 1/5 & 1/6 & 1/7 & 1 & 1 & 2 \\ 1/7 & 1/5 & 1/7 & 1 & 1 & 3 \\ 1/9 & 1/6 & 1/6 & 1/2 & 1/3 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 & 3 & 7 & 7 \\ 1/2 & 1 & 0 & 4 & 7 & 6 \\ 1/3 & 0 & 1 & 0 & 4 & 5 \\ 1/3 & 1/4 & 0 & 1 & 1 & 2 \\ 1/7 & 1/7 & 1/4 & 1 & 1 & 3 \\ 1/7 & 1/6 & 1/5 & 1/2 & 1/3 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 2 & 4 & 0 & 0 \\ 1/2 & 1 & 1 & 3 & 0 & 0 \\ 1/2 & 1 & 1 & 2 & 0 & 0 \\ 1/4 & 1/3 & 1/2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 2 & 3 & 4 & 6 & 8 \\ 1/2 & 1 & 2 & 2 & 4 & 6 \\ 1/3 & 1/2 & 1 & 1/2 & 4 & 7 \\ 1/4 & 1/2 & 2 & 1 & 3 & 2 \\ 1/6 & 1/4 & 1/4 & 1/3 & 1 & 2 \\ 1/8 & 1/6 & 1/7 & 1/2 & 1/2 & 1 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 2 & 0 & 7 & 8 & 0 \\ 1/2 & 1 & 2 & 8 & 7 & 7 \\ 0 & 1/2 & 1 & 0 & 7 & 8 \\ 1/7 & 1 & 0 & 1 & 1 & 2 \\ 1/8 & 1/7 & 1/7 & 1 & 1 & 1 \\ 0 & 1/7 & 1/8 & 1/2 & 1 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 2 & 4 & 0 & 7 & 0 \\ 1/2 & 1 & 2 & 6 & 7 & 7 \\ 1/4 & 1/2 & 1 & 0 & 2 & 2 \\ 0 & 1/6 & 0 & 1 & 0 & 2 \\ 1/7 & 1/7 & 1/2 & 0 & 1 & 1 \\ 0 & 1/7 & 1/2 & 1/2 & 1 & 1 \end{bmatrix}$$

3. Apply the four algorithms to calculate the score of the four movies and make comparisons on the deviations of the real weight vectors of the target users. The result is listed as follows:

Table 1: The comparison between algorithms

	Movie 1 (6 order)	Movie 2 (7 order)	Movie 3 (8 order)	Movie 4 (9 order)
Arithmetic weighted average method	0.1589	0.0564	0.1985	0.1132
Logarithmic least squares method	0.1054	0.0534	0.1398	0.0831
Compatibility correction algorithm	0.0877	0.0556	0.1042	0.0687
Our algorithm	0.0780	0.0543	0.0885	0.0683

When the scores of a part of similar users have a large deviation from those of the other users, the algorithm proposed in this paper can solve the problem of early convergence better than the other algorithms and obtain an accurate result, as shown in Table 1. The core of our algorithm is the revised values of the comprehensive evaluation matrix determined by the majority of users. Accordingly, the highly deviated evaluation values are revised. The result of seven order matrix experiment shows that the deviations of the result of any algorithms are not notable when all the similar users have unanimous evaluation matrixes, The result of nine order matrix experiment shows that the result of the proposed algorithm is similar to that of compatibility correction algorithm when all the similar users have unanimous evaluation matrixes, while still have some incomplete values, and is better than the other two algorithms obviously. When there are 5 similar users and six order evaluation matrix is executed with our algorithm, the change tendencies of the main indicators are shown in Figure 1:

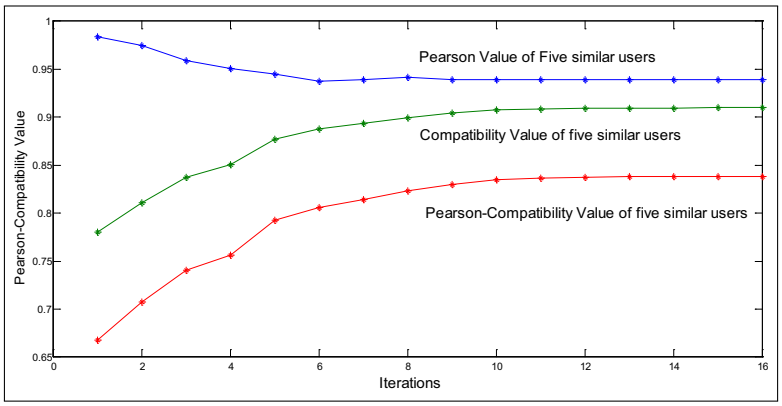


Figure 1: The main indicators change in our example

4. For example, for the first movie with the six order evaluation matrix, the influence of the user number on the accuracy of recommendation results is examined. Suppose the user number is 3, 5, 7 and 9. The accuracy and the number of iterations are calculated with different means of permutation and combination. Part of the result is shown in Table 2:

Table 2: The comparison of different similar user numbers

User number	3	5	7	9
initial Indicator of Comprehensive consistency degree	0.5872	0.6890	0.6872	0.7081
Deviation of result	0.1680	0.1093	0.0828	0.0780
Number of iterations	12	16	28	37

More deviation items are generated as the user number increases. Therefore, the iterations of this algorithm rise. This test indicates that the effectiveness of this algorithm is highly related to the initial consistency degrees of all users and the number of users. In general, when the initial consistency degree is low and the similar user set is limited (e.g. there are 3 users), it is hard for the algorithm to dig out the common information among the users. Therefore, the result deviation is huge. However, when the number of similar users increases to a certain degree (e.g. the number is equal or bigger than 7), the algorithm still remains a good accuracy, even if the initial compatibility is low.

Regarding the provision of personalized services to the target users, this paper calculates the comprehensive evaluation weight vectors of each movie with the group-decision making model.

Take the first movie with 6 attributes as an example. The comprehensive evaluation score of nine similar users is $G(u,p) = 3.94$. The comprehensive evaluation vectors are $V = [4.0653$ 2.9492 1.7630 0.3972 0.3044 0.3607]. Each value of the weight vector represents the potential interest degree of the target user on the corresponding product attributes. Thus, the total score calculation formula is

$$TScore = G(u,p) \times \sum_{i=1}^{n} V_i/n \tag{10}$$

where *TScore* denotes the total score, n denotes the number of attributes, and V_i denotes the comprehensive evaluation value of the (i)th attribute of the product. The recommendation set can be fixed through the way of ranking or threshold setting. The total scores of the four movies is shown in Figure 2:

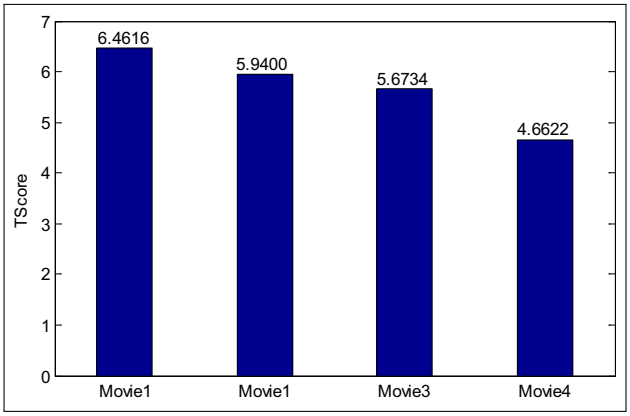


Figure 2: The TScore of four movies

The first movie has the highest score with six attributes, as shown in Figure 2. The characters of this movie are analyzed as follows using the user interest model. Firstly, attribute V_i of weight vector V is normalized and generate vector
$\overline{V} = [0.4131, 0.2997, 0.1792, 0.0404, 0.0309, 0.0367]$. The three attributes whose values are bigger than the average value 0.1666 are picked out. When the attribute value is bigger than 0.1666, the majority of users have evident preference on movie one. In the target user interest model, there are 124 attributes totally. The 3 attributes of movie one that are bigger than the average value are connotation, characteristic and special efficiency are still larger than the average value (1/124 = 0.0081) among 124 target user attribute preferences. This result indicates that target user has evident preference to these 3 attributes and the popularity of this movie is mainly determined by these attributes. Therefore, we could introduce movie one to target user and provide the reasons why this movie is introduced. We also could use semantic analysis technique to describe each attribute in detail and provide more personal service to target user. The comparison analysis histogram is shown in Figure 3.

M bar means the percentages which attributes are ranking at top 3 in movie one, U bar means percentages which attributes are ranking at top 3 in target user interest model. The analysis show us that target user may have larger interest preference to attribute one in movie one.

5 Conclusion

The traditional collaborative filtering personal recommended algorithms seldom consider the multi-attribute problem. Our approach is based on group-decision making. We propose an

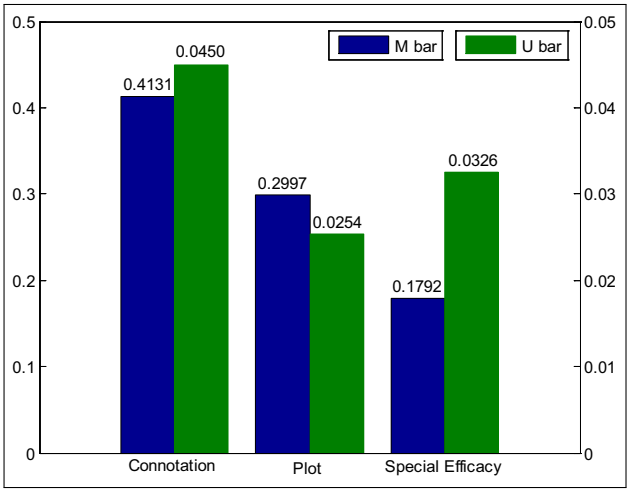


Figure 3: Comparison analysis histogram of movie one with target user interest.

improved Pearson-Compatibility algorithm which is applied to the collaborative filtering recommend field. We then build a virtual recommend environment and testify the effectiveness and feasibility of this algorithm. The advantages of the collaborative filtering personal recommended algorithm based on group-decision include:

Identifying a more suited similar users set for the target user. An accurate target user model could be set up via field subdivision according to field attributes. Then the users who have similar interest distribution with target user can be found. The similar user set is generated.

Providing more accurate and personal recommend service to the target user. The traditional collaborative filtering method could neither recommend a result set to target user, nor provide analysis service [6]. This weakness is overcome by making an information integration to know what are mainly factors determining the user preference, so that we could handle the user need more accurate.

Considering evaluation deviation between the similar users and revising the user evaluation. Instead of weighted mean, group-decision making method is used to calculate the comprehensive evaluation score. Deleting the deviation item and revising the evaluation matrix could make the result have a better fitting effect [6]. The collaborative filtering method based on Pearson-Compatibility is applied to the personal recommended field. The result of the experiment shows that the algorithm is stable when dealing with the deviation items and identifies the common preference information between similar users.

Acknowledgment

This research work is supported by Innovation Program of Shanghai Municipal Education Commission (No. 12ZZ070, No. 12YS018) and Humanity and Social Science Youth Foundation of Ministry of Education, China (No. 12YJC630136).

Bibliography

[1] G. Adomavicius, A. Tuzhilin (2005); Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734-749.

[2] Xiaoyuan Su, Taghi M. Khoshgoftaar(2009); A survey of collaborative filtering techniques, *Advances in Artificial Intelligence*, 2009:1-19.

[3] D. Goldberg, D. Nichols, B. M. Oki, D. Terry(1992); Using collaborative filtering to weave an information tapestry, *Communications of ACM*, 35(12):61-70.

[4] P. Resnick, H. R. Varian(1997); Recommender systems, *Communications of the ACM*, 40(3):56-58.

[5] K. Goldberg, T. Roeder, D. Gupta, C. Perkins (2001); Eigentaste: a constant time collaborative filtering algorithm, *Information Retrieval*, 4(2):133-151.

[6] C. Yu, Y. Luo, K. Liu(2014); A multi-attribute collaborative filtering recommendation algorithm based on improved group decision-making, *ICISO 2014, IFIP AICT*, 426:320-330.

[7] F. Herrera, E. Herrera-Viedma, F. Chiclana (2001); Multiperson decision-making based in multiplicative preference relation, *European J. of Operational Research*, 129:372-385.

[8] A. Shepitsen, J. Gemmell, B. Mobasher, R. Burke (2008); Personalized recommendation in social tagging systems using hierarchical clustering, *Proc. of the 2008 ACM conference on Recommender systems*, Lausanne, Switzerland.

[9] A. Ypma, T. Heskes (2002); Categorization of Web pages and user clustering with mixtures of hidden Markov models, *Proc. of the WEBKDD 2002 Workshop: Web Mining for Usage Patterns and User Profiles*, SIGKDD 2002, Edmonton, Alberta, Canada.

[10] Z.S. Hua, B.G. Gong, X.Y. Xu (2008); A DS–AHP approach for multi-attribute decision making problem with incomplete information, *Expert Systems with Applications*, 34(3):2221-2227.

[11] H. Jeon, T. Kim, J. Choi (2010); Personalized information retrieval by usingadaptive user profiling and collaborative filtering, *AISS: Advances in Information Sciences and Service Sciences*, 2(4):134-142.

[12] Liu, A., Yang, Z.(2010); Watching, thinking, reacting: a human-centered framework formovie content analysis, *International Journal of Digital Content Technology and its Applications*, 4(5):23-37.

[13] Y. Dong , Y. Chen, S. Wang (2004); Algorithm of solving weights for group decision making by improving compatibility, *Systems Engineering-theory & Practice*, 2004-10.

[14] C.-S. Yu (2002); A GP-AHP method for solving group decision-making fuzzy AHP problems, *Computers & Operations Research*, 29(14): 1969-2001.

[15] Y. Zhang, H. Wang (2002); Development and application of P-S aided decision system, *Systems Engineering-Theory Methodology Application*, 2002-04.

[16] L. Liang, L. Xiong, G. Wang (2004); A new method of determining the reliability of decision-makers in group decision, *Systems Engineering*, 22(6):91-94.

[17] J. Barzilai, F. A. Lootsma (1994); Power relations and group aggregation in the multiplicative AHP and SMART, *Proc. 3rd Int. Symp. AHP*, 157-168.

[18] J. Sun, W. Xu, Q.D. Wu(2005); A new algorithm for incomplete matrixes' compatibility improvement and group ranking in group decision making, *Systems Engineering-Theory & Practice*, 10(10):89-94.

Author index