Advances in Intelligent Systems and Computing 474

Sigeru Omatu
Ali Selamat
Grzegorz Bocewicz
Paweł Sitek
Izabela E. Nielsen
Julián A. García García
Javier Bajo *Editors*

Distributed Computing and Artificial Intelligence, 13th International Conference

Springer

Advances in Intelligent Systems and Computing

Volume 474

About this Series

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Sigeru Omatu · Ali Selamat
Grzegorz Bocewicz · Paweł Sitek
Izabela Nielsen · Julián A. García-García
Javier Bajo
Editors

Distributed Computing and Artificial Intelligence, 13th International Conference

Springer

Editors

Sigeru Omatu
Faculty of Engineering
Osaka Institute of Technology
5-16-1 Omiya, Asahi-ku, Osaka, 535-8585
Japan

Ali Selamat
Department of Software Engineering
Universiti Teknologi Malaysia (UTM)
81310 Johor Baharu, Johor
Malaysia

Grzegorz Bocewicz
Department of Computer Science
and Management
Koszalin University of Technology
Śniadeckich 2, 74-453 Koszalin
Poland

Paweł Sitek
Department of Computer Science
and Management
Koszalin University of Technology
Śniadeckich 2, 74-453 Koszalin
Poland

Izabela Nielsen
Department of Mechanical
and Manufacturing Engineering
Aalborg University
Fibigerstræde 16, Building: 4105
9220, Aalborg Ø
Denmark

Julián A. García-García
E.T.S. Ingeniera Informática
Universidad de Sevilla
Av. Reina Mercedes S/N
41012
Sevilla

Javier Bajo
Department of Artificial Intelligence
Technical University of Madrid
Bloque 2, Despacho 2101,
Campus Montegancedo, Boadilla del Monte,
28660, Madrid
Spain

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-40161-4 ISBN 978-3-319-40162-1 (eBook)
DOI 10.1007/978-3-319-40162-1

Library of Congress Control Number: 2016941289

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

The 13th International Conference on Distributed Computing and Artificial Intelligence 2016 is an annual forum that will bring together ideas, projects, lessons, etc. associated with distributed computing and artificial intelligence, and their application in different areas. His meeting will be held in Sevilla (Spain) from 1st to 3rd June, 2016.

Nowadays, most computing systems from personal laptops/computers to cluster/grid/cloud computing systems are available for parallel and distributed computing. Distributed computing performs an increasingly important role in modern signal/data processing, information fusion and electronics engineering (e.g. electronic commerce, mobile communications and wireless devices). Particularly, applying artificial intelligence in distributed environments is becoming an element of high added value and economic potential. Research on Intelligent Distributed Systems has matured during the last decade and many effective applications are now deployed. The artificial intelligence is changing our society. Its application in distributed environments, such as the Internet, electronic commerce, mobile communications, wireless devices, distributed computing, and so on is increasing and is becoming an element of high added value and economic potential, both industrial and research. These technologies are changing constantly as a result of the large research and technical effort being undertaken in both universities and businesses. The exchange of ideas between scientists and technicians from both academic and business areas is essential to facilitate the development of systems that meet the demands of today's society. The technology transfer in this field is still a challenge and for that reason this type of contributions will be specially considered in this symposium. This conference is the forum in which to present application of innovative techniques to complex problems.

This year's technical program will present both high quality and diversity, with contributions in well-established and evolving areas of research. Specifically, 82 papers were submitted to the main track from over 40 different countries, representing a truly "wide area network" of research activity. The DCAI'15 technical program has selected 59 papers, the best papers will be selected for their publication on the the following special issues in journals such as Neurocomputing, Knowledge and Information Systems: An International Journal, Frontiers of Information Technology & Electronic Engineering and the International Journal of Interactive Multimedia and Artificial Intelligence. These special issues will cover extended versions of the most highly regarded works.

Moreover, DCAI'16 Special Sessions have been a very useful tool in order to complement the regular program with new or emerging topics of particular interest to the participating community. Special Sessions that emphasize on multi-disciplinary and transversal aspects, such as *AI-driven methods for Multimodal Networks and Processes Modeling* and *Multi-Agents Macroeconomics* have been especially encouraged and welcome.

We thank the sponsors (IBM, Indra, Fidetia and IEEE SMC Spain), and finally, the Local Organization members and the Program Committee members for their hard work, which was essential for the success of DCAI'16.

June 2016

Sigeru Omatu
Ali Selamat
Grzegorz Bocewicz
Pawel Sitek
Izabela Nielsen
Julián A. García-García
Javier Bajo

The original version of this book was revised.
An erratum can be found at DOI: 10.1007/978-3-319-40162-1_60

Organization

Honorary Chairman

Masataka Inoue	President of Osaka Institute of Technology, Japan

Scientific Committee

Sigeru Omatu (Chairman)	Osaka Institute of Technology, Japan
Ali Selamat	Universiti Teknologi Malaysia, Malaysia
Silvana Aciar	Institute of Computer Science, National University of San Juan, Argentina
Mohd Sharifuddin Ahmad	University of Tenaga Nasional, Malaysa
Azhana Ahmad	National University of Tenaga, Malaysa
Sebastian Ahrndt	University of Technology of Berlin, Germany
Rula Al-Azawi	De-Montfort University, UK
Gustavo Almeida	Federal Institute of Espirito Santo, Brazil
Ana Almeida	Polytechnic of Porto, Potugal
Giner Alor Hernandez	Instituto Tecnologico de Orizaba, Spain
Victor Alves	Universidade do Minho, Portugal
Cesar Analide	University of Minho, Portugal
Luis Antunes	University of Lisbon, Portugal
Fidel Aznar	University of Alicante, Spain
Luis Daniel Azpeitia Herrera	University of Ciudad Juarez, Mexico
Ameni Azzouz	Higher Institute of Management of Tunis, Lab. SOIE, Tunisia
Zbigniew Banaszak	Warsaw University of Technology, Poland

Olfa Belkahla Driss	Higher Institute of Management of Tunis (lab SILK), Tunisia
Óscar Belmonte Fernández	University Jaume I, Spain
Carmen Benavides	University of Leon, Spain
Ahcene Bendjoudi	University Abderrahmane Mira of Béjaïa/ CERIST, Argeria
Enrique J Bernabeu	Polytechnic University of Valencia, Spain
Xiomara Patricia Blanco Valencia	University of Salamanca, Spain
Francisco Blanes	University of Valencia, Spain
Amel Borgi	University of Tunis El Manar, Tunisia
Pierre Borne	University of Lille, France
Lourdes Borrajo	University of Vigo, Spain
Joan Borrell	Autonomous University of Barcelona, Spain
Adel Boukhadra	National high School of Computer Science, Algeria
Francisco Javier Calle	University Carlos III of Madrid, Spain
Rui Camacho	University of Porto, Portugal
Juana Canul Reich	Juarez Autonomous University of Tabasco, Mexico
Javier Carbo	University Carlos III of Madrid, Spain
Margarida Cardoso	University Institute of Lisboa, Potugal
João Carneiro	Polytechnic of Porto, Portugal
Davide Carneiro	University of Minho, Portugal
Carlos Carrascosa	Polytechnic University of Valencia, Spain
Walid Chainbi	University of Sousse/Sousse, Tunisia
Na Chang	Tokyo Institute of Technology, Japan
Camelia Chira	Technology Institute of Castilla and Leon, Spain
Rafael Corchuelo	University of Seville, Spain
David Cortes	Autonomous University of Ciudad Juarez, Mexico
Paulo Cortez	University of Minho, Portugal
Ângelo Costa	University of Minho, Portugal
Stefania Costantini	University Of L' Aquila, Italy
Dolores Cuadra	University Carlos III of Madrid, Spain
Jaml Dargham	universiti Malaysia Sabah, Malaysia
Giovanni De Gasperis	University of L'aquila, Italy
Carlos Alejandro De Luna-Ortega	Autonomous University of Aguascalientes, Mexico
Fernando Diaz	University of Valladolid, Spain
Worawan Diaz Carballo	Thammasat University, Thailand
Youcef Djenouri	laboratoire de recherche en intelligence artificielle _USTHB, Argelia
David Emele	University of Aberdeen, Scotland
Ramon Fabregat	University of Girona, Spain

Johannes Fähndrich	University of Technology of Berlin, Germany
Ana Faria	Polytechnic University of Porto, Portugal
Florentino Fdez-Riverola	University of Vigo, Spain
Alberto Fernandez	University King Juan Carlos of Madrid, Spain
Peter Forbrig	University of Rostock, Germany
Felix Freitag	Polytechnic University of Catalonia, Spain
Ruben Fuentes-Fernandez	University Complutense of Madrid, Spain
Toru Fujinaka	University of Hiroshima, Japan
Svitlana Galeshchuk	University of Lille, France
Francisco Garcia-Sanchez	University of Murcia, Spain
Irina Georgescu	Academy of Economic Studies of Bucarest, Romania
Ana Belén Gil González	University of Salamanca, Spain
Federico Gobbo	University of L'aquila, Italy
Juan Gomez Romero	University of Granada, Spain
Carina Gonzalez	University of La Laguna, Spain
Evelio Gonzalez	University of La Laguna, Spain
Angélica González Arrieta	University of Salamanca, Spain
David Griol	Universidad Carlos III de Madrid, Spain
Francisco D. Guillen-Gamez	Open University of Madrid (UDIMA), Madrid, Spain
Felipe Hernández Perlines	University of Castilla-La Mancha, Spain
Elisa Huzita	State University of Maringa, Brazil
Boutheina Jlifi	Higher Institute of Management of Tunis, Lab. SOIE, Tunisia
Gustavo Isaza	University of Caldas, Colombia
Patricia Jimenez	University of Huelva, Spain
Bo Noerregaard Joergensen	University of Southern Denmark, Denmark
Vicente Julian	Polytechnic University of Valencia, Spain
Samer Hassan	University Complutense of Madrid, Spain
Amin Khan	Polytechnic University of Catalonia, Spain
Naoufel Khayati	LI3 - ENSI, Tunisia
Egons Lavendelis	Riga Technical University, Letonia
Rosalia Laza	University of Vigo, Spain
Tiancheng Li	Northwestern Polytechnical University, USA
Johan Lilius	Åbo Akademi University, Finland
Dolores Mª Llidó Escrivá	University Jaume I, Spain
Faraón Llorens-Largo	University of Alicante, Spain
Ivan Lopez-Arevalo	Cinvestav - Tamaulipas, Mexico
Jose J. Lopez-Espin	University Miguel Hernández of Elche, Spain
Antonio A. Lopez-Lorca	Swinburne University of Technology, Australia
Ramdane Maamri	LIRE laboratory UC Constantine, Algelia
Moamin Mahmoud	University of Tenaga Nasional, Malaysa
Renato Maia	State University of Montes Claros, Brazil
Benedita Malheiro	Institute of Porto, Portugal
Eleni Mangina	University College Dublin, Irland

Fabio Marques	University of Aveiro, Portugal
Goreti Marreiros	Polytechnic of Porto, Portugal
Angel Martin Del Rey	University of Salamanca, Spain
Ester Martinez-Martin	Universitat Jaume I, Spain
Philippe Mathieu	University of Lille, France
Sérgio Matos	University of Aveiro, Portugal
Jacopo Mauro	University of Bologna, Italy
Amaia Mendez	University of Deusto, Spain
José Ramón Méndez Reboredo	University of Vigo, Spain
Lourdes Mariani	Technical University of Aguascalientes, Mexico
Mohamed Arezki Mellal	M'Hamed Bogara University
Fethi Mguis	Faculty of Sciences of Gabes, Tunisia
Seyedsaeid Mirkamali	Payam Noor University, Iran
Heman Mohabeer	University of Mauritius, Africa
Mohd Saberi Mohamad	Technology University of Malaysia, Malaysa
Felicitas Mokom	Catholic University Institute of Buea, Cameroon
Jose M. Molina	University Carlos III of Madrid, Spain
Miguel Molina-Solana	University of Granada, Spain
Hugo Morais	ISEP/IPP Polytechnic of Porto, Portugal
Jorge Morais	Portuguese Open University, Portugal
Victor Morales Rocha	Institute of Engineering and Technology, Mexico
Antonio Moreno	University of Tarragona, Spain
Paulo Mourao	University of Minho, Portugal
Muhammad Marwan Muhammad Fuad	University of Tromsø, Norway
Eduardo Munera	Polytechnic University of Valencia, Spain
Susana Muñoz Hernández	Technical University of Madrid, Spain
Wady Naanaa	Univ. of Monastir, Tunisia
Jose Neves	University of Minho, Portugal
Julio Cesar Nievola	Pontifical Catholic University of Paraná, Brazil
Agris Nikitenko	Technical University of Riga, Latvia
Paulo Novais	University of Minho, Portugal
Nadia Nouali-Taboudjemat	CERIST,, Argelia
Alberto Ochoa-Zezatti	Autonomous University of Ciudad Juarez, Mexico
José Luis Oliveira	University of Aveiro, Portugal
Eugénio Oliveira	University of Porto, Portugal
Jordi Palacin	Universitat de Lleida, Spain
Miguel Angel Patricio	University Carlos III of Madrid, Spain
Juan Pavón	University Complutense of Madrid, Spain
Reyes Pavón	University of Vigo, Spain
Pawel Pawlewski	University of Technology of Poznan, Poland

Diego Hernán Peluffo-Ordoñez	Cooperative University of Colombia, Colombia
Stefan-Gheorghe Pentiuc	University Stefan cel Mare Suceava, Romania
Antonio Pereira	School of Technology and Management IPLeiria, Portugal
Rafael Pereira	University of Minho, Portugal
João Miguel Pereira Alves	University of Minho, Portugal
Tiago Pinto	Polytechnic of Porto, Portugal
Julio Ponce	Autonomous University of Aguascalientes, Mexico
Juan-Luis Posadas-Yagüe	Polytechnic University of Valencia, Spain
Jose-Luis Poza-Luján	Polytechnic University of Valencia, Spain
Isabel Praça	Polytechnic of Porto, Portugal
Radu-Emil Precup	Politehnica University of Timisoara, Romania
Mar Pujol	University of Alicante, Spain
Francisco A. Pujol	University of Alicante, Spain
Mariano Raboso Mateos	Pontifical University of Salamanca, Spain
Miguel Rebollo	Polytechnic University of Valencia, Spain
Manuel Resinas	University of Seville, Spain
Ramon Rizo	University of Alicante, Spain
Mark Roantree	Dublin City University, Ireland
Sergi Robles	Autonomous University of Barcelona, Spain
Miguel Rocha	University of Minho, Portugal
Cristian Aaron Rodriguez Enriquez	Technology Institute of Orizaba, Mexico
Miguel Ángel Rodríguez-García	King Abdullah University of Science and Technology, Arabia Saudi
Luiz Romao	University of Univille, Brazil
José Leandro Salles	Federal Institute of Espirito Santo, Brazil
Gustavo Santos-Garcia	University of Salamanca, Spain
Ichiro Satoh	National Institute of Informatics, Tokio, Japan
Ken Satoh	National Institute of Informatics and Sokendai, Japan
Markus Schatten	Faculty of Organization and Informatics, Croatia
Thorsten Schöler	Augsburg University of Applied Sciences, Germany
Yann Secq	Lille University of Science and Technology, France
Carolina Senabre	University Miguel Hernández of Elche, Spain
Nuno Silva	Polytechnic of Porto, Portugal
Fábio Silva	University of Minho, Portugal
Pedro Sousa	University of Minho, Portugal
Agnieszka Stachowiak	Poznan University of Technology, Poland
Y.C. Tang	University of Tenaga Nasional, Malaysa
Victor Torres Padrosa	University of Girona, Spain
Leandro Tortosa	University of Alicante, Spain

Volodymyr Turchenko	Ternopil National Economic University, Ukraine
Miki Ueno	Toyohashi University of Technology, Japan
Zita Vale	Polytechnic of Porto, Portugal
Rafael Valencia-Garcia	University of Murcia, Spain
Miguel A. Vega-Rodríguez	University of Extremadura, Spain
Maria João Viamonte	GECAD/ISEP Engineering School Polytechnic of Porto, Portugal
Jose Vicent	University of Alicante, Spain
Paulo Vieira	Polytechnic Insitute of Guard, Portugal
José Ramón Villar	University of Oviedo, Spain
Friederike Wall	Alpen-Adria-University of Klagenfurt, Austria
Zhu Wang	XINGTANG Telecommunications Technology Co., Ltd., China
Li Weigang	University of Brasilia, Brazil
Morten Gill Wollsen	University of Southern Denmark, Denmark
Bozena Wozna-Szczesniak	University in Czestochowa, Poland
Michal Wozniak	Wroclaw University of Technology, Poland
Takuya Yoshihiro	Wakayama University, Japan
Michifumi Yoshioka	Osaka Pref. Univ., Japan
Agnieszka Zbrzezny	University in Czestochowa, Poland
Kamel Zidi	Science Faculty of Monastir, Tunisia
André Zúquete	University of Aveiro, Portugal

Special Session on AI–driven methods for Multimodal Networks and Processes Modeling Special Session Committee

Chairs

Grzegorz Bocewicz	Koszalin University of Technology, Poland
Paweł Sitek	Kielce University of Technology, Poland
Izabela E. Nielsen	Aalborg University, Denmark

Program Committee

Paweł Pawlewski	Poznan University of Technology, Poland
Zbigniew Banaszak	Warsaw University of Technology, Poland
Peter Nielsen	Aalborg University, Denmark
Robert Wójcik	Wrocław University of Technology, Poland
Marcin Relich	University of Zielona Gora, Poland
Arkadiusz Gola	Lublin University of Technology, Poland
Krzysztof Bzdyra	Koszalin University of Technology, Poland
Katarzyna Grzybowska	Poznan University of Technology, Poland

Organizing Committee

María José Escalona Cuaresma (Chair)	University of Sevilla, Spain
Carlos Arevalo Maldonado	University of Sevilla, Spain
Gustavo Aragon Serrano	University of Sevilla, Spain
Irene Barba	University of Sevilla, Spain
Miguel Ángel Barcelona Liédana	Technological Institute of Aragon, Spain
Juan Manuel Cordero Valle	University of Sevilla, Spain
Francisco José Domínguez Mayo	University of Sevilla, Spain
Juan Pablo Domínguez Mayo	University of Sevilla, Spain
Manuel Domínguez Muñoz	University of Sevilla, Spain
José Fernández Engo	University of Sevilla, Spain
Laura García Borgoñón	Technological Institute of Aragon, Spain
Julian Alberto García García	University of Sevilla, Spain
Javier García-Consuegra Angulo	University of Sevilla, Spain
José González Enríquez	University of Sevilla, Spain
Tatiana Guardia Bueno	University of Sevilla, Spain
Andrés Jiménez Ramírez	University of Sevilla, Spain
Javier Jesús Gutierrez Rodriguez	University of Sevilla, Spain
Manuel Mejías Risoto	University of Sevilla, Spain
Laura Polinario	University of Sevilla, Spain
José Ponce Gonzalez	University of Sevilla, Spain
Francisco José Ramírez López	University of Sevilla, Spain
Isabel Ramos Román	University of Sevilla, Spain
Jorge Sedeño López	University of Sevilla, Spain
Nicolás Sánchez Gómez	University of Sevilla, Spain
Juan Miguel Sánchez Begines	University of Sevilla, Spain
Eva-Maria Schön	University of Sevilla, Spain
Jesús Torres Valderrama	University of Sevilla, Spain
Carmelo Del Valle Sevillano	University of Sevilla, Spain
Antonio Vázquez Carreño	University of Sevilla, Spain
Carlos Torrecilla Salinas	University of Sevilla, Spain
Ainara Aguirre Narros	University of Sevilla, Spain
Diana Borrego	University of Sevilla, Spain
Fernando Enríquez de Salamanca Ros	University of Sevilla, Spain
Juan Antonio Alvarez García	University of Sevilla, Spain
Antonio Tallón	University of Sevilla, Spain

Contents

Part I 13th International Conference Distributed Computing and Artificial Intelligence

Part II Special Session on AI–driven methods for Multimodal Networks and Processes Modeling

Part I
13th International Conference Distributed Computing and Artificial Intelligence

Fuzzy Topsis Method Associated with Improved Selection of Machines of High Productivity

Lourdes Margain, Alberto Ochoa, Oscar Castillo, Saúl González and Guadalupe Gutiérrez

Abstract This study combines Fuzzy Logic and multicriteria TOPSIS method for the selection, from three different alternatives, which machines of high productivity is more convenient to a construction company. The evaluation of each alternative is made through group decision making which identifies the most important criteria according to the requirements presented by the company. To assess the selected criteria in the TOPSIS method is weighted by a group of experts who, based on their experience and knowledge of this type of machinery, assess the relevance of these in the operation and functioning of the hydraulic excavator. Both qualitative and quantitative studies are used in this work, however the experts evaluate, through surveys based on Likert scale all the criteria in which they want to measure the perception. Data provided from the surveys is used for the construction and association of the groups of expert's opinion through the use of fuzzy sets to avoid ambiguity problems of the linguistic variables.

Keywords TOPSIS · Likert · Linguistic labels · Fuzzy sets

L. Margain(✉) · G. Gutierrez
Universidad Politécnica de Aguascalientes, Aguascalientes City, Mexico
e-mail: lourdes.margain@upa.edu.mx

A. Ochoa · S. González
Laboratorio de Aeronáutica, Universidad Autónoma de Ciudad Juárez,
Ciudad Juárez, Mexico
e-mail: {alberto.ochoa,saugonza}@uacj.mx

O. Castillo
Instituto Tecnológico de Tijuana, Tijuana, Mexico
e-mail: ocastillo@tectijuana.mx

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_1

1 Introduction

Multicriteria decision making is a process of finding the best alternative among a set of optimal alternatives. Among the various compensatory methods of multicriteria decision making, it is possible to consider a subgroup that includes cost and benefits aspects. One of them is the TOPSIS (Technique for Order Performance by Similarity to Idea Solution) method [2], technique that allows to sort preferences by similarity to an ideal solution. The TOPSIS method is a model for making decisions with which it is possible to sort requirements in comparison to an ideal solution in order to acquire a hierarchical order of the compared alternatives. The intuitive concept of the ideal alternative would be the one that, without hesitate, would be selected by the decision maker. Similarly, the anti-ideal alternative would be the one, without hesitate, never would be selected by the decider [1]. This paper is organized as follows: in the next section, fuzzy numbers, fuzzy sets and linguistic variables are described. The third section presents TOPSIS and Likert methods. An application example is discussed in the fourth section and finally the main conclusions are presented.

2 Theoretical Framework

2.1 Fuzzy Numbers and Fuzzy Sets

A fuzzy set A in R1 is called a fuzzy number if A is convex and exists in an exact point, M ∈ R1, with mA(M) = 1 (Aα = 1 = M), [3]. The linguistic expression of such fuzzy number would be: "Approximately M". For a better manipulation, the fuzzy numbers are usually defined L-R (left-right). Fuzzy numbers are very useful to process the information in a fuzzy environment and implement a representation [3]. The usage of fuzzy sets is precise with expressions that do not have clear boundaries. A fuzzy set is associated to a linguistic value defined by an expression that in most cases is denominated linguistic label. The vaguely defined sets play an important role in human thought, particularly in fields such as pattern recognition, communication of information and abstraction. Fuzzy sets are used to perform a qualitative evaluation of a physical quantity [4]. For a fuzzy set, the belonging of an element to the set is not a question of all or nothing, but are possible different degrees of membership. Membership functions can take any real value in the interval [0,1]. That is, mA |U → [0, 1] is the membership function of a fuzzy set, if X has a non-empty set. A fuzzy set A in the X domain is characterized by membership function.

2.2 Linguistic Variables

When people use language to convey something they want to communicate, they use expressions that do not relate exactly what they want to express. However, the use of these expressions do not implicate they should be accurate since in the language process the expressions are used collectively between the sender and receiver, without being necessary that they mean the same for both. For example, if it is desired that a group of people describe the temperature of water contained in a container, it is possible to use the expressions such as cold, hot and warm, in that way it is obtained a set of fuzzy elements denominated by Zadeh [4]. The meaning of the linguistic labels is determined by the fuzzy sets.

3 TOPSIS

It is a mathematical programming technique originally used in continuous contexts and has been modified for the analysis of discrete multicriteria problems. This technique was developed by Hwang and Yoon [2] in 1981 and refined by the authors in 1987 and 1992, they also have developed different versions of other authors. In this paper the results of the weighting of the qualitative criteria by the Likert scale with fuzzy sets and quantitative criteria associated to the data sheet of each alternative are used to evaluate these by a multicriteria TOPSIS model. This methodology proposes an algorithm to determine the preferred choice among all possible alternatives.

3.1 Likert Scale

This scale requires the respondents to select a label (answer) to represent their personal perception of each of the statements that are presented [7]. In this paper the scale is used to acquire the opinion or perception of the evaluators to qualify and weigh the selected criteria to evaluate each alternative. The linguistic labels are used to evaluate the importance level of each criteria and aptitude of each alternative. In this surveys linguistic labels are used to calculate the weight of each criteria and for the construction of the decision matrix in the TOPSIS method.

3.2 Defuzzification Method

Among the possible existing defuzzification methods, this paper uses the method proposed by Garcia-Cascales and Lamata [1], because it allows to include bias parameters that affect the fuzzy set's value.

The calculation is made by the following expression:

$$I_{\beta,\lambda}(A_i) = \beta S_M(A_i) + (1-\beta)\lambda S_R(A_i) + (1-\beta)(1-\lambda) S_L(A_i)$$

where :

S_L (A_i) represents the lower medium value associated with the inverse function $g^{L}_{A}(x)$

S_M (A_i) is the heart's area of the fuzzy number.

S_R (A_i) represents the high medium value of the fuzzy number associated with the inverse function $f\overset{R}{\underset{A}{}}(x)$.

β ∈ [0,1] is the modality index representing the importance of the central value.

λ ∈ [0,1] is the optimism's degree of the decision maker.

3.3 Methodology

Step 1. Decider-makers group establishment and expert profile definition. The experts and decider-makers groups were integrated as the TOPSIS methodology requires.

Step 2. Planning. Determining the finite set of alternatives to achieve the goal.

Information search. The information of three backhoe loader was collected.

Step 3. Determination and justification of the evaluation criteria. The hydraulic excavators that are compared in this paper have similar physic and technical characteristics.

Step 4. Weighting for each criterion. To calculate the criteria's weight, a double survey is applied to each of the evaluators.

Step 5. Triangular fuzzy sets approximation for each label. The selected criteria for the evaluation of each of the proposed alternatives and parameters for the construction of the triangular fuzzy sets are associated to the linguistic labels: Very important (VI), Moderately important (MI), Important (I), Indifferent (IN) and No importance (NI).

Step 6. Survey Implementation. Likert and Semantic differential scale are used for the application of the surveys applied to the decider and expert groups.

Step 7. Fuzzy triangular sets modification for each label. To modify the parameter's values of each set associated to the linguistic labels, the numeric values that the evaluators related to the linguistic labels were identified and tabulated.

Step 8. TOPSIS implementation. The fuzzy TOPSIS methodology proposed by Garcia-Casales and Lamata is used.

Step 9. Obtaining of positive and negative ideal solution (A+, A–). Ideal solutions were obtained to be compared with each of the proposed alternatives.

Step 10. Calculation of the relative proximity of each alternative to the ideal solutions found.

Step 11. Calculation of the relative proximity to each of the alternatives with respect to the ideal solution it is made.

Step 12. Triangular sets defuzzification. A value that represents the triangular set is calculated and the associated to a linguistic label according to the numerical value obtained.

4 Study Case

This section describes a real case decision. This case is shown to illustrate the Fuzzy Topsis Method. The goal is to decide which is the best productivity machine that should use a constructor company to excavate, when there are floods of land on roads caused by rains, earthquakes or landslides for any other reason. Construction companies have machines as hydraulic excavators that have flaws in some of its parts. These companies need to perform a quality job quickly and to remove the alluvium. This activity should consider the time and cost of repairing failures in key parts as the rotation mechanism and the hydraulic system. Construction companies have machines type hydraulic excavators, with flaws in some of its parts. These companies should perform a quality job and quick to remove the alluvium. This activity should consider the time and cost of repairing failures in the key parts as the rotation mechanism and the hydraulic system. In addition, it is required to consider aspects such as ergonomic interface and load capacity. Some criteria are feasible to quantify, in other evaluation is performed only by using linguistic values. The hydraulic excavators compared in this paper have very similar physical and technical characteristics, they are manufactured by world renowned companies that stand out for their quality and prestige. Figure 4 shows the three different alternatives that are evaluated in this paper, the first one is a 349DL model from CAT, the second one is a 22T is a BC-6225 from SANY and the third one is a HB12LC-1 from KOMATSU.

4.1 Criteria to Evaluate

To derive the characteristics or criteria to be analyzed, opinions and needs of the decision making group were taken. In the next table each criteria is defined, as in table 1.

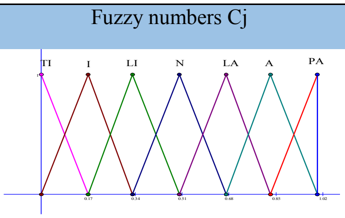
Table 1 Selected criteria

C1	Social Cost	C5	Engine power
C2	Rotation mecanism	C6	Fuel capacity
C3	Hydraulic system	C7	Weight
C4	Ergonomic interface		

4.2 Triangular Fuzzy Sets Approximation for Each Label

The linguistic variables that integrate the fuzzy sets are shown in table 2, wherein also the triangular weighting of each label is shown.

Table 2 Linguistic labels

Linguistic labels	Fuzzy numbers Cj	Fuzzy numbers Cj
Totally unacceptable (TU)	[0, 0, 0.167]	
Unacceptable (U)	[0, 0.167, 0.333]	
Lightly unacceptable (LU)	[0.167, 0.333, 0.5]	
Neutral (N)	[0.333, 0.5, 0.667]	
Lightly acceptable (LA)	[0.5, 0.667, 0.833]	
Acceptable (A)	[0.667, 0.833, 1]	
Perfectly acceptable (PA)	[0.833, 1, 1]	

4.3 Survey Application

For the survey application both, deciding group and experts group, their results were weight it, to obtain an average which became part of the TOPSIS. Likert scale is used to associate a linguistic label to each of the criteria evaluated. This survey is in charge to associate the importance level for each criteria on the personal perception of a group of evaluators. The results of this survey are used to weight and calculate the level of importance of the selected criteria through a double survey. In this paper fuzzy sets are defuzzyfied after the surveys have been applied, weighting the labels by the frequency they were selected.

4.4 Triangular Fuzzy Sets Modification for Each Label

To calculate the new parameter values for each associated fuzzy set to a linguistic label, all numerical values are identified and tabulated according to the linguistic labels the evaluator relate. The core value of the triangular number takes the geometric average value, the lower limit is the least numerical value received for the label, while the upper limit represents the larger label received it.

4.5 TOPSIS Application

Based on TOPSIS methodology with fuzzy numbers proposed by Garcia and Lamata, it starts with the identification of the weight of each criteria for each of the evaluators as is shown in in table 3.

Table 3 Labels associated to weights.

	E1			E2			E3			E4			E5			E6		
C1	0.9	0.967	1	0.9	0.967	1	0.9	0.967	1	0.9	0.967	1	0.9	0.967	1	0.9	0.967	1
C2	0.833	0.85	0.9	0.5	0.73	0.9	0.833	0.85	0.9	0.833	0.85	0.9	0.5	0.73	0.9	0.833	0.85	0.9
C3	0.85	0.89	0.9	0.85	0.89	0.9	0.9	0.967	1	0.85	0.89	0.9	0.85	0.89	0.9	0.9	0.967	1
C4	0.667	0.7	0.833	0.833	0.85	0.9	0.667	0.7	0.833	0.667	0.7	0.833	0.833	0.85	0.9	0.833	0.85	0.9
C5	0.95	0.983	1	0.95	0.983	1	0.95	0.983	1	0.95	0.983	1	0.95	0.983	1	0.95	0.983	1
C6	0.833	0.85	0.9	0.5	0.592	0.8	0.833	0.85	0.9	0.5	0.592	0.8	0.5	0.592	0.8	0.5	0.592	0.8
C7	0.833	0.85	0.9	0.833	0.85	0.9	0.833	0.85	0.9	0.833	0.85	0.9	0.833	0.85	0.9	0.5	0.592	0.8

The obtained normalized vectors for each criteria are shown in (table 4 and table 5).

Table 4 Normalized matrix for each criteria.

	Vector de pesos normalizado		
C1	0.14026	0.16287	0.17930
C2	0.11252	0.13647	0.16137
C3	0.13507	0.15426	0.16735
C4	0.11689	0.13058	0.15537
C5	0.14806	0.16568	0.17930
C6	0.09522	0.11420	0.14942
C7	0.12117	0.13596	0.15838

Table 5 Description of comparative values after initial analysis.

	C1			C2			C3			C4			C5			C6			C7		
A1	0.569	0.833	0.894	0.586	0.658	0.747	0.917	0.950	0.967	0.580	0.783	0.867	0.725	0.820	0.875	0.797	0.822	0.883	0.750	0.824	0.872
A2	0.508	0.576	0.683	0.622	0.705	0.767	0.758	0.865	0.928	0.725	0.820	0.875	0.672	0.823	0.917	0.800	0.839	0.911	0.589	0.700	0.806
A3	0.797	0.822	0.883	0.683	0.749	0.867	0.683	0.730	0.833	0.650	0.728	0.822	0.797	0.822	0.883	0.742	0.839	0.894	0.672	0.823	0.917
Σx^2	1.104	1.304	1.431	1.094	1.221	1.377	1.372	1.478	1.578	1.134	1.347	1.481	1.270	1.423	1.545	1.351	1.444	1.552	1.167	1.359	1.500

1. Construction of the initial matrix with the attribute's weight

2. Normalized decision matrix construction: The matrix normalized values for each criteria, as is shown in (Table 6).

3. Associated normalized weighted matrix construction: is obtained from equation:

$$\overline{v_{ij}} = \underline{w_j}\, \overline{n_{ij}}$$

Table 6 Normalized weighted matrix

	C1			C2			C3			C4			C5			C6			C7		
A1	0.056	0.104	0.145	0.048	0.074	0.110	0.078	0.099	0.118	0.046	0.076	0.119	0.069	0.095	0.124	0.049	0.065	0.098	0.061	0.082	0.118
A2	0.050	0.072	0.111	0.051	0.079	0.113	0.065	0.090	0.113	0.057	0.079	0.120	0.064	0.096	0.129	0.049	0.066	0.101	0.048	0.070	0.109
A3	0.078	0.103	0.143	0.056	0.084	0.128	0.058	0.076	0.102	0.051	0.071	0.113	0.076	0.096	0.125	0.045	0.066	0.099	0.054	0.082	0.124

4. Positive and negative ideal solutions obtainment: are obtained from equations 3 and

$$\bar{A}^{+} = \{\bar{v}_1^{+}, \ldots., \bar{v}_n^{+}\} = \{(\max \bar{v}_{ij}, j \ \in J), (\min \bar{v}_{ij}, j \ \in J')\}$$

$$\bar{A}^{-} = \{\bar{v}_1^{-}, \ldots., \bar{v}_n^{-}\} = \{(\min \bar{v}_{ij}, j \ \in J), (\max \bar{v}_{ij}, j \ \in J')\}$$

5. Distance calculation of each alternative to the ideal solutions found by and shown in (Table 7)**:**

$$\bar{d}_i^+ = \{\sum_{j \in J} (\bar{v}_{ij} - \bar{v}_j^+)^2\}^{\frac{1}{2}}, \quad i = 1, \dots, m$$

Table 7 Distance to the positive ideal solution

	C1			C2			C3			C4			C5			C6			C7		
A1	0.00050	0.00000	0.00000	0.00006	0.00010	0.00031	0.00000	0.00000	0.00000	0.00013	0.00001	0.00000	0.00005	0.00000	0.00003	0.00000	0.00000	0.00001	0.00000	0.00000	0.00004
A2	0.00080	0.00102	0.00118	0.00002	0.00002	0.00022	0.00018	0.00008	0.00002	0.00000	0.00000	0.00000	0.00014	0.00000	0.00000	0.00000	0.00000	0.00000	0.00017	0.00015	0.00023
A3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00040	0.00053	0.00026	0.00004	0.00008	0.00005	0.00000	0.00000	0.00002	0.00001	0.00000	0.00000	0.00004	0.00000	0.00000

The sum of all the criteria for each of the alternatives is made and equation 5 is used to calculate the distance to the positive ideal solution (Table 8).

Table 8 Distance to the positive ideal solution

	Sum			Distance		
	a	m	b	a	m	b
A1	0.000740147	0.000116566	0.00039239	0.02720564	0.010796573	0.019808831
A2	0.00132409	0.001281679	0.001643481	0.036388053	0.035800544	0.04053987
A3	0.000486779	0.000610494	0.000346002	0.022063059	0.024708181	0.018601126

To calculate the ideal negative solution (Table 9), equation is used:

$$\bar{d}_i^- = \{\sum_{j \in J} (\bar{v}_{ij} - \bar{v}_j^-)^2\}^{\frac{1}{2}}, \quad i = 1, \dots, m$$

Table 9 Distance to the negative ideal solution

	C1			C2			C3			C4			C5			C6			C7		
A1	4E-05	1E-03	1E-03	0E+00	0E+00	0E+00	4E-04	5E-04	3E-04	0E+00	3E-05	4E-05	3E-05	0E+00	0E+00	1E-05	0E+00	0E+00	2E-04	2E-04	8E-05
A2	0E+00	0E+00	0E+00	9E-06	3E-05	8E-06	4E-05	2E-04	1E-04	1E-04	8E-05	5E-05	0E+00	1E-07	3E-05	1E-05	2E-06	9E-06	0E+00	0E+00	0E+00
A3	8E-04	9E-04	1E-03	6E-05	1E-04	3E-04	0E+00	0E+00	0E+00	3E-05	0E+00	0E+00	1E-04	7E-08	1E-06	0E+00	2E-06	1E-06	5E-05	2E-04	2E-04

The sum of all the criteria for each of the alternatives is made and equation is used to calculate the distance to the negative ideal solution, as is shown in (Table 10).

Table 10 Distance calculation to the negative solution

	Suma			Distancia		
	a	m	b	a	m	b
A1	0.000642318	0.001736658	0.001559024	0.025344002	0.04167323	0.039484484
A2	0.00019314	0.0003082	0.000237193	0.013897488	0.017555626	0.015401071
A3	0.001084495	0.001197742	0.001597029	0.032931667	0.034608411	0.039962846

6. Relative proximity calculation for each positive and negative ideal solution through proximity index.

To conclude this study, The ideal alternative "Distance" will be divided by the ideal alternative "Distance" minus the anti-ideal alternative "Distance". In this paper the alternate ranking method is used. To defuzzify the results, the proposed values by Garcia-Cascales and Lamata β=1/2 and λ=1/3 are taken, corresponding to a neutral decision level and a second option is calculated to define the best alternative, because the observed bias during data collection in the surveys, for this reason β=1/2 and λ=1/3 values are proposed, in this case equal importance is given to the right and left areas of the triangular fuzzy numbers obtained by the assigned values by the experts, as is shown in (Table 11).

Table 11 Decision index

	Neutral defuzzified output	**Bias defuzzified output**	**Neutral order**	**Bias order**	
A1	0.729528116	0.707880799	1	1	CAT
A2	0.313698264	0.308572001	3	3	SANY
A3	0.591864834	0.594656304	2	2	Komatsu

5 Results and Recommendations

The analysis of the final classificated data allows to observe that the three alternatives exhibit distant values, however it would be difficult to decide which criteria make a trend to generate a selection without making a analysis to determine which is the best option for the decision making group. Although the three alternatives meet the requirements and satisfy the buyer at some point, the fuzzy TOPSIS method finds which is the optimal alternative, however it is possible to consider the real bias that can be calculated taking the frequency of the surveys.

References

1. García, M.Y., Lamata, M.: Nueva aproximación al método TOPSIS difuso con etiqueta lingüística. In: Tecnologías y lógica fuzzy, pp. 619–624 (2010)
2. García, M., García, M.: Métodos para la comparación de alternativas mediante un sistema de ayuda a la decisión (S.A.D.) y "Soft Computing. Tesis doctoral, Departamento de electrónica, tecnología de computadoras y proyectos, Universidad politécnica de Cartagena (2009)
3. Morilla, A.: Introducción al análisis de datos difusos. Edición electrónica. Texto completo en www.eumed.net/libros/2006b/amr/ (2006)
4. Zadeh, L.A.: Fuzzy sets. Information and Control **8**, 338–353 (1965)
5. Hwang, C.L., Yoon, K.: Multiple Attribute Decision Making: Methods and Applications. Springer Verlag, Alemania / Estados Unidos (1981)
6. Nettleton, D.: Técnicas para el análisis de datos clínicos. Ediciones Díaz de Santos, España (2005)
7. Malhotra, N.: Investigación de mercados, 5th edn. Pearson educación, México (2008)

A Conceptual Automated Negotiation Model for Decision Making in the Construction Domain

Moamin A. Mahmoud, Mohd Sharifuddin Ahmad and Mohd Zaliman M. Yusoff

Abstract In this paper, we propose a conceptual automated multi-agent negotiation model for decision making in the construction domain. The proposed model enables software agents to conduct negotiations and autonomously make decisions. The model consists of two components, namely, Agent Architecture and Negotiation Approach. The Agent Architecture combines a Negotiation Algorithm and a Negotiation Protocol that enable each agent to conduct and regulate the negotiation process. The Negotiation Approach combines a Decision-making Process based on Value Management, a Negotiation Process Base that hosts negotiation operations among agents and a Conflict Resolution Algorithm. The paper presents the conceptual findings of these components.

Keywords Intelligent software agent · Multi-agent systems · Agent and negotiation · Automated negotiation · Value Management · Construction domain

1 Introduction

In the construction domain, deciding on a new project is dependent upon a company's strategy. If the strategy is based on a decision by a stakeholder, then it takes a shorter time to decide. However, such decision has no significance in terms of value management, because the decision-making process does not include other experienced stakeholders that hold different backgrounds.

In the construction domain, a project manager usually cares more about the cost and schedule of a project than the function while a design manager is more concerned about the function than the cost. Thus, for any decision to be made

M.A. Mahmoud(✉) · M.S. Ahmad · M.Z.M. Yusoff
Centre for Agent Technology, College of Information Technology,
Universiti Tenaga Nasional, Kajang, Selangor, Malaysia
e-mail: {moamin,sharif,zaliman}@uniten.edu.my

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_2

regarding a new project, stakeholders must propose an optimal solution. However, a problem may arise when stakeholders propose many solutions. In such a situation, stakeholders need to negotiate on the proposed solutions and agree on an optimal solution. But the negotiation may not be easy and smooth because when stakeholders possess different backgrounds, often their views about an optimal solution for a particular project are different. Such differences cause conflicts in arriving at a decision. In addition, stakeholders may work at different branches throughout the country or other parts of the world which make a meeting for decision more difficult and costly. While applying Value Management on decision making in the construction domain is useful, it faces communication difficulties between stockholders and conflicting issues that require negotiation.

In this paper, we attempt to overcome these difficulties by proposing a Value-based Automated Negotiation model utilizing the multi-agent system's approach. It enables software agents to conduct negotiations and autonomously arrive at a decision. This paper is an extension to our work in the concepts of automated multi-agent negotiation [1, 2].

The proposed automated negotiation model consists of an Agent Architecture and a Negotiation Approach. The Agent Architecture combines i) a Negotiation Algorithm that enables agents to conduct a negotiation and arrive at a decision of an optimal solution and ii) a Negotiation Protocol that enables each agent to conduct and regulate the negotiation process. The Negotiation Approach combines, iii) a Decision-making Process based on Value Management that coordinate the negotiation between agents until a decision is reached, iv) a Negotiation Process Base that hosts all negotiation activities between agents and v) a Conflict Resolution Algorithm, which resolves conflicts of two or more proposed solutions by agents by filtering solutions that do not meet a required level of safety.

While this work is inspired by the work of Utomo [3], his study is only in conceptual level and lacks a complete negotiation process that aids an agent to interact and negotiate with other agents and respond to its environment and eventually influences its autonomy level in decision making.

2 Related Work

In this section, we discuss two prominent topics of this research which are value management and application of negotiation in multi-agent systems.

Value Management (VM) is defined as "a structured, organized team approach to identify the functions of a project, product, or service that will recognize techniques and provide the necessary functions to meet the required performance at the lowest overall cost" [4]. Utomo et al. [3] defined VM as one of the decision methodologies that include a multi-disciplinary, team-oriented approach to problem solving [5]. Therefore, negotiation plays an important role on VM using a value-based group decision [3]. VM is based on a data collection method from reliable resources and functional requirements to fulfill the needs, wants and desires of customers [3].

The application of VM in decision making has been reported by many researchers [3, 6, 7]. One of the techniques that is relevant to VM is weighting and scoring in which a decision needs to be made in selecting an option from a number of competing options, and the best option is not immediately identifiable [3, 8, 9].

Intelligent software agents have been widely used in distributed artificial intelligence and due to their autonomous, self-interested, rational abilities [11, 12, 13, 14, 15, 16], and social abilities [17, 18, 19, 20], agents are well-suited for automated negotiation on behalf of humans [10]. According to Kexing [10], automated negotiation is a system that applies artificial intelligence and information and communication technology to negotiation strategies, utilizing agents and decision theories.

Numerous research have discussed negotiation on multi-agent systems in various domains [21, 22, 23, 24, 25]. Coutinho et al. [26] proposed a negotiation framework to serve collaboration in enterprise networks to improve the sustainability of interoperability within enterprise information systems. Utomo [3] presented a conceptual model of automated negotiation that consists of a negotiation methodology and an agent-based negotiation. Dzeng and Lin [27] presented an agent-based system to support a negotiation between constructors and suppliers via the Internet. Anumba et al. [28] proposed a collaborative design of light industrial buildings based on multi-agent systems to automate the interaction and negotiation between the design members. Ren et al. [22] developed a multi-agent system representing participants, who negotiate with each other to resolve construction claims.

3 A Conceptual Model for Automated Negotiation

As shown in Figure 1, the proposed automated negotiation model consists of five main components which are i) Negotiation Algorithm, ii) Negotiation Protocol, iii) Decision-making Process based Value Management, iv) Negotiation Process Base and v) Conflict Resolution Algorithm.

In this paper, we present the Conceptual Model for Automated Negotiation and the Automated Negotiation Process and detail out two components: the Negotiation Protocol and the Decision-making Process.

As shown in Figure 1, each agent is equipped with a negotiation algorithm and a negotiation protocol. The agent negotiate with other agents via the negotiation process base. Agents' negotiation operation via the negotiation process base is coordinated by a process of decision-making and influenced by a conflict resolution algorithm. The output of the approach is a single solution.

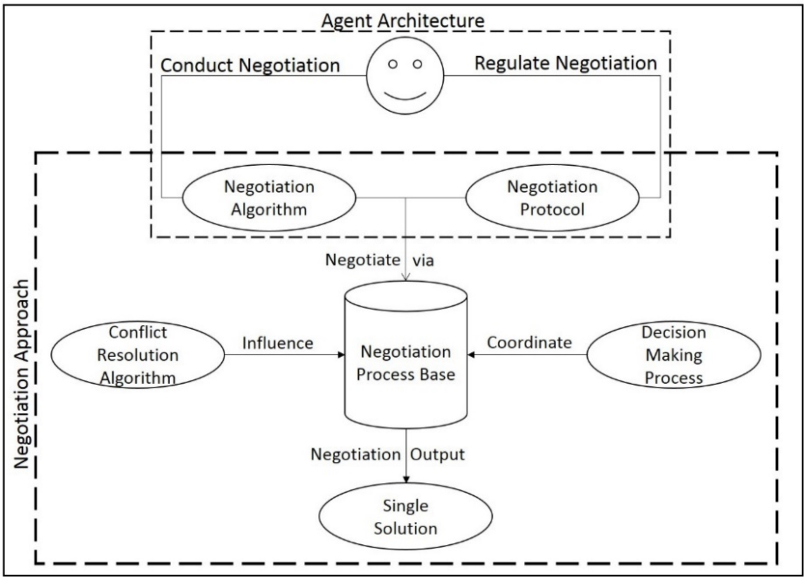


Fig. 1 A Conceptual Model for Automated Negotiation

4 The Automated Negotiation Process

Figure 2 shows a simple scenario of two agents negotiating via the negotiation process base. The negotiation algorithm and the negotiation protocol enable the agents to conduct negotiation while the negotiation process base provides a middle

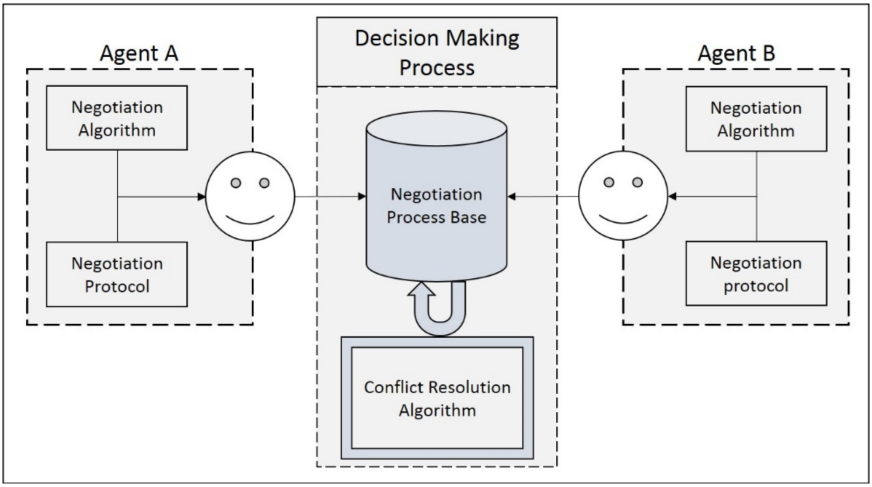


Fig. 2 The Automated Negotiation Process

ground for the agents to negotiate. If a conflict occurs after negotiation, the conflict resolution algorithm mitigates the conflict. The flow of negotiation including conflict resolution to reach a decision is coordinated by a decision-making process. The following sections discuss and present the process of each component presented in Figure 2.

4.1 Decision-Making Process Based on Value Management

A decision made by an agent goes through several processes. These processes work by gradually reducing candidate solutions of a project until a single solution is reached. Consequently, in this work, the process of nominating a single solution from a set of solutions is called decision-making.

There are three main processes in decision-making for a specific project, which are propose solutions, negotiate solutions and handling conflicting outcomes (conflict resolution).

- Propose solutions: In this process, each agent proposes solutions and ranks them from 1st to nth solution where n is any natural number. In this work, we assume an agent has the ability to do these tasks.
- Negotiate solutions: When ranked solutions are ready, agents negotiate by submitting their ranked solutions to each other. Since each agent's target is to maximize its utility by selecting a solution that has a better order, each agent prepares a plan. Using these plans, agents form coalitions among them based on similar plans. These coalitions continuously compare plans with each other until a single or more solutions converge after exhausting all attempts. Section 3.3 discusses the process in greater details.
- Resolve conflict: If agent coalitions agree upon a single solution, then this process is completed, but if there are two or more conflicting solutions, then the conflicts need to be resolved. This process resolves conflicts based on each coalition's strength and its solutions' risks. From these two parameters, this process drops solutions until a single solution is reached. Section 4 discusses this in more details.

Figure 3 shows the flowchart of the decision-making process as described above. The process starts when agents receive a new project. The agents first propose solutions in ranked order. They then negotiate these solutions. If they agree upon a single solution, then the decision is made, otherwise, the conflict resolution process takes over to drop the weak and risky solutions. If the outcome of the conflict resolution process is a single solution then the decision is made. Otherwise, the agents negotiate the outcome of the conflict resolution process. Ultimately, one coalition's solution is accepted.

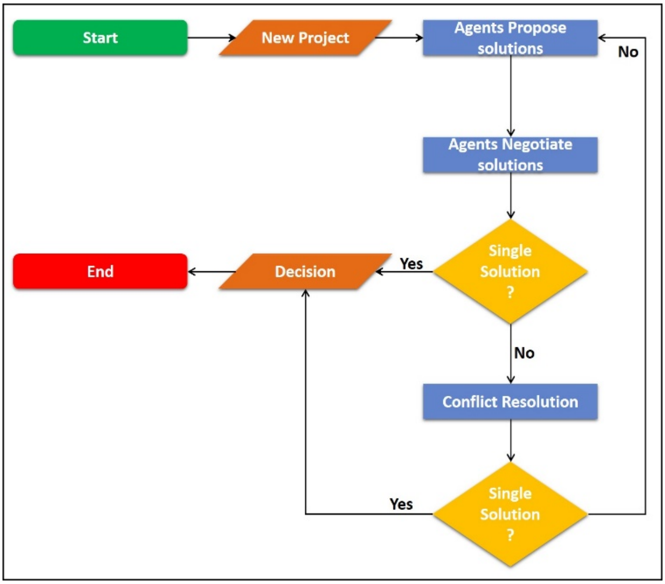


Fig. 3 Decision Making Process

4.2 Negotiation Protocol

In this work, agents conduct negotiation according to a predefined protocol. Such protocol ensures that the negotiation progresses smoothly. As shown in Figure 4, an agent first proposes its solutions (in rank) and then submits them to the negotiation base. When all agents have submitted their solutions, the negotiation base allows agents to review each solution weightage. From this information, the

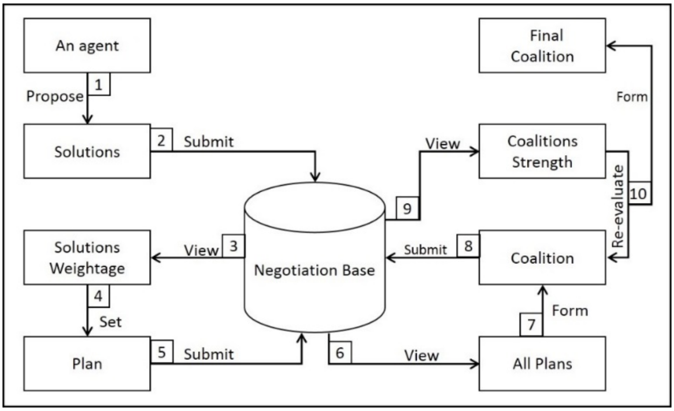


Fig. 4 Agent's Negotiation Protocol

agent sets a plan and then submits its plan to the negotiation base. When all agents have submitted their plans, the negotiation base allows the agents to review each other's plans. The agents then form coalitions and submit their decisions to the negotiation base. When all agents have formed coalitions, the negotiation base allows the agents to view each coalition's strength. The agents re-evaluate their coalitions according to the strength and then forms the final coalition.

4.3 Negotiation Algorithm

The algorithm implements the negotiation process between agents. The process starts when each agent submits its solutions to the negotiation base. Each agent then reviews each solution's and accordingly sets a plan to conduct negotiation.

4.4 Negotiation Process Base

The Negotiation Process Base represents the negotiation hub that is used by agents to form negotiations by sharing their solutions and form coalitions. The base helps in reducing direct interactions between agents that increase the network load. All negotiations are processed via this base which is accessible by all agents.

4.5 Conflict Resolution Algorithm

The need for this algorithm is based on the negotiation outcomes. Since any project needs a single solution, then when the negotiation outcome is a single solution, agents skip this algorithm. But when the outcome is several solutions, then another process is needed to resolve this conflict. Such situation represents a conflict between agents about the solution of that project.

5 Conclusion and Further Work

In this paper, we present our initial findings on our research to develop a conceptual automated multi-agent negotiation model for decision-making in the construction domain. The automated negotiation model consists of agent architecture and a negotiation approach and combines five components which are i) a negotiation algorithm and ii) a negotiation protocol that enables each agent to conduct and regulate the negotiation process. In addition to the agent architecture components, the negotiation approach combines iii) a decision-making process based on value management, iv) a negotiation process base that host negotiation operations among agents and v) a conflict resolution algorithm.

We also present a predefined Negotiation Protocol that ensures that the negotiation progresses smoothly and a Decision-making Process flow that coordinates the negotiation between agents until a decision is reached.

Since this work is in its theoretical stage, it only presents the conceptual underpinnings of pertinent issues in negotiation and does not present the experimental results. Such outcome will be presented in our future work.

In addition, for our future work, we shall study and propose mechanisms for the three components presented in the model which are Negotiation Algorithm, Negotiation Process Base and Conflict Resolution Algorithm.

References

1. Mahmoud, M.A., Ahmad, M.S., Yuso, M.Z.M., Idrus, A.: An Automated Negotiation-based Framework via Multi-Agent System for the Construction Domain. International Journal of Artificial Intelligence and Interactive Multimedia **3**(5), 23–27 (2015)
2. Mahmoud, M.A., Ahmad, M.S., Yusoff, M.Z.M., Idrus, A.: Automated Multi-agent Negotiation Framework for the Construction Domain. In: 12th International Conference Distributed Computing and Artificial Intelligence, pp. 203–210. Springer International Publishing (2015)
3. Utomo C.: Development of a negotiation Support Model for Value Management in Construction, Ph.D. Thesis, University Teknologi PETRONAS, December 2009
4. SAVE International, value methodology standards (2001)
5. Kelly J., Male S.: Value Management in Decision and Construction, The Economic Management of Projects. Spon Press, London
6. Jaapar, A., Endut, I.R., Bari, N.A.A., Takim, R.: The impact of value management implementation in Malaysia. Journal of Sustainable Development **2**(2) (2009)
7. Shen, Q., Chung, J.K.H., Li, H., Shen, L.: A Group Support System for improving value management studies in construction. Automation in Construction **13**(2004), 209–224 (2004)
8. Cariaga, I., El-Diraby, T., Osman, H.: Integrating Value Analysis and Quality Function Deployment for Evaluating Design Alternatives. Construction Engineering and Management **133**(10), 761–770 (2007)
9. Qing, Y., Wanhua, Q.: Value Engineering Analysis and Evaluation. For the Second Beijing Capital Airport. Value World, Spring, SAVE International (2007)
10. Kexing L.: A survey of agent based automated negotiation. In: 2011 International Conference on Network Computing and Information Security (NCIS), vol. 2, pp. 24–27. IEEE (2011)
11. Ahmed, M., Ahmad, M.S., Yusoff, M.Z.M.: Modeling agent-based collaborative process. In: The 2nd International Conference on Computational Collective Intelligence Technology and Applications (ICCCI 2010), pp. 296-305, Taiwan, November 10–12, 2010. ISBN: 3-642-16692-X 978-3-642-16692-1
12. Mahmoud, M.A., Ahmad, M.S., Yusoff, M.Z.M., Idrus, A.: Automated multi-agent negotiation framework for the construction domain. In: Distributed Computing and Artificial Intelligence, (DCAI 2015), Spain. Advances in Intelligent Systems and Computing, vol. 373, pp. 203–210, June 3–5, 2015. Springer International Publishing (2015)
13. Itaiwi, A.K., Ahmad, M.S., Hamid, N.H.A., Jaafar, N.H., Mahmoud, M.A.: A framework for resolving task overload problems using intelligent software agents. In: 2011 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2011 (2011)

14. Ahmed, M., Ahmad, M.S., Yusoff, M.Z.M.: A Collaborative Framework for Multiagent Systems. International Journal of Agent Technologies and Systems (IJATS) **3**(4), 1–18 (2011)
15. Ahmed, M., Ahmad, M.S., Yusoff, M.Z.M.: Mitigating human-human collaboration problems using software agents. In: The 4th International KES Symposium on Agents and Multi-Agent Systems – Technologies and Application (AMSTA 2010), pp. 203–212, Gdynia, Poland, June 23–25, 2010. ISBN:3-642-13479-3 978-3-642-13479-1
16. Itaiwi, A.-M.K., Ahmad, M.S., Hamid, A., Hamimah, N., Jaafar, N.H., Mahmoud, M.A.: A multi-agent framework for dynamic task assignment and delegation in workload distribution. In: International Conference on Computer & Information Sciences, June 12, 2012
17. Mahmoud, M.A., Ahmad, M.S., Mohd Yusoff, M.Z., Mustapha, A.: A Review of Norms and Normative Multiagent Systems. The Scientific World Journal (2014)
18. Mahmoud, M.A., Ahmad, M.S., Ahmad, A., Yusoff, M.Z.M., Mustapha, A., Hamid, N.H.A.: Obligation and prohibition norms mining algorithm for normative multi-agent systems. In: KES-AMSTA, pp. 115–124, May 2013
19. Mahmoud, M.A., Ahmad, M.S., Ahmad, A., Yusoff, M.Z.M., Mustapha, A.: Norms detection and assimilation in multi-agent systems: a conceptual approach. In: Knowledge Technology, pp. 226–233. Springer, Heidelberg (2012)
20. Mahmoud, M.A., Ahmad, M.S., Ahmad, A., Mohd Yusoff, M.Z., Mustapha, A.: A norms mining approach to norms detection in multi-agent systems. In: 2012 International Conference on Computer & Information Science (ICCIS), vol. 1, pp. 458–463. IEEE, June 2012
21. Beer, M., d'Inverno, M., Jennings, R.N., Luck, M., Preist, C., Schroeder, M.: Negotiation in multi-agent systems. Knowledge Engineering Review **14**(3), 285–289 (1999)
22. Ren, Z., Anumba, C.J.: Multi-agent systems in construction—state of the art and prospects. Automation in Construction **13**, 421–434 (2004)
23. Wang, M., Wang, H., Vogel, D., Kumar, K., Chiu, D.K.W.: Agent-based negotiation and decision making for dynamic supply chain formation. Engineering Applications of Artificial Intelligence **22**(7), 1046–1055 (2009)
24. Utomo C., Idrus A.: A Concept toward Negotiation Support for Value Management on Sustainable Construction. Journal of Sustainable Development **4**(6) (2011)
25. Sanchez-Anguix, V., Julian, V., Botti, V., García-Fornes, A.: Tasks for agent-based negotiation teams: Analysis, review, and challenges. Engineering Applications of Artificial Intelligence **26**(10), 2480–2494 (2013)
26. Coutinho, C., Cretant, A., Ferreira da Silva, C., Ghodous, P., Jardim-Goncalves, R.: Service-based negotiation for advanced collaboration in enterprise networks. Journal of Intelligent Manufacturing (2014). doi:10.1007/s10845-013-0857-4
27. Dzeng, R.J., Lin, Y.C.: Intelligent agents for supporting construction procurement negotiation. Expert Systems with Applications **27**(1), 107–119 (2004)
28. Anumba, C.J., Ren, Z., Thorpe, A., Ugwu, O.O., Newnham, L.: Negotiation within a multi-agent system for the collaborative design of light industrial buildings Adv Eng Software **34**(7), 389–401 (2003)

Self-Adaptive Organizations for Distributed Search: The Case of Reinforcement Learning

Friederike Wall

Abstract In this paper we study the effects of learning by reinforcement and adaptive change of distributed search systems' organizations. We find that employing learning by reinforcement to direct organizational alterations of distributed search systems may lead to high levels of systems' performance and this, in particular, with rather high efficiency in terms of effort of reorganization. The results also suggest that the complexity of the search problem together with the aspiration level, relevant for the positive or negative reinforcement, considerably shape the effects of learning.

Keywords Agent-based simulation · Complexity · NK fitness landscapes · Reinforcement learning

1 Introduction

The organization of distributed search processes has been studied in various disciplines, like control theory, complex systems science or computational organization theory to name but a few (for reviews [16], [5], [4]). A major topic of these research efforts relates to the coordination of distributed agents which has been differentiated into streams on consensus, formation control, optimization, task assignment, and estimation [16]. However, there is some evidence that distributed search processes could remarkably benefit from inducing organizational dynamics in the course of the search for better solutions ([15], [14], see also [1]): In particular, organizational dynamics per se - i.e., even if alterations of organizational features are driven by random - appear to induce a shift towards more exploration, i.e., discovery of new solutions, and less exploitation, i.e., stepwise improvement. With respect to practical

F. Wall(✉)
Alpen-Adria-Universitaet Klagenfurt, 9020 Klagenfurt, Austria
e-mail: friederike.wall@aau.at
http://www.aau.at/csu

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_3

use, these findings suggest that frequent organizational changes are beneficial - may it be, for example, in the organizational set-up of collaborating robots or “swarms” of unmanned aerial vehicles.

Building on these results, in this paper we go a step beyond previous research (especially [15], [14]) by introducing a form of learning-based organizational dynamics: *In particular, the paper analyzes the effects of endowing distributed search systems with some capabilities to learn about their organization’s performance and to adapt the organization accordingly on the search systems’ performance.* For this purpose, we employ an agent-based simulation model which captures two intertwined adaptive processes: (1) In the short term the search agents - operating on NK fitness landscapes [7], [8] - seek to find superior levels of the search system’s overall performance. (2) In the mid term the search systems adapt major features of their organizational structure via learning by reinforcement based on the performance enhancements achieved.

2 Outline of the Simulation Model

2.1 Short Term Adaptive Search for Higher Levels of Performance

Search Problem. In each time step t of the observation period T, our search systems face an N-dimensional binary search problem, i.e., they seek for a superior configuration $\mathbf{d_t} = (d_{1t}, ..., d_{Nt})$ with $d_{it} \in \{0, 1\}$, $i = 1, ...N$, out of 2^N different binary vectors possible. Each of the two states $d_{it} \in \{0, 1\}$ contributes with C_{it} to fitness $V(\mathbf{d_t})$ of the search system. In line with the NK framework, C_{it} is randomly drawn from a uniform distribution with $0 \leq C_{it} \leq 1$. An advantage of NK fitness landscapes is that they allow to easily control the complexity of the underlying search problem by parameter K [10]. In particular, K reflects the number of choices $d_{jt}, j \neq i$ which also affect the fitness contribution C_{it} of choice d_{it}. In case of no interactions K is 0, and K equals $N - 1$ for maximum interactions. With this, fitness contribution C_{it} might not only depend on the single choice d_{it} but also on K other choices d_{jt} where $j \in \{1, ...N\}$ and $j \neq i$:

$$C_{it} = f_i(d_{it}, d_{jt, j\in\{1,...N\}, j\neq i}). \quad (1)$$

The overall fitness (performance) V_t achieved in period t results as the normalized sum of contributions C_{it} from

$$V_t = V(\mathbf{d_t}) = \frac{1}{N}\sum_{i=1}^{N} C_{it}. \quad (2)$$

Agents and Their Choices. In our model, the search for higher levels of fitness is conducted collaboratively by several search agents. In particular, the N-dimensional search problem is partitioned into M disjoint partial problems and each of these sub-problems is exclusively delegated to a search agent r, $r = 1, ..., M$. Hence, from the perspective of search agent r the search problem is segmented into a partial search vector d_t^r for those choices which are in its own primary control and into partial vector $d_t^{r,res}$ for the residual choices that the other search agents $q \neq r$ are in charge of. However, with cross-segment interactions among the sub-problems, choices of agent r might affect the contribution of the other agents' choices on overall performance, and vice versa.

In each time step t, a search agent seeks to identify the best configuration for the "own" choices d_t^r assuming that the other agents do not alter their prior choices. For this, an agent r randomly discovers two alternatives to status quo d_{t-1}^{r*} - an alternative configuration that differs in one choice ($a1$) and another ($a2$) where two bits are flipped compared to the current configuration. Hence, in time step t, agent r has three options to choose from, i.e., keeping the status quo or switching to $d_t^{r,a1}$ or $d_t^{r,a2}$. Which of these options is favorable from a search agent's perspective depends on the agent's "objective" P_t^r when assessing the options. An agent might focus only on the "own" partial search problem or may, at least partially, take the residual part of the search problem into consideration - depending on parameter α^r in Eq. (3):

$$P_t^r(\mathbf{d_t}) = P_t^{r,own}(d_t^r) + \alpha^r \cdot P_t^{r,res} \tag{3}$$

$$\text{with } P_t^{r,own}(d_t^r) = \frac{1}{N} \sum_{i=1+p}^{N^r} C_{it} \text{ and } P_t^{r,res} = \sum_{q=1,q\neq r}^{M} P_t^{q,own}. \tag{4}$$

where $p = \sum_{s=1}^{r-1} N^s$ for $r > 1$ and $p = 0$ for $r = 1$.

However, the agents' evaluations of alternatives need not necessarily be perfect, i.e., the agents may misjudge the options' contributions to objective $P_t^r(\mathbf{d_t})$. This may not only be an unintentional shortcoming of, e.g., agents' information processing capacities but also may be intentionally induced: Some evidence suggests that imperfect information on the fitness (performance) of options could increase the effectiveness of search processes (e.g. [9], [13]). (This is, in particular, since false-positive evaluations of options increase the diversity of search by providing the opportunity to leave a local peak and, by that, to eventually find higher levels of fitness.) Hence, intentionally or not, our agents may eventually be endowed with slightly distorted information about the fitness of options. We capture distortions by adding error terms as exemplarily shown in Eq. (5):

$$\tilde{P}_t^{r,own}(d_t^r) = P_t^{r,own}(d_t^r) + e^{r,own}(d_t^r) \tag{5}$$

For the sake of simplicity, we depict distortions as relative errors imputed to the true performance (for other functions see [9]). The error terms follow a Gaussian distribution $N(0; \sigma)$ with expected value 0 and standard deviations $\sigma^{r,own}$ and $\sigma^{r,res}$

are assumed to be the same for search agents r and stable in time; errors are assumed to be independent from each other.

Apart from the search agents, our model captures a kind of "central agent" whose role is a twofold one: (1) In the short termed adaptive search the central agent could - depending on the particular mode of coordination - intervene in the selection of choices. (2) In the mid term, the central agent assesses performance enhancements and "learns" about successful organizational structures by reinforcement. We go more in detail of both roles in the next section.

2.2 Mid Term Adaptation of the Organizational Set-Up Based on Reinforcement Learning

Mode of Reinforcement Learning. The very core of our research is whether learning on the search system's organization together with altering the organizational set-up accordingly could enhance performance. For this, from time to time, our search systems can alter their organizational set-up according to L dimensions. In particular, in each T^*-th time step the central agent faces an L-dimensional decision problem and chooses a configuration $\boldsymbol{\phi}_\mathbf{t} = (a_1(t), ..., a_L(t))$ of alternatives $a_l \in A_l$ for all $l = 1, ...L$ and with $|A_l|$ giving the number of alternatives a_l in set A_l.

We employ a simple mode of reinforcement learning (for overviews see [12], [6]) based on statistical learning, i.e., a generalized form of the Bush-Mosteller model ([2], [3]): The propensities of choices are updated according to the - positive or negative - stimuli resulting from the outcome (payoff) of prior choices. Whether the outcome ω of configuration $\boldsymbol{\phi}_\mathbf{t}$ at time step t is regarded positive or negative, depends on whether, or not, it at least equals an aspiration level v. Outcome ω_t of configuration $\boldsymbol{\phi}_\mathbf{t}$ is defined as the maximal relative performance enhancement achieved within the last T^* periods of the adaptive walk, i.e.,

$$\omega_t(\boldsymbol{\phi}_\mathbf{t}) = \max[(V_{t-\tilde{t}} - V_{t-T^*})/V_{t-T^*}, \tilde{t} = 1, ...(T^* - 1)]. \tag{6}$$

Hence, the stimulus $\tau(t)$ is

$$\tau(t) = \begin{cases} 1 \text{ if } & \omega_t(\boldsymbol{\phi}_\mathbf{t}) \geq v \\ -1 \text{ if } & \omega_t(\boldsymbol{\phi}_\mathbf{t}) < v \end{cases} \tag{7}$$

$p(a_l, t)$ denotes the probability of an alternative within dimension l of organizational design to be chosen at time t (with $0 \leq p(a_l, t) \leq 1$ and $\sum_{a_l \in A_l}(p(a_l, t)) = 1$); $a_l(t)$ denotes that option of set A_l which is implemented in time step t. The probabilities of options $a_l \in A_l$ are updated according to the following rule, where λ (with $0 \leq \lambda \leq 1$) reflects the reinforcement strength [3]:

$$p(a_l, t+1 = p(a_l, t) + \begin{cases} \lambda \cdot \ \tau(t) \cdot (1 - p(a_l, t)) & \text{if } \ a_l = a_l(t) \wedge \ \tau(t) = 1 \\ \lambda \cdot \ \tau(t) \cdot p(a_l, t) & \text{if } \ a_l = a_l(t) \wedge \ \tau(t) = -1 \\ -\lambda \cdot \ \tau(t) \cdot p(a_l, t) & \text{if } \ a_l \neq a_l(t) \wedge \ \tau(t) = 1 \\ -\lambda \cdot \ \tau(t) \cdot \frac{p(a_l,t) \cdot p(a_l(t),t)}{1 - p(a_l(t),t)} & \text{if } \ a_l \neq a_l(t) \wedge \ \tau(t) = -1 \end{cases} \tag{8}$$

After the probabilities are updated as given in Eq. (8) the "next" organizational configuration $\boldsymbol{\phi}$ to be implemented from $t + 1$ to $t + T^*$ is determined randomly according to the updated probabilities.

Organizational Design. In our simulations, the organizational design vector $\boldsymbol{\phi}$ is three-dimensional, i.e. $L = 3$. Within each dimension three options are given (i.e. $|A_l| = 3 \forall l = 1 \dots 3$). These dimensions relate to (see also Table 1):
(1) the *objective* of the search agents as controlled by parameter α^r in Eq. (3)
(2) the *precision of ex ante-evaluations* as given by $\sigma^{r,own}$, $\sigma^{r,res}$ and σ^{cent}
(3) the *mode of coordination*: (a) "decentralized": the search agents decide on their "own" partial choices d_t^r autonomously without any intervention by the central agent; (b) "lateral veto": the search agents inform each other about their preferences and are endowed with mutual veto power; (c) "centralized": each search agent informs the central agent about the two most preferred options from d_{t-1}^{r*}, $d_t^{r,a1}$ and $d_t^{r,a2}$, the central agent chooses that combination of preferences which promises the highest overall performance V.

3 Simulation Experiments and Parameter Settings

In the simulation experiments, after a fitness landscape is generated, the initial organizational set up (i.e. $\boldsymbol{\phi}$) of a search system is determined randomly out of the options in each dimension l as introduced above and summarized in Table 3 with uniform probabilities $p(a_l, t = 0)$ within each dimension. Then the systems are placed randomly in the fitness landscape and observed while searching for higher levels of performance and, in each T^*-th period, updating probabilities and altering their organization. In order to oppose learning search systems (esp. $\lambda = 0.5$) to non-learning systems employing organizational change we also conduct simulations for$\lambda = 0$. Moreover, we simulate search systems which do not alter their organization within the observation time T (i.e., with $T^* > T$). In order to capture the complexity of the underlying search problem, we conduct simulations for two interaction structures which, in a way, represent two extremes [11]: in the *block-diagonal* structure the overall search problem can be segmented into two disjoint parts with maximal intense intra-sub-problem interactions but no cross-sub-problem interactions. In contrast, in the *full interdependent* case all single options d_i affect the performance contributions of all other choices (i.e., intensity of interactions K is maximal).

Table 1 Parameter Settings

Parameter	Values / Types
observation period	$T = 250$
number of choices	$N = 10$
interaction structures	block-diagonal ($K = 4$); full interdependent ($K = 9$)
number of search agents	$M = 2$, agent 1: $d^1 = (d_1, ...d_5)$, agent 2: $d^2 = (d_6, ...d_{10})$
number of org. dimensions	$L = 3$
agents' objective	$\alpha^r \in \{0; 0.5; 1\}$
precision of evaluation	$(\sigma^{r,own}; \sigma^{r,res}; \sigma^{cent}) \in \{(0.1; 0.15; 0.125);$ $(0.05; 0, 2; 0.125); (0; 0; 0)\}$
coordination mode	decentralized, lateral veto, central
interval of change	$T^* = 25$ and for contrasting to "no change": $T^* > T$
learning strength	$\lambda \in \{0; 0.5\}$
aspiration level	$v \in \{0; 0.01\}$

4 Results

Table 2 displays condensed results of the scenarios simulated. The final performance ($V_{t=250}$) achieved in the end of the observation period and the performance achieved on average in each of the 250 periods ($\bar{V}_{\{0;250\}}$) may serve as indicators for the effectiveness of the search process. The same applies to the frequency of how often the global maximum is found in the final period. Figure 1 depicts the adaptive walks for the different scenarios and interaction structures.

Results confirm findings of prior research indicating that altering the organizational set-up in the course of distributed search processes may be favorable ([15], [14], [1]): It has been argued that this is since organizational change increases the diversity of search and, thus, reduces the peril of sticking to local peaks. This is confirmed by the ratio of altered configurations implemented and the frequency of how often the global maximum is found at $t = T$ as reported in Table 2.

However, results suggest that learning by reinforcement is not universally beneficial. Apparently, the complexity of the search problem in conjunction with the aspiration level subtly affects its benefits. In order to explain why, in case of the highly complex search problems, a high aspiration level induces such a rather poor performance, we argue that this is due to the specific selective effects induced: With a high aspiration level it becomes rather unlikely that a positive stimulus $\tau(t)$ follows from a certain organizational set-up - even if it had brought some (lower than v) performance enhancements in the last T^* periods. Hence, even potentially advantageous organizational options are likely to receive low probabilities to be chosen next time again. Highly complex search problems are particularly prone to this effect, since they incorporate the peril of sticking in local peaks and, with that, no performance enhancements occurring. The latter would satisfy a low aspiration level $v = 0$ (and, thus, induce a positive stimulus) but not in case of a high aspiration level.

The average number of dimensions in which alterations occur during the adaptive search (rightmost column in Table 2) might be regarded as an indicator for the effort

Table 2 Condensed Results

Scenario of learning and change*	Final perf. $V_{t=250}$*	Avg. perf. $\bar{V}_{\{0;250\}}$*	Frequency glob. max. in $t = 250$	Ratio alt. conf. of **d**	Avg. no. changed dimensions
block-diagonal structure					
no change	0.9562	0.9502	26.66%	8.53%	0.0
not learning, change	0.9698	0.9587	40.50%	10.57%	20.1
learning, low aspir.lv., change	0.9676	0.9574	36.50%	10.23%	6.0
learning, high aspir.lv., change	0.9606	0.9546	34.92%	13.41%	17.0
full interdependent structure					
no change	0.8839	0.8713	6.14%	6.62%	0.0
not learning, change	0.8953	0.8772	9.70%	11.18%	20.0
learning, low aspir.lv., change	0.8965	0.8771	8.44%	9.88%	5.8
learning, high aspir.lv., change	0.8774	0.8666	8.64%	15.86%	17.3

* Notes: Confidence intervals, at a confidence level of 99.9%, for $V_{t=250}$ range between 0.002 and 0.005, for $\bar{V}_{\{0;250\}}$ between 0.001 and 0.003; scenarios: "no change": $T^* > 250$; "not learning, change": $\lambda = 0$, $\upsilon = 0.01$); "learning with low aspiration level, change": $\lambda = 0.5$, $\upsilon = 0$; "learning with high aspiration level, change": $\lambda = 0.5$, $\upsilon = 0.01$. For further parameter settings see Table 1. Each row shows the results of 5,000 adaptive walks.

("costs"), if any, of organizational dynamics and, thus, the *efficiency* of the mode of change and learning. Obviously, the context of the search organization is relevant for whether, or not, and, if so, in which shape costs of organizational change occur. For example, in case of a network of unmanned aerial vehicles collaboratively serving a certain service area the switch from one coordination mode to another might not cause any costs (apart from activating another of already available coordination mechanisms); however, in case of firm managers collaboratively searching for better configurations of key performance drivers reorganizations are rather costly, including, for example, learning costs of new organizational procedures or the adjustment of incentive schemes. Hence, the average number of dimension changes may be rather critical for the efficiency of inducing organizational dynamics of search.

In respect of efficiency of organizational dynamics we find that in both interaction structures under investigation employing learning combined with a low aspiration level yields good performance combined with rather high efficiency if compared to the other scenarios: For example, the "not learning, change"-scenario requires on average 20 reorganizations; however, in case of the full interdependent structure (i.e. highly complex search problem) "learning with low aspiration level" leads to similar final and average performance levels - with only around 6 alterations on average of the search processes. Hence, when alterations are not costless employing learning by reinforcement may be particularly valuable due to its effects on efficiency of organizational dynamics.

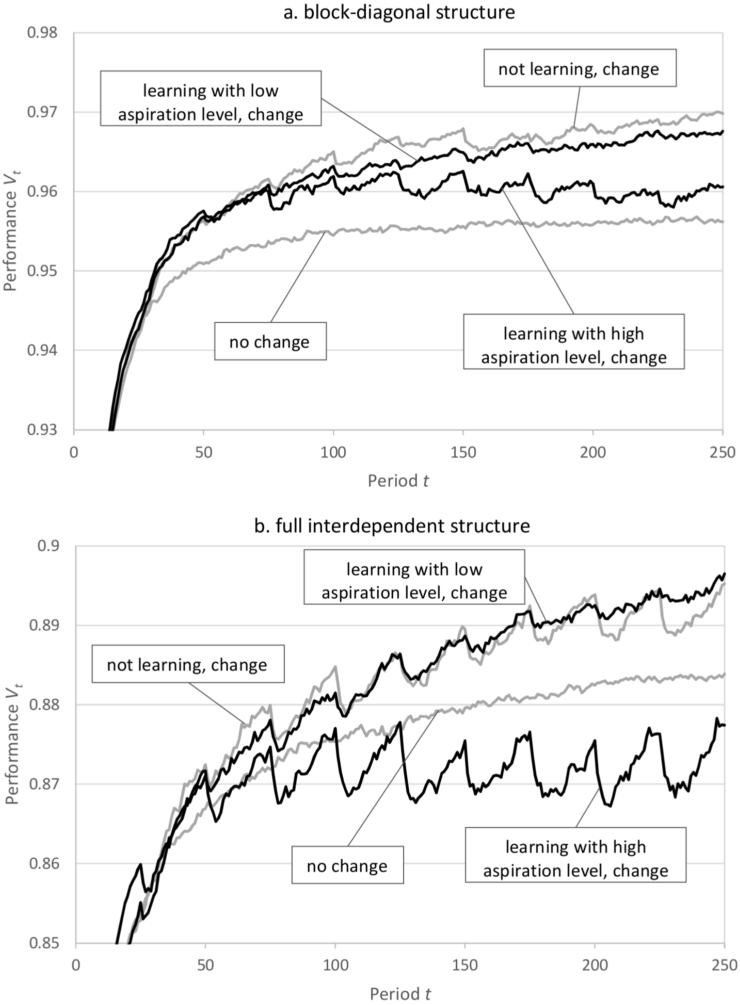


Fig. 1 Adaptive search processes for a. the block-diagonal and b. the full interdependent interaction structure. Each curve represents the average of 5,000 adaptive walks, i.e., 1,000 distinct fitness landscapes with 5 adaptive walks on each. For parameter settings see Table 1.

5 Conclusion

The major finding of our study is that employing learning by reinforcement for organizational alterations of distributed search systems potentially may lead to high levels of performance of the system and this, in particular, with rather high efficiency as given by the costs of reorganization. These findings are of particular interest especially when reorganizing the search system causes marginal costs - may it, for example, due to learning of new organizational procedures on the agents' site or required adjustments in institutional arrangements.

However, our results also suggest that the complexity of the search problem together with the aspiration level considerably shapes the effects of reinforcement learning - which, at worst, may even be harmful if compared to refraining from any organizational alterations. In particular, these findings may sensitize the designer of a distributed search system employing learning by reinforcement to that the level of performance enhancements aspired should not be overstretched in order to avoid "hyper-actively and ineffectively" alternating search systems.

Of course, these findings call for further research efforts. For example, further studies should investigate more into detail the role of the aspiration level and other parameters like the interval between of organizational alterations or the learning strength - which were fixed in the simulation experiments presented in this paper. Moreover, the basic search problem, as it is based on the NK-framework, is rather unstructured in terms of randomized performance contributions (apart from the structure of interactions); hence, in further research studies learning-based organizational adjustments of the search system may turn out even more beneficial in case of more structured search problems.

References

1. Baumann, O.: Distributed Problem Solving in Modular Systems: the Benefit of Temporary Coordination Neglect. Syst. Res. Behav. Sci. **32**, 124–136 (2015)
2. Bush, R.R., Mosteller, F.: Stochastic Models for Learning. Wiley, Oxford (1955)
3. Brenner, T.: Agent learning representation: advice on modelling economic learning. In: Tesfatsion, L., Judd, K.L. (eds.) Handbook of Computational Economics, vol. 2, pp. 895–947. Elsevier, Amsterdam (2006)
4. Carley, K.M., Gasser, L.: Computational organization theory. In: Weiss, G. (ed.) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 299–330. MIT Press, Cambridge (1999)
5. Gross, T., Blasius, B.: Adaptive coevolutionary networks: a review. J. R. Soc. Interface **20**, 259–271 (2008)
6. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
7. Kauffman, S.A., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. J. Theor. Biol. **128**, 11–45 (1993)
8. Kauffman, S.A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, Oxford (1993)
9. Levitan, B., Kauffman, S.A.: Adaptive walks with noisy fitness measurements. Mol. Divers. **1**, 53–68 (1995)
10. Li, R., Emmerich, M.M., Eggermont, J., Bovenkamp, E.P., Bck, T., Dijkstra, J., Reiber, J.C.: Mixed-integer NK landscapes. In: Parallel Problem Solving from Nature IX, vol. 4193, pp. 42–51. Springer, Berlin (2006)
11. Rivkin, R.W., Siggelkow, N.: Patterned interactions in complex systems: Implications for exploration. Manage. Sci. **53**, 1068–1085 (2007)
12. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press, Cambridge (2012)

13. Wall, F.: The (beneficial) role of informational imperfections in enhancing organisational performance. In: Lecture Notes in Economics and Mathematical Systems, vol. 645, pp. 115–126. Springer, Berlin (2010)
14. Wall, F.: Effects of organizational dynamics in adaptive distributed search processes. In: 12th International Conference on Distributed Computing and Artificial Intelligence. Advances in Intelligent Systems and Computing, vol. 373, pp. 121–128. Springer, Berlin (2015)
15. Wall, F.: Organizational dynamics in adaptive distributed search processes: effects on performance and the role of complexity. Frontiers of Information Technology & Electrical Engineering (in press)
16. Yongcan, C., Wenwu, Y., Wei, R., Guanrong, C.: An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. IEEE Trans. on Industrial Informatics **9**, 427–438 (2013)

Distributed Denial of Service (DDoS) Attacks Detection Using Machine Learning Prototype

Manuel S. Hoyos Ll, Gustavo A. Isaza E, Jairo I. Vélez and Luis Castillo O

Abstract The Distributed Denial of Service (*DDoS*) attacks affect the availability of Web services for an indeterminate period of time, flooding the company's servers with fraudulent requests and denying requests from legitimate users, generating economic losses by unavailable rendered services. Therefore, the aim of this paper is to show the process of detection prototype *DDoS* attacks using a supervised learning model by Support Vector Machines (SVM), which captures network traffic, filters HTTP headers, normalizes the data on the basis of the operational variables: rate of false positives, rate of false negatives, rate of classification and then sends the information to corresponding SVM's training and testing sets. The results show that the proposed DDoS SVM prototype has high detection accuracy (99 %) decrease of the false positives and false negatives rates compared to conventional detection models.

Keywords SVM · Machine learning · Intrusion detection · DDoS

1 Introduction

DDoS attacks are a critical issue for companies that have been integrating their technology to public networks, allowing multiple attackers to access data or render services to large companies or countries, such as North Korea, the most recent

M.S. Hoyos Ll(✉) · J.I. Vélez
Department of Computer Science,
University Autónoma of Manizales, Manizales, Colombia
e-mail: shoyos@gmail.com, jvelez@autonoma.edu.co

G.A. Isaza E · L. Castillo O
GITIR Research Group, Department of Systems and Informatics,
University of Caldas, Manizales, Colombia
e-mail: {gustavo.isaza,luis.castillo}@ucaldas.edu.co, lfcastilloos@unal.edu.co

L. Castillo O
Department of Industrial Engineering, National University of Colombia (Manizales),
Manizales, Colombia

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_4

relevant DDoS Attack turning internet communications offline [1]. A DDoS attack consists in to throw tens or hundreds of thousands of requests per second to a server from different locations or IPs; the concept of "Distributor" is concerning that these requests are made from hundreds of thousands of infected machines (commonly called "zombies") which are governed by "botnets" in a coordinated way at the same time, i.e. SYN Flood, Smurf attacks, which are a sum of bandwidth, memory usage and target's processing, usually no servers could handle ending in a collapse of service because it cannot answer every request; therefore it's necessary the development of new techniques and prototypes to detect fraudulent attacks of concurrent requests in an effective and efficient way also it's necessary in order to avoid the unavailability of service and economic losses. Machine learning using SVM have been used with great success in the field of information security and pattern recognition research in different processes of classification, prediction and regression. The application of techniques with a SVM supervised model has large advantages over rule-based techniques, since the generation of the model is based on a statistical model that changes its behavior according to the input parameters defined and based on a training rule that requires human interaction; in the prototype evaluation it was found that the correct classification rate of normal or abnormal requests in the training phase is directly related to standardization and proper selection of the input parameters, allowing the output variables are generated with minimum percentage of misclassification, generating confidence in the generated model and the detection of these behaviors. The paper describes: The contextual reference and some relevant work in section 2. The Section 3 shows the proposed model of the development and application of machine learning prototype. In section 4 some results are evident and in the end, finally in Section 5 the conclusions are presented.

2 Context and Recent Literature

Since 1998 and 1999, DARPA has collected and distributed the first standard dataset for evaluation of intrusion detection systems for computer networks. The first formal, repeatable and statistically significant evaluations of intrusion detection systems are coordinated. These assessments measure probability of detection and false alarm for each system under test and are designed to be simple, to focus on the key issues of technology, and to encourage the broadest possible participation by eliminating problems of security and privacy and provide data types that are commonly used by most systems intrusion detection. The evaluation results suggested that future research should aim to develop new algorithms to detect new attacks but creating static rules or signatures. Since that moment, many experts began working on techniques and models able to resolve this problem. Below are presented some proposed techniques in intrusion detection systems (IDS):

Rules: Responsible for analyzing the traffic that goes through the IDS, classifying the frame as normal or as intrusion. This technique uses a database of knowledge where a set of rules is applied to compare traffic patterns with the parameterized rules in the database [2].

As proposed by [4] on a system based intrusion detection agents with a rules engine based on XML, it can be that the argued model was presented as a distributed intrusion detection system, which consists of three intelligent agents that have specific functions and exchange information via XML sure how SSL and a point-to-point and IAP. The decision to do so and distributed intelligent agents, is based on the intrusion detection systems distributed traditional drawbacks, such as the hierarchy analysis, data refinement, bulky modules at all levels and passive interaction. The distributed sensing system offers more functions intrusions and strong individual actions IDS using intelligent agents.

Neural Networks: Artificial neural networks (ANN) are inspired by the behavior of neurons in the biological world, seeking to emulate in technology. [3]; as proposed by [5], about an automatic defense against distributed denial of service, it can argue model; the authors propose a model of automatic defense that prevents human interaction techniques based on artificial neural networks. An architectural design, which can be easily adapted protocols in each layer of the OSI reference model and algorithms of learning machines. Support Vector Machine (SVM) is a technique based on machine learning, where data is classified by determining a group of support vectors and characteristics to be quantified are described. As proposed by [6, 7], the hybrid system Intrusion Detection (HIDS), based on machine learning and specifically the SVM technique, improve the detection rate. More recent study as [10] presents a better classification using an Artificial Neural Network (ANN) to flag detection engine Known and unknown attacks from genuine traffic.

3 Materials and Method

3.1 Design

In the research field of pattern recognition, machine learning using SVM have been used with great success indifferent classification and regression tools. Some cores used internally by the SVM are: linear, polynomial, radial (RBF) and sigmoid. The design will focus on showing the different layers and levels that have high computational prototype and the components involved in the extraction, filter, standardization, training and evaluation. The prototype design of computational architecture exhibits the following logic, shown in Figure 1.

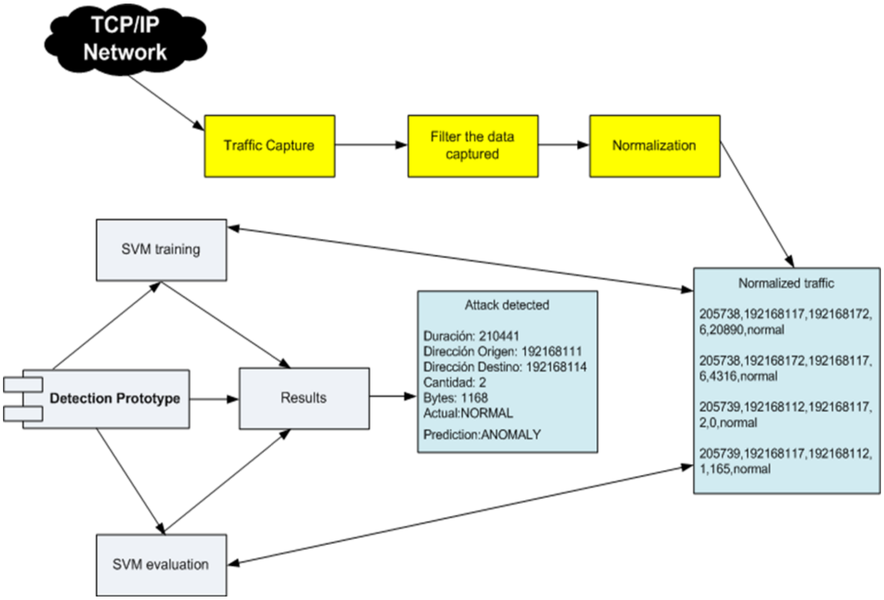


Fig. 1 Logical Architecture Detection Prototype SVM

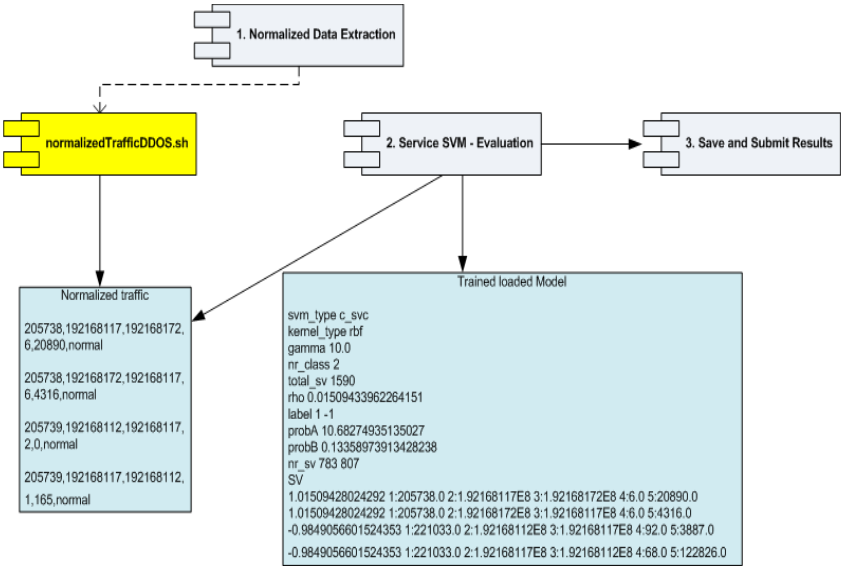
In the Figure 1, the architecture can be seen as the prototype has separate layers capture, filter and standardization of information, with the use of a language to efficiently implement regular expressions required to estimate the volume of information. Evaluation and training will be conducted on a high level language that allows the integration of SVM library and manage the concept of multiprocessing in an effective and efficient way independent of the used operating system.

3.2 Implementation and Training

The detection prototype is technically divided as follows: The collection, filtering and traffic normalization is performed on shell and python scripting, considering the performance of regular expressions and pipes offered by language. Training and evaluation was performed in Java programming language, given that the SVM library was installed on it and the features mentioned before were in the attacks generator. Data collection, filtering and Normalization aims to recover the network traffic, filter and normalizes the data, taking into account the operational variables defined in the design stage, TCPDUMP analyzer was used on the server to capture network traffic in the *Dataset Collection*, after a filtering and normalization stages are applied. The training phase is to receive standardized information and send it to the SVM to perform statistical clustering process of abnormal and normal requests in the generated model.in a first step the whole process of capturing network traffic will take place in the second stage filter HTTP headers on the basis of the information downloaded, in the third stage the standardization process where operational variables are selected to be used, as fourth stage integration process and training prototype with a standard percentage of traffic, and as a last step the respective evaluation of the prototype in the classification of anomalous or normal entry.

The prototype integrates *libsvm*[1] library in JAVA, which allows training, evaluating and generating a statistical model, based on standardized information. The internal structure of the generated model can be seen in Figure 2 that corresponds to a matrix groups of the entered operational variables and generates real value qualifying results: (Normal: 1.0 and Abnormal: -0.0).

The model is physically stored on a server path and is based on the evaluation phase or detection. The evaluation phase is to receive the standard traffic and send it to the SVM to perform the process of detecting the attack based on the model that was generated in the training stage. Below diagram components of the training and evaluation process are shown in Figure 2. The evaluation process has as its starting point the traffic sent by users and it's standardized at the initial stage, the prototype loads the generated model and discusses each record individually, in order to compare the statistical model generated in step training and identify indeed if it is generating an attack.


1. Normalized Data Extraction
normalizedTrafficDDOS.sh
2. Service SVM - Evaluation
3. Save and Submit Results
Normalized traffic
Trained loaded Model

```
205738,192168117,192168172,
6,20890,normal

205738,192168172,192168117,
6,4316,normal

205739,192168112,192168117,
2,0,normal

205739,192168117,192168112,
1,165,normal
```

```
svm_type c_svc
kernel_type rbf
gamma 10.0
nr_class 2
total_sv 1590
rho 0.01509433962264151
label 1 -1
probA 10.68274935135027
probB 0.13358973913428238
nr_sv 783 807
SV
1.01509428024292 1:205738.0 2:1.92168117E8 3:1.92168172E8 4:6.0 5:20890.0
1.01509428024292 1:205738.0 2:1.92168172E8 3:1.92168117E8 4:6.0 5:4316.0
-0.9849056601524353 1:221033.0 2:1.92168112E8 3:1.92168117E8 4:92.0 5:3887.0
-0.9849056601524353 1:221033.0 2:1.92168117E8 3:1.92168112E8 4:68.0 5:122826.0
```

Fig. 2 Components in the evaluation phase or detection

4 Results

The final standard file structure displayed by the operational variables defined in the design stage, presented in Table 1.

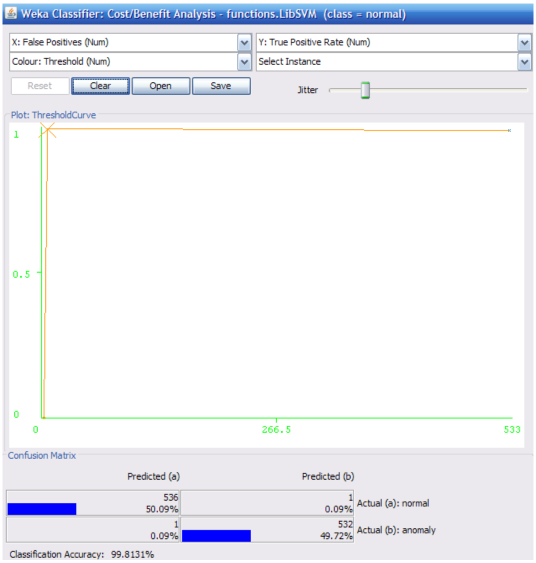
[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 1 Structure of normalized attributes

Variable	Description	Type
DURATION	Time in seconds for each request.	Continuous
IP_SRC	IP Source address	Continuous
IP_DEST	IP destination address.	Continuous
IP_SRC_COUNT	Number of requests from the source IP.	Continuous
BYTES	Number of bytes sent from a source to a destination.	Continuous
CLASS	Traffic Classification (Normal, Anomalous).	Discrete

For the information gathering phase, were sent a set of normal and abnormal requests for parallel and sequential process, with attacks prototype generator system test data, sending five iterations of 500 requests over a period determined time of 15 minutes. Total pooled and stored records 2698 between normal and anomalous traffic were normalized. The complete data normalized traffic (normal 1349, anomalous 1349), are divided into the training stage, randomly selecting 60% of the dataset (normal 809 abnormal 809), the other 40% (normal 539 abnormal 539) for evaluation. With the percentage defined training and evaluation, proceed to perform the respective tests the prototype, in order to determine if it has an acceptable level of detection. Before looking at the results of the training and testing phase, it has been used the metrics applied in [8, 9].

It was randomly selected 60% of the dataset (normal 809, 809 anomalous), and training to perform the respective prototype, saving the information into a file and classifying the data manually with normal and anomalous traffic. This process must be done manually because the SVM is a supervised method which requires prior training for model generation and respective evaluation of the attack. After training, 40% of the dataset of machine learning (normal 539, 539 anomalous) is evaluated, based on the model generated in the previous stage and analyzing the percentage of right and wrong classification of the instances. To make the respective evaluation, the information is saved in a file and the data is automatically sorted, placing as the default normal traffic. Performance metrics shown in the evaluation phase have the same relevance in the training phase because they allow us to see whether or not the SVM is correctly classifying the requests. In this case, the evaluation is a good rating level. ROC curves based viewing rate of false positives and true positives (detection rate) are shown in Figure 3. In the same way as in the training phase in the X axis is the false positive and in the Y axis is the true positive, if the value on the Y axis is close to 1 and the value on the X axis approaches to 0, it means that the events will be better able to detect and therefore greater ability to discriminate between normal and anomalous behavior. In order to make the comparison between a technique that uses a rules engine as SNORT, against another technique using SVM learning machines were taken on the data collected for the evaluation (normal 539 anomalous 539) and became the detection process. The prototype implemented the SVM technique had a significant improvement



Weka Classifier: Cost/Benefit Analysis - functions.LibSVM (class = normal)

X: False Positives (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

Plot: ThresholdCurve

Confusion Matrix

Predicted (a)	Predicted (b)	
536 50.09%	1 0.09%	Actual (a): normal
1 0.09%	532 49.72%	Actual (b): anomaly

Classification Accuracy: 99.8131%

Fig. 3 ROC curve

(approximately 10%) in the classification accuracy when compared with the intrusion detection system SNORT using the rules defined by human experts.

The internal statistical model of SVM identified anomalous or normal data in the evaluation phase without the need to add or modify the rules as another IPS (Intrusion Prevention Systems), causing the intrusion detection process is carried out quickly, automatically and without human interaction. The results obtained comparing metrics using a conventional Open Source IPS [11] is presented in Table 2.

Table 2 SVM DDoS Prototype Performance Results metrics vs Conventional IDS

	SNORT **Rules Engine IDS**		**DDoS Prototype** **using SVM**	
	Anomalous	Normal	Anomolous	Normal
Anomalous	TVP: 88.9%	TFP: 12.2%	TVP: 98.9%	TFP: 0.2%
Normal	TFN	TVN	TFN	TVN
Accuracy: 89%	Sensibility: 88%		Accuracy: 99%	Sensibility: 98%

5 Conclusions

The training process and tuning the Machine Learning from the standard data set are the basis for the generated model and it has an acceptable percentage of classification at the time of the evaluation of the prototype in a production environment with real information. The selection of metrics in the intrusion detection problem: false positive rate, false negative rate, rate classification, ROC curves, allow having a standard of comparison against other models. The application of techniques with supervised training as SVM model, has large advantages over the technique based on rules, since the generation of the model is based on a statistical model that changes its behavior according to the input parameters defined in the training and based on rules it requires human interaction. In the prototype evaluation was found a better classification rate for normal and anomalous requests in the training phase, is directly related to standardization and proper selection of input parameters, allowing output variables to be generated with minimum percentage of misclassification, generating reliability in the generated model and the detection of these behaviors.

References

1. Keizer, G.: Garden-variety DDoS attack knocks North Korea off the Internet. Recovered March 13, 2015 (2014). http://www.computerworld.com/article/2862652/garden-variety-ddos-attack-knocks-north-korea-off-the-internet.html
2. Chan, A., Ng, W., Yeung, D., Tsang, E.C.: Refinement of rule-based intrusion detection system for denial of service attacks by support vector machine. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4252–4256 (2004)
3. Kartalopoulos, S.: Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications, 1st edn. Wiley-IEEE Press (1996)
4. Liu, W.-T.: Research on intrusion detection rules based on XML in distributed IDS. In: International Conference on Machine Learning and Cybernetics, vol. 3, pp. 1400–1403, 12 de 07 de 2008
5. Mukkamala, S., Sung, A.: Detecting denial of service attacks using support vector machines. In: The 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003, vol. 2, pp. 1231–1236 (2003)
6. Seufert, S., O' Brien, D.: Machine Learning for Automatic Defence Against Distributed Denial of Service Attacks. In: IEEE International Conference on Communications, ICC 2007, pp. 1217–1222, 24–28 de Junio de 2007
7. Subbulakshmi, T., Shalinie, S., GanapathiSubramanian, V., BalaKrishnan, K., AnandKumar, D., Kannathal, K.: Detection of DDoS attacks using Enhanced Support Vector Machines with real time generated dataset. In: 2011 Third International Conference on Advanced Computing (ICoAC), pp. 17–22, 14.16 de Diciembre de 2011
8. Isaza, G.A., Castillo, L.F., Trujillo, M.L., Marulanda, C.E.: Modelo híbrido de neuroclasificación y clustering en el problema de detección de intrusiones. Vector, 69–77 (2012)

9. Isaza, G.A., Castillo, A., Lopez, M.F., Castillo, L.: Towards Ontology-based intelligent model for Intrusion Detection and Prevention. Journal of Information Assurance and Security **5**(2), 376 (2010)
10. Saied, A., Overill, R.E., Radzik, T.: Artificial Neural Networks in the Detection of Known and Unknown DDoS Attacks: Proof-of-Concept. In: Communications in Computer and Information Science, vol. 430, pp. 300–320. Springer-Verlag, Heidelberg (2014). doi:10.1007/978-3-319-07767-3_28
11. Kacha, C., Shevade, K.A.: Comparison of Different Intrusion Detection and Prevention Systems. Intl. Journal of Emerging Technology and Advanced. Engineering **2**(12), 243–245 (2012)

The Algorithm of the Snail: An Example to Grasp the Window of Opportunity to Boost Big Data

Jean-Louis Monino and Soraya Sedkaoui

Abstract The arrival of the so-called Petabyte Age has compelled the analytics community to pay serious attention to development of scalable algorithms for intelligent data analysis. This research provides a possible solution to analyze the growing quantity of data. An algorithm that can check coordinates to locate suppliers and applicants who live next door. A business application using our algorithm has been developed by the Autour.com company (located in the department of Herault, Montpellier city) to illustrate its feasibility and availability. The results show that our algorithm can improve the localization accuracy.

Keywords Snail algorithm · Localization · Big data analysis · Herault

1 Introduction

The age of big data is now coming. But the traditional data analytics may not be able to handle such large quantities of data. Therefore, there is a need to be able to recognize the appropriate analytics technique to use for the data and business problem. However, these companies are interested in applications that enable a range of data (coordinates) relative to a reference point, for example, the situation

J.-L. Monino(✉)
Laboratoire Traitement et Recherche Sur l'Information et la Statistique -
TRIS-Montpellier University, Montpellier, France
e-mail: jean-louis.monino@univ-montp1.fr

S. Sedkaoui
University of Khemis Miliana, Khemis Miliana, Algeria
e-mail: Soraya.sedkaoui@gmail.com

S. Sedkaoui
TRIS Laboratory, Montpellier University, Montpellier, France

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_5

of a manufacturer to offer a good or a service in relation to applicant of this type of good or service. The challenge identified is to organize the flow of data from their sources, process them and make them available to different users, but as applicant information is needed still more. To locate the closest ads you need a fast and efficient way which allows for the status or position of what is sought with respect to an address that sums up to a reference point which is the point of research. So it is more than necessary to plan and implement modern applications with sufficient details and observations covering the points between.

This work explores a new application which can effectively meet different localization accuracy requirements of most data location services studying the interactions between customers and suppliers. It helps to have the status or position of what is sought with respect to an address that summarizes thus a reference point which is the point of research. This proposal explains what snail algorithm is and how we can benefit from using it for the localization of information for business applications especially in the field of analytics. To show the importance of this algorithm we will use the department of Herault (Montpellier-France), although specifically, through the company "Autour.com" that handles every day huge files of its customers. The application of our algorithm by this company shows that people living next door (applicants and suppliers) can be put into connection; is a kind of creation of a district social networking. The remainder of this paper is organized as follows. The authors in section 2 discuss an integrative literature review on the localization algorithms. Our algorithm is explained in section 3. Finally, section 4 concludes this paper.

2 Literature Review

For several years, localization issues become more and more a challenging subject to our dynamic era. For this, a wide range of studies was performed to characterize the performance of these systems in different environments. Going from presentation of research capabilities of an interest point to the positioning of the user and its true location on the map [1]. That presents an overview of tools and documentation to better understand the city either in whole, or in its environment through its districts. The benefits of grid mapping are well known as the readability of the gridded map enables detailed analysis of intra-urban dynamics. If these benefits are common to all types of grid, two families can be distinguished: the grid resulting from a breakdown of data collected in any zoning (zonal data transfer) and the one resulting from the automatic addressing (transfer point data). The interpretation of aerial photographs from 1986, based on a grid of 125 m side, allowed Dayre and Mazurek [6] to analyze the land use on the urban district Montpellier. This method offers many possibilities for automated processing of data, especially regarding the dynamics of built areas.

As part of a mapping grid, in [5] authors studied the disintegration of zonal variables, returning on the assumption of the spatial distribution of populations. The aim is to approach differentiated rules disaggregation variables by merging morphological nature of information plans, topological and environmental. Modeling of Antoni [2] appears as an interesting tool to better understand urban sprawl and to simulate various useful development scenarios for development. The methodological approach presented by the author combines three steps. Each step corresponds to a model:

- The first quantifies the spreading,
- The second locates it
- The third differentiates it.

The three stages are associated with a spatio-temporal database that uses the grid to store all the necessary information in a single GIS layer.

In recent years, many researchers have proposed solutions to the positioning system implementation based on different technologies, such as systems based on GPS [11], the ultrasonic sensor [1], video cameras [4] and the systems based on radio frequency identification. These different approaches and technologies offer different ways to address the problem of locating and monitoring in real time. Localization is a broad and active research area, and a diverse set of solutions have been proposed. Significant research has been done using location data of mobile users. Some were fundamental research such as Song et al.'s [8] work on identifying patterns of human mobility. Some [9] focused on building new services that may have great public or business potentials, such as modeling city living neighborhood by Cranshaw et al. [3] and recommending friends and locations [10].

Our work also benefited from excellent works analyzing and making use of INSEE on the signs of the Toulouse Diversity in 2008, but this study do not focus on the usual zoning (town, district) but on a continuous territorial coverage formed tiles. For us the interesting applications are those that contain devices allowing the user to enjoy all the benefits of positioning and location services. Recently, the realization of a location model involves:

- The availability of an infrastructure containing all the data needed to acquire the necessary information.
- Determining a reference against the position to locate.
- The treatment of acquired data and extracting the necessary information to determine position.

Our algorithm is inspired by related work in geo-localization and big data analytics. It offers solutions that can be deployed to large population and used for mobile proximity marketing or social networking services or why not in the field of health. The details of this method are discussed in the previous sections.

3 Presentation of "The Snail's Algorithm"

In this section we will present our localization algorithm, which consists of three phases:

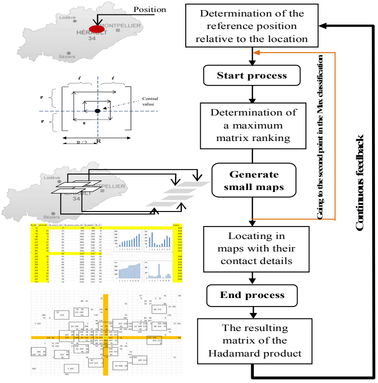

Position
Lodève
MONTPELLIER
HÉRAULT
34
Béziers
Central value
P
R/2
R
Determination of the reference position relative to the location
Start process
Determination of a maximum matrix ranking
Generate small maps
Locating in maps with their contact details
End process
The resulting matrix of the Hadamard product
Going to the second point in the Max classification
Continuous feedback

Fig. 1 Snail algorithm process

We are interested to the presentation of a point defined by a position on a plan or map that could be helpful to the user. And the most effective way for a retail designer map is to define the geographic coordinates in the coordinate system (X,Y). It should be noted that the situation of a point is expressed as coordinates in a reference geo-localization system in terms of territorial coverage and the quality of resulting data.

3.1 Processing Phase

First we must cut out the Local Area Map 3 km away into square (piece).

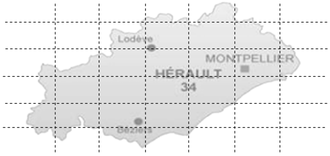


Fig. 2 Cutting out the map of Herault

Then, to start searching it must first define the position. We start from the point which is the central value of the indicated positions, around which rotates on a radius of 3 km range.

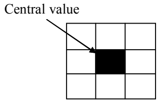


Fig. 3 The central value for starting

The first step is to determine a ranking of maximum matrix, it is not necessary that the starting point or the central value is a Max card, and then determine a point (x: for example) than 1000 people, in a square of 200 m on both sides, which makes a first square matrix, and each time there is a superimposed array. All that to find the median matrix “ M2p +1.2 p +1”.

3.2 Generate Small Maps

In which the target region (map Herault department) is split into small grids. On a central point (max) removing the map is going to have a “millefeuille” on geo-location we will find small maps localized that bring together applicants and suppliers.

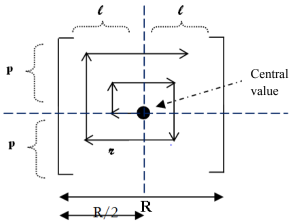


Fig. 4 Graphic presentation of the snail algorithm

Then, the same operation each time passing the second point in the ranking of Maxs, but there will be some cases where the second matrix in the ranking of maximum points will be included in the first round. It should be noted that used the Hadamard matrix product, which represents a binary operation for two matrices of the same dimensions, combines another matrix of the same size where each coefficient is the term by term product of the two matrices. Just like the addition, the Hadamard product is defined only for the matrices of the same dimensions, and the product of two matrices A and B is defined by:

A*B= [ai,j*bi,j]=

$$\begin{pmatrix} a1,1*b1,1 & a1.2*b1,2 & \ldots\ldots\ldots & a1,j*b1,j \\ a2,1*b2,1 & a2.2*b2,2 & \ldots\ldots\ldots & a2,j*b2,j \\ . \quad . & . \quad . & \ldots\ldots\ldots & . \quad . \\ . \quad . & . \quad . & \ldots\ldots\ldots & . \quad . \\ . \quad . & . \quad . & \ldots\ldots\ldots & . \quad . \\ ai,1*bi,1 & ai.2*bi,2 & \ldots\ldots\ldots. & ai,j*bi,j \end{pmatrix}$$

3.3 *Get the Results*

The previous phase will generate little maps that facilitate the research. So, instead of working on full maps we can have small maps. These small cards will be identified quickly with their contact information. But every time we finish analyzing a sub card the algorithm is spirit to pit the matrix (we replace by zero). The resulting matrix of the Hadamard product between two same-sized matrices contains the result of a multiplication element by element. The navigation data are determined by setting a plane tangent to a fixed point on the map of Herault. This method is a technique for mapping a set of points in a multidimensional space, using a matrix of distances calculated in the departure space. Items placed on the borders of central calculate their coordinates iteratively until it converges to the Max of Maxs, and in this state there is performed the rotation around this point to calculate the product. That is to say, one seeks the Max throne the matrix "A" and removes the matrix "B".

$$\begin{tikzcd}
MA_1 \arrow[r] \arrow[d] & M_1A_2 \arrow[r] \arrow[d] & M_2A_3 \arrow[r, "\cdots\cdots"] \arrow[d] & M_{n-1}A_n \arrow[d] \\
B_1 & B_2 & B_3 & B_n
\end{tikzcd}$$

Fig. 5 transition from the first to the last matrix

Such as:
A1 is the first matrix with enthroned extraction B1, An is the first matrix with enthroned extraction Bn. Note that "An" that represents the last enthroned matrix is a zero matrix because it always replaces the zero value as follows:

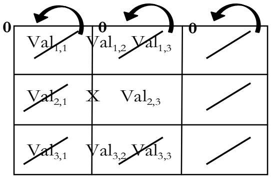


Fig. 6 The last matrix form

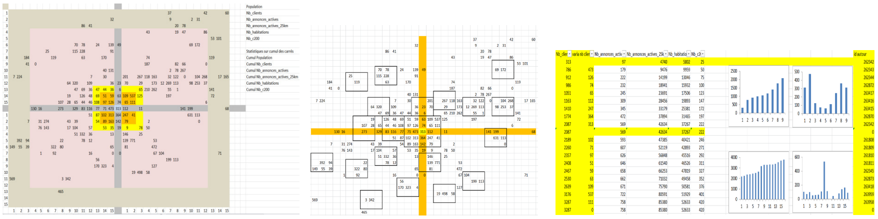


Fig. 7 Sample Application

The great quest data is primarily aimed at understanding what customers' need, what works, what could be improved. Concretely it is to quickly collect raw data, explore and analyze, translate data into actionable information, and therefore globally to reduce the time between the discovery of relevant facts, the characterization of business opportunity and the outbreak of shares. The use of this algorithm reduces the information and develops a methodology based on the interaction data from several sources to provide reliable positioning and determining a navigation method. The Big data is becoming a tool to help companies explore new territories [7]. For these gold mines can reveal intrinsic value which was not necessarily expected to start the analysis.

3.4 Application with: "autour.com"

Identify the number of limited sends from a message packet (mails) that rotates around the center point (median matrix) to make offers and requests; such was the purpose of the company autour.com. The objective of the establishment of such algorithm was to bring people (suppliers and demanders) especially those who live next door. The intent is to create a local social network mixing ads and messages, allowing individuals, local shops, public institutions and associations to explore the activity around their homes, communicate with their neighbors and to know better the resources of their neighborhood. "Autour.com" is a cooperative society of collective interest since the end of 2014. In addition to the "Autour.com" website, the company publishes the sites "j'announce", "Annonces Vertes" and "G-tout". This application was previously incubated by "Alter'incub", incubation of social enterprises in Languedoc-Roussillon. Know and meet their neighbors, helping each other, exchanging objects or services or to make some money by renting or selling property is what Autour.com offers to its members through use of algorithm.

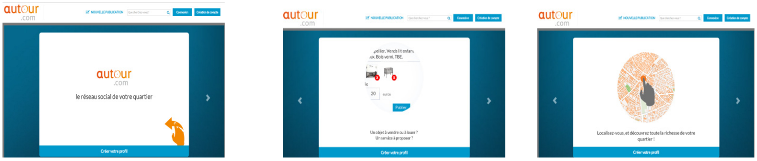

autour .com
NOUVELLE PUBLICATION
Connexion
Création de compte
autour .com
le réseau social de votre quartier
Créer votre profil
...pellier. Vends lit enfant ...ux. Bois verni, TBE.
20 euros
Publier
Un objet à vendre ou à louer ?
Un service à proposer ?
Créer votre profil
Localisez-vous, et découvrez toute la richesse de votre quartier !
Créer votre profil

Fig. 8 autour.com platform

But it is also an opportunity to promote a different type of consumption, sustainable and for social cohesion, focusing on rental, lending or exchange of goods or services. Autour.com today has over 15,000 profiles mainly in Montpellier but also throughout France as the national network is now.

4 Conclusion

The algorithm just presented is only one example enabling companies to monetize their data. In this paper, we open a new application area for people living in the same area or district. This research provides a possible solution to the growing quantity of data. This work has immediate business implications with autour.com. The results of the application of our algorithm with this company show that our algorithm has a better performance, but it needs to be developed, with integration of many other operations.

This algorithm is a means that enables the company to better monetize Big data is a matter of understanding the model it wants to follow to define its way into the big data. For that currently have data on its customers or in the market is a competitive advantage. Finally, we wish that we can develop our algorithm and extend its advantages, in the field of health for example, recently we had a contact with a startup company in Morocco in order to develop this method in the field of health. This small company wants to create "moving parts" for kidney dialysis, with the integration of all necessary the tools and techniques in semi trailers and go to the patient.

References

1. Addlesee, M., Curwen, R., Hodges, S., et al.: Implementing a Sentient Computing System. IEEE Computer **34**(8), 50–56 (2001)
2. Antoni, J.P.: Urban sprawl modelling: A methodological approach, Published in Cybergeo, European Journal of Geography, 2002, Topics, 12th European Colloquium on Quantitative and Theoretical Geography, St-Valery-en-Caux, France, September 7–11, 2001
3. Cranshaw, J., Schwartz, R., Hong, J.I., Sadeh, N.: The livehoods project: utilizing social media to understand the dynamics of a city. In: Proceedings of ICWSM 2012 (2012)
4. Hoey, J., Chantier, M.J., Trucco, E., et al.: Tracking using flock of features, with application to assisted handwashing, British Machine Vision Conference BMVC, pp. 367–376 (2006)
5. Langlois, P., Lajoie, G.: Cartographie par carroyage et précision spatiale. Mappemonde **49**(1), 20–23 (1998)
6. Mazurek, H., Dayre, P.: Analyse de l'utilisation du sol par la méthode du carroyage: le District Urbain de Montpellier, Mappemonde, no 88/3, GIP. RECULS, pp. 27–30 (1988)
7. Monino, L.L., Sedkaoui, S., Matouk J.: Big data, éthique des données, et entreprises, Les Cahiers du CEDIMES, Dossier "Economie et gouvernance" **8**(2) (2014). http://cedimes.com/images/documents/cahiers_cedimes/cahier_cedimes_vol_8_N2_2014.pdf
8. Song, C., Qu, Z., Blumm, N., Barabási, A.L.: Limits of Predictability in Human Mobility. Science **327**(5968), 1018–1021 (2010)

9. Zhang, D., Guo, B., Li, B., Yu, Z.: Extracting social and community intelligence from digital footprints: an emerging research area. In: Ubiquitous Intelligence and Computing, pp. 4–18 (2010)
10. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.Y.: Recommending Friends and Locations Based on Individual Location History. ACM Transactions on the Web (TWEB) **5**(1), 5 (2011)
11. Zhou, J., Shi, J.: RFID localization algorithms and applications—a review. Journal of Intelligent Manufacturing **20**(6), 695–707 (2008)

Structure and Operation of a Basic Genetic Algorithm

Francisco João Pinto

Abstract This work describes the structure and the operation of a basic genetic algorithm. The studies show that the genetics algorithms (GAs) always offer an answer that tends to be the best over time, satisfied with knowledge on the problem, we can improve the function of evaluation that was always search of inside the current population those solutions that possess the best characteristic and tries to combine them of form to generate solutions still better and the process is repeated until we have obtained a solution for our problem**.**The (GA) go in the scene to resolve those problems whose exact algorithms are extremely slow or unable to obtain a solution.

Keywords Structure and operation · Basic genetic algorithm · Artificial intelligence

1 Introduction

The (GAs) are a technician used in problems of search/discovery. The designation results that the appearances to discover are represented in a structure of data, on which will go to become operations like crossing and mutation, of form that, interaction after interaction, the structure go encoding better «values», for the aims of the problem. This form to reach the aims (by evolution), by means of operations on a structure encoded (chromosome), has high parallel with what happens in the nature, satisfied studied in genetics. Also the theory of the natural selection, of Charles Darwin, is important to comprise the evolution in (GAs). In accordance with this theory, in a population of individuals, survive and reproduce more, the best adapted in half environment; that is to say, the aptest, for the problem.

F.J. Pinto(✉)
Department of Computer Engineering, Faculty of Engineering,
University Agostinho Neto, University Campus of the Camama, S/N, Luanda, Angola
e-mail: fjoaopinto@yahoo.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_6

On hire purchase, the no adapted will extinguish .In the archipélago of the Galápagos, Darwin found, in next islands, different species, with exclusive characteristics, for the peculiarities of his habitat, like birds whose beaks are specialised in some seeds. These beaks were the result of a lot of generations, that were reinforcing the expression of the characteristic.

To notice that the theory of Darwin was attacked by not achieving answer to (1) as they arise the variations in the individuals and to (2) how is that the individuals happen his characteristics to the descent. Only the 20th century, with the Genetics, is that these questions would be answered: living beings have, in the core of his cells, arrangement of DNA, called of Genes, that determine his characteristics. A chromosome is a sequence of genes. During the sexual production, the chromosomes of the progenitors cross , generating new chromosomes**.** With low probability, also can become mutations**,** that are alterations chromosomal haphazard, that can confer advantages (cases of the borbuletas black of Manchester), disadvantages (syndrome of Down), or not having expression. In (GAs), all these concepts are present: chromosomes, characteristic encoded, reproduction sexual, crossing, mutation, individual, function of evaluation of aptitude of an individual (fitness function), population and generation.

In Historical terms, the fatherhood of the AGs is attributed to Jonh Holland, mainly after his book Adaption and Artificial Systems (1975). In terms of positioning, the (GAs) can consider evolutionary computation, that is a bouquet of the artificial intelligence. Holland pretended to improve the understanding of the process of natural adaptation and conceive artificial systems, that manifested it. The main application are problems of search and of optimisation, although they can not guarantee optimum solutions, because of heuristic nature of the objective function, that is explained of followed.The problem of the strictly analytical methods, very effective for problems of optimisation, is that the «reality» can not being easy to describe, with mathematical functions and determinative – think in the class of problems NP-hard (non deterministic polynomial, with answer no necessarily binary).

A concept key in (GAs) is the of objective function, or fitness function. This function receives like argument an individual of the population and gives back his aptitude, relatively to the remaining individuals. While more elevated the fitness of «somebody», greater is his probability of reproduction. The objective function serves like this to orient the algorithm in the space of search, as the space of search (or space of states) is the collection of possible individuals, where, interaction after interaction, If it will have to assist to the deactivation of the worse solutions evaluated and to the reproduction of the ones of fitness upper, of form to that go «activating» better chromosomes, Marques (2005).

2 Structure of a Basic Genetic Algorithm

1) Start: generate a random population initial. This population can be seen like a collection of chromosomes, as the individuals are reduced to the representation of his notable characteristics for the problem (chromosome);

2) Fitness: Evaluate each chromosome, by the function of fitness;
3) New population: produce a new population (or generate, or descendants), by execution of the following steps, so many times, those that the new individuals pretended;
   a. Selection: select two chromosomes for crossing, respecting that while more fitness greater his probability of selection.
   b. Crossing: The chromosomes of the parents have to cross somehow.
   c. Mutation: consider, with low probability, the application of mutation in some position of the descendants chromosomes.
   d. To accept: to accept the descendant and place it in the new population.
4) Replacement: To substitute the old population by the new population, generated in the step 3.
5) To test:
   a. To test the condition of paragem of the algorithm;
   b. If satisfied, finish, producing how «better solution» (not confusing like optimum solution) the common population; of the contrary, continue .
6) Goto 2

The algoritim genetic basic presented, left questions in open. Like encoding chromosomes? It depends entirely of the problem. A very frequent option is to do a binary coding, in that each bit corresponds to the (no) activation of a characteristic and opted by this representation in the problem that follows. Like doing crossing and mutation? Crossing and mutation are the operators of the (GAs). The exert of any (GA), is above all else, influencied by the form like these operators comport. The most usual form to do crossover, consists in choosing a random point of crossing and afterwards joint the genes the left of the father with the genes to the right of the mother. In the representation down, the vertical bar (|) represents the point of crossing, Goldberg(1989).

Table 1 Crossing of chromosomes

Father / chromosome #1:	**11011 | 00100110110**
Mother / chromosome #2:	**11011 | 11000011110**
Son #1 / chromosome descendant #1:	**11011 | 11000011110**
Son #2 / chromosome descendant #2:	**11011 | 00100110110**

It is possible to do crossing multi-point and with changes of genes more elaborated, all depending on the coding for the problem. The mutation only has to occur after the crossing, satisfied described in the basic algorithm. The aim is to diminish the probability of optimum venues, motivated by the fault of genetic diversity. To do mutation it is necessary to select how many and which the gene to change and afterwards change them. In the case of a binary coding, this corresponds to transform some zeros in ones, and viceversa. It is important to comprise that the mutation has a variable violence, that is to say, during the production of the same new population, can affect more genes in some individuals, of the that others. It follows an example, in that the mutation are distinguished with yellow bottom.

Table 2 Mutations of genes

Falter / chromosome #1:	**1101100100110110**
Mother / chromosome #2:	**1101111000011110**
Son #1 / chromosome descendant #1:	**1100111000011110**
Son #2 / cromossoma descendant #2:	**1101101100110110**

The technician of mutation is dependent of the structure of data of the chromosome. The basic parameters of an (AG) are probability of crossing, the probability of mutation, and the size of the population.

3 Operation of a Genetic Algorithm

We go from a function f(x) very simple

$$f(x) = x^2 \text{ (in other words, x squared)}$$

Imagine that you want to find the value of x which makes the function f (x) reaches its maximum value, but restricting to the variable x to take values ranging between 0 and 31. Further, x you will only allow to take integer values, i.e.: 0, 1, 2, 3,..., 30, 31. Obviously the maximum is for x = 31, where f is worth 961. We don't know the genetic algorithms to solve this problem, but its simplicity makes the algorithm easier to understand. The first thing you must do is to find a way to encode junkies (possible values of x). One way to do it is with binary codification. With this encoding a possible value of x is (0, 1, 0, 1, 1).How do you interpret this? Very simple: multiply the last component (1) 1, the next to last (1) by 2, the previous (0) by 4, the second (1) 8 and the first (0) by 16 and then get the sum: 11. Observed as (0, 0, 0, 0, 0) is equivalent to x = 0 and (1, 1, 1, 1, 1) is equivalent to x = 31. Each possible value of the variable x in binary representation we are going to call individual. A collection of individuals is referred to as population and the number of individuals who compose it is the size of the population. Once we have coded the solution, we must choose a size of population. For this example we will choose 6 individuals, Goldberg(1987).

We must start from an initial population. One way to generate it is randomly: pick up a coin and throw it in the air; if it goes face, the first component of the first individual is a 0 and a 1 otherwise. Repeat the launch of the currency and we will have the second component of the first individual (a 0 if it goes face and a 1 if you exit cross). Thus up to 5 times and get the first guy. Now repeat the above sequence to generate the remaining individuals of the population. In total, you have to throw 5 * 6 = 30 times the coin. Our next step is to compete each other individuals. This process is known as selection. Table 3 summarizes the process, Alander(1992).

Table 3 Selection

1	(0,1,1,0,0)	12	144	6
2	(1,0,0,1,0)	18	324	3
3	(0,1,1,1,1)	15	225	2
4	(1,1,0,0,0)	24	576	5
5	(1,1,0,1,0)	26	676	4
6	(0,0,0,0,1)	1	1	1

Each row in table 1 is associated with an individual of the initial population. Yorey of each column in the table is as follows:

(1) = number that we assign to the individual.
(2) = individual in binary encoding.
(3) = value of x.
(4) = f (x) value.

Notes that the best individual is 5 with f = 676. Calculates the average of f and get f average = 324.3..

As for column (5) now I explain you. One way of carrying out the selection process is through a tournament between two. Each individual in the population is assigned a partner and among them sets a tournament: best generates two copies, and the worst is discarded. The column (5) indicates the couple assigned to each individual, which has realised randomly. The column (5) indicates the couple assigned to each individual, which has realised randomly. They exist a lot of variants of this process of selection, although this method costs us to illustrate the example. After realising the process of selection, the population that have is the showed in the column (2) of the table 2. It observes, for example, that in the tournament between the individual 1 and the 6 of the initial population, the first of them has received two copies, whereas the second falls in the forget, Perez(2005)..

Table 4 Crossing

(1)	(2)	(3)	(4)
1	(0,1,0,1,0)	10	100
2	(1,1,1,0,0)	28	784
3	(0,1,1,1,0)	14	196
4	(1,0,0,0,0)	16	256
5	(1,1,0,1,0)	26	676
6	(1,0,0,1,0)	18	324

After realising the selection, realises the crossing. A way to do it is by means of the crossing 1X: They form couples between the individuals randomly of similar form to the selection. Dice two individuals couple establishes a point of random crossing, that is not more than a random number between 1 and 4 (lalongitud of the individual less 1). For example, in the couple 2-3 the point of crossing is 3, what means that a son of the couple conserves the three first bits of the father and inherits the two last of the mother, whereas the another son of the couple conserves the three first bits of the mother and inherits the two last of the father. The resultant population shows in the column (2) of the table 4.

Table 5 Population after the crossing

(1)	(2)	(3)	(4)
1	(0,1,1,0,0)	5	1
2	(0,1,1,0,0)	3	3
3	(1,0,0,1,0)	2	3
4	(1,0,0,1,0)	6	1
5	(1,1,0,1,0)	1	1
6	(1,1,0,1,0)	4	1

In the column (3) have the value of x; in the following have the value of corresponding. Fix you in that now the maximum value of f is 784 (for the individual 2), whereas before the selection and the crossing was of 676. Besides f average has gone up of 324.3 to 389.3. What wants to say this? Simply that the individuals after the selection and the crossing are better that before these transformations.The following step is to go back to realise the selection and the crossing taking like initial population the one of the table 5. This way to proceed repeats so many times like number of interactions you fix. And which is the optimum? In reality a genetic algorithm does not guarantee you the obtaining of the optimum but, if it is very built, will provide you a reasonably good solution. It can that you obtain the optimum, but the algorithm does not confirm you that it was it. So remain you with the best solution of the last interaction. Also it is good idea to save the best solution of all the previous interactions and at the end remain you with the best solution of the explored. Costa (2008).

4 Conclusions

The (GAs) are systems of natural inspiration especially adapted for problems of search, NP-hard. They can not guarantee optimum, but can be used in problems of optimisation. In real problems in which they apply the genetic algorithms, exists the tendency to the homogeinization of the population, that is to say that all the individuals of the same are identical. This prevents that the algorithm follow exploring new solutions, with what can remain stagnated in a local minimum no very good. They exist technicians to counter this “derives genetic”. The most elementary mechanism, although no always sufficiently effective, is to enter a mutation after the selection and the crossing. Once that have realised the selection and the crossing choose a determinate number of bits of the population and alter them randomly. In our example consists simply in changing some(s) bit(s) of 1 to 0 or 0 to 1.

References

1. Alander, J.T.: On aptimal population size of genetic algorithms. In: Procedings CompEuro 1992, Computer Systems and Software Engenering, 6th Annual European Computer Conference, pp. 65–70 (1992)
2. Costa, E., Simões, A.: Inteligência Artificial-Fundamentos e Aplicações, Capítulo 6, página 294, FCA-Editora informática, Lda, Lisboa-Portugal (2008)
3. Goldberg, D.E., Richardson, J.T.: Genetic Algorithms with sharing for multimodal function optimization. Genetic Algorithms and their Aplications. In: Procedings of the Second International Conference on Genetic Algorithms and their Aplication, pp. 41–49 (1987)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
5. Marques, A.: Introdução aos algoritmos genéticos, Compilação Brasileira (2005)
6. Perez, M.A.M.: Funcionamiento de un algoritmo genético, Grupo de Ingeniería de Organización, Universidad de Sevilla (2005)

Comparison Study Between Chinese Family Tree and Occidental Family Tree

Elton Sarmanho Siqueira, Patrick Cisuaka Kabongo and Li Weigang

Abstract Family trees are one of the use efficiency data structures to present and store the information. There are two kinds of mechanisms to present the relationship of the elements: Occidental Family Trees (OFT) and Oriental Family Trees, especially, Chinese Family Trees (CFT). This paper analyzes the efficiency of these two kinds of trees in the context of relationship presentation and extraction of information. Using the developed OFT and CFT presentation models and search algorithms, the paper conducts the comparison of search performance between OFT and CFT search algorithms. The study reveals that the computational cost is higher in CFT model, but it provides a greater gain information and produces in details the relationships between the individuals in the family trees for information retrieval applications.

Keywords Genealogical information · Family tree · Family relationship · Search algorithm

1 Introduction

A family tree is a chart representing family relationships in a conventional tree structure. There are vast applications of family trees in age of internet. The first important issue is to construct efficiency mechanism to present the relationship of the data. And then, the high performance search algorithms need to be developed as the common ways to trace the genealogy of a certain information of an element such as a person or a user in social network. In addition, family trees can also be useful in administration, medical and anthropological studies [1].

E.S. Siqueira · P.C. Kabongo · L. Weigang(✉)
Department of Computer Science, University of Brasilia (UnB),
Campus Universitário Darcy Ribeiro, Brasilia Cx Postal 4466, Brazil
e-mail: eltonss@ufpa.br, {patrick,weigang}@unb.br

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_7

Unfortunately, the depiction of relationships in a large family is challenging, as it is generally the case with large graphs. In addition, family trees (or genealogical graphs, as we will call them) are not arbitrary or unconstrained graphs. They have special structural properties that can be exploited for the purposes of drawing and interactive visualization [2].

Although genealogical graphs are often referred to as *family trees*, this is misleading to get useful information. Every individual has a tree of ancestors (sometimes called *pedigree*), as well as a tree of descendants, each of which can be drawn in familiar and easily understood ways. A drawing of both of these trees is sometimes called an *hourglass* chart in the genealogical community, and has been called a *centrifugal view* in the literature [2].

Genealogical relationships have been recorded and depicted for centuries, however the traditional charts appearing in books tend to be simple, and usually showing at most a few dozen individuals. They are often organized around simple patterns such as lineages (e.g. one's father, paternal grandfather, etc.) or a single tree of ancestors, or a single tree of descendants [2]. Many family trees fail to properly encode all the necessary and useful information. There are some models of trees for every type of situation, for example, the Occidental Family Trees (OFT) and Oriental Family Trees. However, the Eastern model has several segments, such as Chinese Family Trees (CFT) and Japanese Family Tree (JFT). Both of them are well defined and have complex structures. The Western model has reduced number of structures compared with Eastern model. For example, OFT has a very simple model and is easy to identify the elements and relationship from its structure. This mechanism, however, tend to ignore some important information or personal relationships. Although the mechanism of presentation of CFT is slightly complicated, and increasing the difficulty of the querying, the expression of complete information makes the query results more complete and consistent with the actual situation.

This paper analyzes the efficiency of OFT and CFT in the context of relationship presentation and extraction of information. Using the developed OFT and CFT presentation models and search algorithms, the paper conducts the comparison of search performance between OFT and CFT search algorithms. The study reveals that computational cost is higher in the CFT model, but it provides a greater gain information and produces in details the relationships between the individuals in the family trees. This will satisfy most applications of information retrieval and others.

2 Preliminary Concepts

2.1 Occidental Family Relationship

Family patterns have changed substantially in Europe over the past fifty years. By the late 20th – early 21st century, a wide variety of family forms and relationships emerged along the married nuclear family with children, as young women and men have increasingly refrained from long term commitments with respect to partnerships and childbearing [3].

Some genealogical community have called for the ability to encode richer information and more kinds of relationships, e.g. foster children, family friends, etc. Increased freedom in a genealogical system would make it approach a general hypermedia system, with a correspondingly general interface. However, we have found as [2] that the constraints imposed by first following a traditional family model inspire interesting design and visualization possibilities.

2.2 *Chinese Family Relationship*

With the influence of Confucianism, the concepts of family kinship are deeply ingrained in Chinese culture. Confucius considered the Cardinal Relationships of Chinese society, which includes: a) Ruler and Subject b) Father and Son c) Elder and Younger brother d) Husband and Wife and e) Friends [4]. The Confucian Chinese family relationship has three main features [5]: **Subordinate, Paternalistic and Hierarchical**. Several important observations from the two family trees (Chinese and Occidental) are described as following: 1) CFT has maternal and paternal lineages distinguished (e.g. a mother's brother and a father's brother have different terms), but OFT do not; 2) The relative age of a sibling relation in the Chinese genealogy is considered. For example, a father's younger brother (Chinese: *shushu*) has a different terminology than his older brother (Chinese: *bobo*); 3) In both family trees, the gender of the relative is distinguished.

2.3 *Description of Family Tree Models*

This work follows a model structure for construction of the algorithms of the CFT and OFT (see Figure 1). The model has the following characteristics: 1) Circle are female individuals and Square is male individuals; 2) The red line means marriage; 3) The descendants are only linked to males. This model uses some fundamental concepts of graph. Such a family tree is connected acyclic graph that can mathematically be expressed $G = (V, A)$ where $V = v_1, v_2, ..., v_n$ is a set of Vertices and $A = a_1, a_2, ..., a_n$ is a set of Edges. The edges have a label (E) identifying a vertex v is

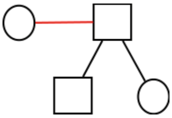


Fig. 1 Model Structure in Family Tree

married, ascendant or descendant of u. For marriage relationship the model has the label $v \xrightarrow{M} u$, it means that vertex v is married with u or u is married with v. For relationship of ascendant the model has the label $v \xrightarrow{A} u$, it means that vertex v is ascendant of u. For relationship of descendant the model has the label $v \xrightarrow{D} u$, it means that vertex v is descendant of u. To determine the relationship of individuals v and u in family tree, it needs to calculate the path $C_{v..u}$ between these two vertices by equation $C_{v..u} = \sum_{i=1}^{n} E_i, \forall i \in \mathbb{N}$.

Breadth-First Search (BFS) Algorithm is used on the Source vertex and Target Vertex to obtain the value of $C_{v..u}$. It's observed that there is only one path between v and u. With $C_{v..u}$ value, it's possible to determine the relationship between v and u using the Table 1 (simplified table).

Table 1 Simplified Table

Relationship (v,u)	**Value**
Father(v,u)	A
Mother(v,u)	AM
Great GrandFather(v,u)	AAA or AMA
Great GrandMother(v,u)	MAAA or MAAMA
Son(v,u)	D or DM
Brother(v,u)	DA
GrandSon(v,u)	DD or DMD
Cousin(v,u)	DDAA or DDAMA

3 Related Work

Family trees have been a topic of interest for researchers. Various approaches for visualizing tree-like structures have been researched.

Keller *et al.* [1] discusses the layout of a family tree that emphasizes temporal data. The ancestors and descendants are laid out radially around a centered person. The layout also supports dynamic interaction with the family tree.

McGuffin *et al.* [2] considered the general problem of visualizing family trees, or genealogical graphs, in 2D. Given a graph theoretic analysis, which identifies why genealogical graphs can be difficult to draw. This motivates some novel graphical representations, including one based on a dual-tree, a subgraph formed by the union of two trees.

Johnson *et al.* [7] describes a novel method for the visualization of hierarchically structured information. The Tree-Map visualization technique makes 100% use of the available display space, mapping the full hierarchy onto a rectangular region in a space-filling manner. This efficient use of space allows very large hierarchies to be displayed in their entirety and facilitates the presentation of semantic information.

4 Family Tree Search Algorithms

4.1 OFT Algorithm

The pseudocode of OFT algorithm was developed with details explaination of interaction in following. To determine the value for variable $path$, the distance between vertices v and u needs to be calculated using Algorithms 2.

The method **Custom BFS** requires that a undirected graph $G = (V, A)$ is passed by parameter and a vertex v from which to start the search. The vertices are numbered from 1 to $n = |V|$, i.e. $V = \{1, 2, \ldots, n\}$. This method has a queue Q and a list D of relationships of all vertices from v. While Q is not empty then captures and removes a element (Dequeue operation). If $w \in adj(v')$ and w is not visited then do $D[w]$ receives $D[v]$ plus $A(w, v)$ and then insert w in Q. The method $A(x, y)$ is the value of relationship (D, A or M) between individuals. Lastly, the method returns the full path between vertices v and u.

The variable $path$ will be parameter in the method $getNameRelashionship$ that must return name of relationship between v and u according to Table 1. This algorithm is with the time complexity of $O(|V| + |A|)$, which is similar to the complexity of BFS algorithm.

Algorithm 1. Occidental family tree algorithm

```
Require: Graph $G(V, A)$, $v$ (Source vertex) and $u$ (Target vertex)
Ensure: $C_{v..u}$
  path = $BFS(G, v, u)$
  relationship = $getNameRelashionship(path)$ {Search in Table 1}
  print relationship
```

Algorithm 2. Custom BFS

```
Ensure: A list $D$ of relationships of all vertices from $v$.
  $Q \leftarrow [v]$; $D \leftarrow [\infty, \infty, \ldots, \infty]$; $D[v] \leftarrow 0$;
  $Enqueue(Q, v)$
  while $|Q| > 0$ do
    $v' \leftarrow Dequeue(Q)$
    while $w \in adj(v')$ do
      if $D[w] = \infty$ then
        $D[w] \leftarrow D[v'] + A(w, v')$
        $Enqueue(Q, w)$
      end if
    end while
  end while
  return $PATH = D[u]$
```

4.2 CFT Algorithm

This subsection shows pseudocode of CFT algorithm with details about interaction. To determine the value of $path$, the distance between vertices v and u should be calculated using Algorithm 2. Obtaining the value of the variable $path$, the algorithm will check the hierarchy (paternal, maternal or none) and age (older, young or none). The variables $side$, age and $path$ are parameters in the method $getNameRelashionship$ that return name of relationship between v and u according with Table 1.

Algorithm 3. Chinese family tree algorithm

```
Require: Graph $G(V, A)$, $v$ (Source vertex) and $u$ (Target vertex)
Ensure: $C_{v..u}$
  path = $BFS(G, v, u)$; side ← ∅; age ← ∅
  if $path$ is paternal then
    $side \leftarrow paternal$
  else
    if $path$ is maternal then
      $side \leftarrow maternal$
    end if
  end if
  $age \leftarrow getHierarchicalByAge(G, v, u)$ {older or young}
  relationship = $getNameRelashionship(path, side, age)${Search in Table 1}
  print relationship
```

The Algorithm 3 has time complexity of $O(|V|+|A|+|M|)$, which is similar to the complexity of BFS added the time complexity of the method $getHierarchicalByAge$. The age of u and v is calculated based on the birth date and compares with M elements (both genders: male and female) that are on the same level in the tree (i.e. brothers, uncles or cousins).

5 Study of Case - Succession of Ascendants

The case of the succession of ascending was selected because it is a real problem concerning the goverment administration in Brazil. The bid of a great project or to establish an office of a congress man, the related paricipants should not be the relatives of the authority that manages this project according to Brazilian Civil Code (Art. 1836 [8]): *In the absence of descendants, the ascendants are called to the succession in competition with the surviving spouse. §1° In the class of ancestors, the nearest degree excludes the most remote, without distinction of lines. §2° Occurring equal in degree and diversity at the same level, the ascendants of the paternal line (or side) inherit half and the other half to the maternal line (side).*

From generic family tree where exist a reference point called **ME**, it is represented by color purple. Both models discussed in this work on generic family tree and extract

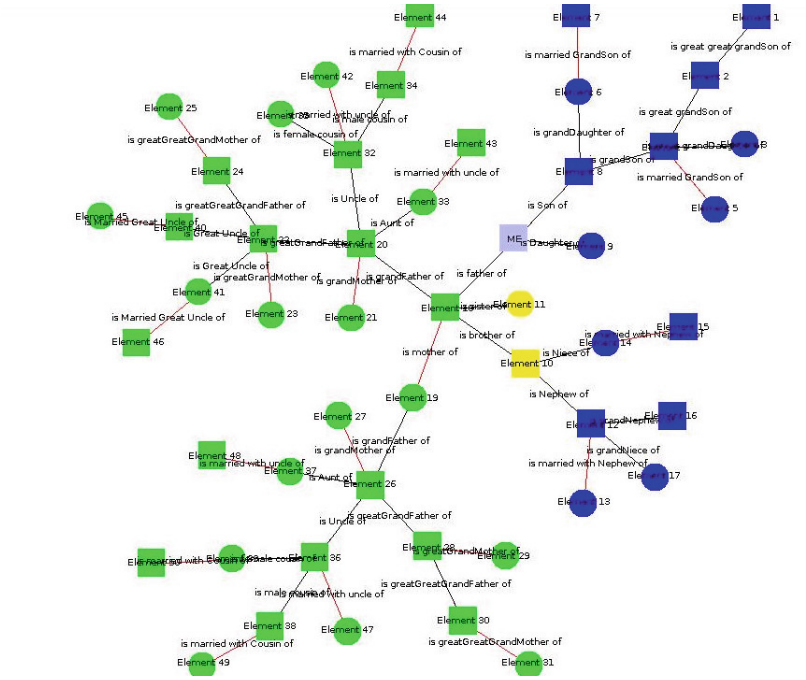


Fig. 2 Occidental Genealogical Graph

the necessary information which can be applied to support the Brazilian Civil Code correctly without errors in the identification of lineages.

First, the application of the OFT model presents the result in a family tree (see Figure 2) where the ascendants are represented by green color, the descendants by blue and the brothers (or sisters) by yellow. There is a label in each edge informing the relation name of each individual with **ME**. In this model, it is observed that do not have distinction between the ascending by lineage and that does not apply correctly the law described in Art. 1836, because not known who is paternal and maternal side this family tree.

However, using the CFT model presents the result in a family tree (see Figure 3) that the descendants are represented by blue color, brothers (or sisters) represented by yellow, the ascendants of maternal side represented by pink and the ascendants of paternal side represented by black. With this model, it is able to apply correctly the law of succession of ascending, because it differentiates the lineages (paternal and maternal). So the law can be applied correctly and no prejudice any of the involved parties. The algorithms over a generic family tree together with $PrefuseFramework$ [6] are applied to get two genealogical graphs, OFT (Figure 2) and CFT (Figure 3) models.

The results show that the computational cost is higher in CFT model, but it provides greater gain information, and produces in details the relationships between the

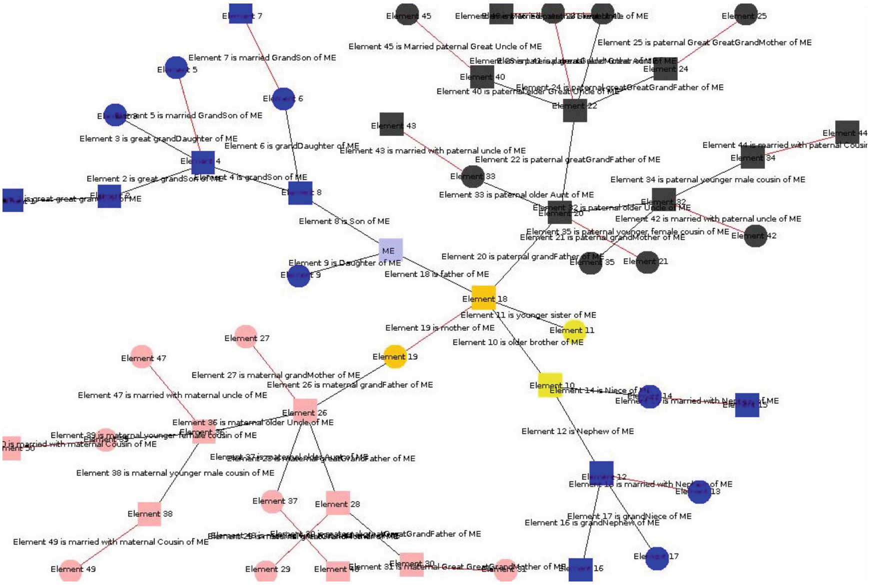

Element 7
Element 45
Element 25
Element 7 is married GrandSon of ME
Element 45 is Married paternal Great Uncle of ME
Element 25 is paternal Great GreatGrandMother of ME
Element 5
Element 6
Element 40
Element 24
Element 40 is paternal older Great Uncle of ME
Element 24 is paternal greatGreatGrandFather of ME
Element 22
Element 5 is married GrandSon of ME
Element 43
Element 44
Element 3 is great grandDaughter of ME
Element 6 is grandDaughter of ME
Element 43 is married with paternal uncle of ME
Element 22 is paternal greatGrandFather of ME
Element 34
Element 4
Element 2 is great grandSon of ME
Element 4 is grandSon of ME
Element 33
Element 34 is paternal younger male cousin of ME
Element 8
Element 33 is paternal older Aunt of ME
Element 32
Element 32 is paternal older Uncle of ME
Element 20
Element 42 is married with paternal uncle of ME
Element 8 is Son of ME
Element 35 is paternal younger female cousin of ME
Element 21 is paternal grandMother of ME
Element 42
ME
Element 20 is paternal grandFather of ME
Element 35
Element 21
Element 9 is Daughter of ME
Element 9
Element 18 is father of ME
Element 18
Element 11 is younger sister of ME
Element 11
Element 19 is mother of ME
Element 27
Element 10 is older brother of ME
Element 47
Element 19
Element 27 is maternal grandMother of ME
Element 26 is maternal grandFather of ME
Element 10
Element 14 is Niece of ME
Element 14
Element 47 is married with maternal uncle of ME
Element 26
Element 36 is maternal older Uncle of ME
Element 39 is maternal younger female cousin of ME
Element 12 is Nephew of ME
Element 38 is maternal younger male cousin of ME
Element 12
Element 13
Element 37
Element 28
Element 38
Element 17 is grandNiece of ME
Element 16 is grandNephew of ME
Element 49 is married with maternal Cousin of ME
Element 17
Element 49
Element 29
Element 30
Element 16

Fig. 3 Chinese Genealogical Graph

individuals in the family tree. In a real situation the information about ascendant of the maternal or paternal side can be observed: the CFT model is better than OFT because it separates clearly the maternal and paternal lines, i.e., decreasing the search for a particular element by half of the ascendants.

6 Conclusions

This research is developed to search the elements of successors in a family tree, and to detect the difference between Chinese Family Trees (CFT) and Occidental Family Trees (OFT). By distinguishing the maternal and paternal sides, the CFT is more efficient than OFT regardless of the size of the tree, and even can reduce half of the search time. The CFT model is effective in the application by identify the age relationship between the elements in the tree.

The paper also shows clearly the comparison between OFT and CFT models and presents the fundamental information on both models. From the comparison results, it implies that the CFT model is more computationally costly than OFT model, but it has a higher number of information that helps to identify more clearly the relationships in a genealogy tree. Thus, in situations with the necessary for more accurate information to find relations between individuals in a large scale tree, CFT model has a better solution by providing more detail relationship information in the family tree. The future work may concentrate the deeply study of more complete family trees as case study to show the benefits of application of CFT.

References

1. Keller, K., Reddy, P., Sachdeva, S.: Family Tree Visualization, Berkeley (2010)
2. McGuffin, M.J., Balakrishnan, R.: Interactive visualization of genealogical graphs. In: IEEE Symposium on Information Visualization, INFOVIS 2005, pp. 16–23. IEEE (2005)
3. Olh, L.S.: Changing families in the European union: trends and policy implications. In: United Nations Expert Group Meeting, Family policy development: Achievements and challenges, New York (2015)
4. Woon, Y.F.: Social organization in South China, 1911-1949: the case of the Kuan lineage of Kai-ping County, no. 48. Center for Chinese Studies, The Universi (1984)
5. Wang, C.: Confucian Chinese Family Relationship and the Obama Family (2012). http://courses.ischool.berkeley.edu/i202/f12/node/662
6. Heer, J., Card, S.K., Landay, J.A.: Prefuse: a toolkit for interactive information visualization. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 421–430. ACM (2005)
7. Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the IEEE Conference on Visualization, Visualization 1991, pp. 284–291. IEEE (1991)
8. Brazil. Art. 1836 of Civil Code - Law 10406/02, 1st edn. Revista dos Tribunais, São Paulo (2002)

Combination of Trees for Guillain-Barré Subtype Classification

Juana Canul-Reich, Juan Frausto-Solis, José Hernández-Torruco and Juan José Méndez-Castillo

Abstract Guillain-Barré Syndrome (GBS) is an autoimmune neurological disorder characterized by a fast evolution. Complications of this disorder vary among the different subtypes. In this study, we use a real dataset that contains clinical, serological, and nerve conduction test data obtained from 129 GBS patients. We apply three different decision tree classifiers: C4.5, C5.0 and random forest to predict GBS subtypes in two classification scenarios: four subtype classification and One vs All (OVA) classification. We evaluate performance under train-test scenario. Experimental results showed comparable performance among all classifiers, although C5.0 slightly outperformed both C4.5 and random forest. Further experiments are being conducted. This is an ongoing research project.

Keywords Data mining and processing · Ensemble methods · Train-test · Performance evaluation · AUC

1 Introduction

Guillain-Barré Syndrome (GBS) is an autoimmune neurological disorder characterized by a fast evolution, usually it goes from a few days up to four weeks.

J. Canul-Reich · J. Hernández-Torruco(✉)
División Académica de Informática y Sistemas, Universidad Juárez Autónoma de Tabasco,
Cunduacán, Tabasco, Mexico
e-mail: {juana.canul,jose.hernandezt}@ujat.mx

J. Frausto-Solis
Instituto Tecnológico de Ciudad Madero, Av. 1o. de Mayo esq. Sor Juana Inés de la Cruz S/n, Col.
Los Mangos, 89440 Ciudad Madero, Tamaulipas, Mexico
e-mail: juan.frausto@gmail.com

J.J. Méndez-Castillo
Hospital General de Especialidades Dr. Javier Buenfil Osorio, Av. Lázaro Cárdenas 208,
Col. Las Flores, 24097 San Francisco De Campeche, Campeche, Mexico
e-mail: juanmdzdr-neuro@yahoo.com.mx

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_8

Complications of GBS vary among subtypes, which can be mainly Acute Inflammatory Demyelinating Polyneuropathy (AIDP), Acute Motor Axonal Neuropathy (AMAN), Acute Motor Sensory Axonal Neuropathy (AMSAN) and Miller-Fisher Syndrome (MF) [4, 8].

In this study, we investigate the predictive power of a reduced set of only 16 features selected from an original dataset of 365 features. This dataset holds data from 129 Mexican patients and contains the four aforementioned GBS subtypes. We apply three different decision tree classifiers: C4.5, C5.0 and random forest. Random forest and C5.0 are two of the most widely used ensemble methods due to their high performance in classification problems of diverse nature. Also, these two methods differ in the ensemble approach they use. Random forest applies bagging while C5.0 uses boosting. In principle, ensemble learning combines multiple classifiers with the aim of obtaining better predictive performance than that obtained using solely one of the constituent classifiers[10]. We compare the results from these two ensemble methods with those encountered by a single classifier, C4.5. We selected C4.5 as a benchmark classifier due to its competitive performance in classification applications, as well as its simplicity of implementation. Further experiments with other algorithms will follow.

This paper is organized as follows. In section 2, we present a description of the dataset, metrics used in the study, a brief description of the classifiers, experimental design, as well as the C5.0 and random forest tuning procedures. In section 3, we show and discuss the experimental results. Finally in section 4, we summarize results, give conclusions of the study, and also suggest some future directions.

2 Materials and Methods

2.1 Data

The dataset used in this work comprises 129 cases of patients seen at Instituto Nacional de Neurología y Neurocirugía located in Mexico City. There are 20 AIDP cases, 37 AMAN, 59 AMSAN, and 13 Miller-Fisher cases. Hence, there are four GBS subtypes in this dataset. In a previous work [2], we identified a set of 16 relevant features out of an original 365 features. The features are listed in Table 1. The first four features are all clinical and the remaining features come from a nerve conduction test.

2.2 Performance Measures

We used typical performance measures in machine learning such as sensitivity, specificity, error rate, ROC curves and Kappa statistic [3, 9]. Also we included average

Table 1 List of features used in this study.

Feature label	**Feature name**
v22	Symmetry (in weakness)
v29	Extraocular muscles involvement
v30	Ptosis
v31	Cerebellar involvement
v63	Amplitude of left median motor nerve
v106	Area under the curve of left ulnar motor nerve
v120	Area under the curve of right ulnar motor nerve
v130	Amplitude of left tibial motor nerve
v141	Amplitude of right tibial motor nerve
v161	Area under the curve of right peroneal motor nerve
v172	Amplitude of left median sensory nerve
v177	Amplitude of right median sensory nerve
v178	Area under the curve of right median sensory nerve
v186	Latency of right ulnar sensory nerve
v187	Amplitude of right ulnar sensory nerve
v198	Area under the curve of right sural sensory nerve

accuracy and balanced accuracy. The former used in four GBS subtype classification, since it is a more suitable measure for multiclass classification problems [7]. The latter used in OVA classification, because it is a better performance estimate of imbalanced datasets [9].

2.3 Classifiers

In this study, we apply three different decision tree classifiers: C4.5, C5.0 and random forest. Random forest and C5.0 are two of the most widely used ensemble methods due to their high performance in classification problems of diverse nature. Also, these two methods differ in the ensemble approach they use. We selected C4.5 as a benchmark classifier due to its competitive performance in classification applications, as well as its implementation simplicity. A brief description of each method is shown below.

C4.5. C4.5 [6] builds a decision tree from training data using recursive partitions. In each iteration, C4.5 selects the attribute with the highest gain ratio as the attribute from which the tree branching is performed. This results in a more simplified tree.

Random Forest (RF). Random Forest, introduced by Breiman and Adele Cutler [1], builds a bootstrap ensemble of CART trees. The class for a new instance is given by the majority vote of this ensemble.

C5.0. C5.0 is a improved version of C4.5, introduced by Quinlan [6]. Its major improvement is the implementation of boosting, which gives trees higher precision.

2.4 Experimental Design

We used the 16-feature subset, described in section 2.1, for experiments. We added the GBS subtype as class variable. Finally, we created a dataset containing the 129 instances and 17 features. We used train-test evaluation scheme in all cases. We used two-thirds for train and one-third for test. We performed 30 runs where we applied each of the methods listed in section 2.3. For each run, we computed accuracy, sensitivity, specificity, Kappa statistic and AUC. Finally, we averaged each of these quantities across the 30 runs. In each run, we set a different seed to ensure different splits of train and test sets across runs, then we had all classifiers use the same seed at same run. These seeds were generated using Mersenne-Twister pseudo-random number generator [5].

We performed experiments in two classification scenarios: four GBS subtype classification and One vs All classification (OVA). In the first scenario, the four GBS subtypes were included in the dataset at the same time, that is, AIDP, AMAN, AMSAN and MF. For OVA classification, we created four new datasets, as the number of GBS subtypes in the dataset. In each dataset, the instances of one class were marked as the positive cases and the instances of the remaining classes were marked as the negative cases.

2.5 Parameter Optimization/Setting

C5.0. C5.0 requires tuning the number of trials. This tuning was performed by 30 train-test runs using different numbers of trials from 5 to 100. Figure 1 shows the tuning results. The lowest average error rate across 30 train-test runs was obtained with a number of trials = 55. This number of trials was used for all experiments with C5.0.

Random Forest. Random forest has only two tuning parameters: the number of variables in the random subset at each node and the number of trees in the forest. The R implementation used in this work is capable of effectively tuning the first parameter. For the second one, we performed 30 train-test runs using different numbers of trees from 100 to 1000. Figure 2 shows the tuning results. The lowest average error rate across 30 train-test runs was obtained with a number of trees = 700. This number of trees was used for all experiments with random forest.

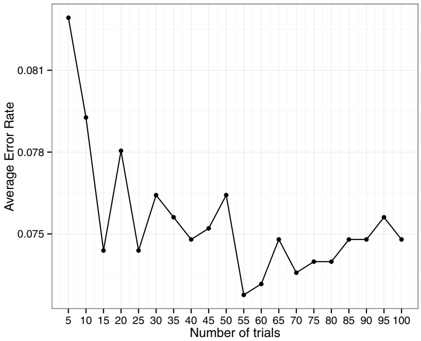


Fig. 1 Number of trials optimization in C5.0.

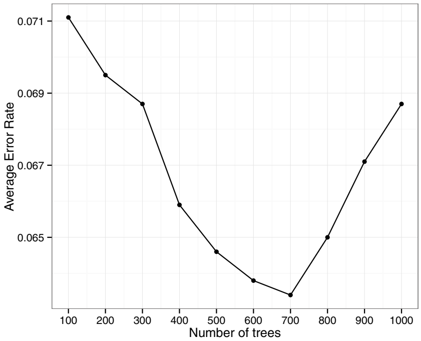


Fig. 2 Number of trees optimization in random forest.

3 Discussion/Results

In this section, we show the results of the experiments in four GBS subtype and OVA classification. All tables show the average results of each classifier across 30 runs for each classification scenario. Also, the standard deviation of each metric is shown.

Table 2 shows four GBS subtype classification. All classifiers obtained an average accuracy above 0.90. The two ensemble methods, random forest and C5.0, outperformed C4.5 in all metrics.

Table 3 shows AIDP vs ALL classification. C5.0 and C4.5 obtained a balanced accuracy above 0.80. Values obtained in specificity were much higher than those obtained in sensitivity in all classifiers. Low values were obtained in Kappa statistic in all classifiers, between 0.6009 and 0.6407. Random forest had the lowest performance in all metrics. Table 4 shows AMAN vs ALL classification. All classifiers obtained a balanced accuracy above 0.90. Specificity was a little higher than sensitivity in all classifiers. Again, random forest obtained the lowest values in all metrics. Overall, the highest classification performance was obtained in AMAN vs ALL.

Table 2 Average results across 30 runs in four GBS subtype classification.

	Average				
Method	**Accuracy**	**Accuracy**	**Sensitivity**	**Specificity**	**Kappa**
RF	**0.9366**	**0.8390**	**0.8120**	**0.9544**	**0.8090**
	0.0245	0.0803	0.0812	0.0178	0.0748
C5.0	**0.9272**	**0.8398**	**0.8126**	**0.9476**	**0.7825**
	0.0251	0.0789	0.0749	0.0191	0.0746
C4.5	**0.9114**	**0.8085**	**0.7727**	**0.9356**	**0.7348**
	0.0283	0.0827	0.0727	0.0218	0.0850

Table 3 Average results across 30 runs in AIDP vs ALL classification.

	Balanced					
Method	**Accuracy**	**Accuracy**	**Sensitivity**	**Specificity**	**Kappa**	**AUC**
C5.0	**0.8171**	**0.9048**	**0.6944**	**0.9398**	**0.6266**	**0.8171**
	0.0976	0.0556	0.1861	0.0565	0.1806	0.0976
C4.5	**0.8042**	**0.8984**	**0.6722**	**0.9361**	**0.6009**	**0.8042**
	0.0902	0.0496	0.1932	0.0611	0.1545	0.0902
RF	**0.7769**	**0.9270**	**0.5667**	**0.9870**	**0.6407**	**0.7769**
	0.0888	0.0306	0.1836	0.0289	0.1645	0.0888

Table 4 Average results across 30 runs in AMAN vs ALL classification.

	Balanced					
Method	**Accuracy**	**Accuracy**	**Sensitivity**	**Specificity**	**Kappa**	**AUC**
C5.0	**0.9136**	**0.9302**	**0.8750**	**0.9522**	**0.8290**	**0.9136**
	0.0487	0.0385	0.0948	0.0461	0.0911	0.0487
C4.5	**0.9067**	**0.9238**	**0.8667**	**0.9467**	**0.8124**	**0.9067**
	0.0548	0.0377	0.1129	0.0416	0.0933	0.0548
RF	**0.9033**	**0.9214**	**0.8611**	**0.9456**	**0.8069**	**0.9033**
	0.0550	0.0406	0.1057	0.0424	0.1006	0.0550

Table 5 shows AMSAN vs ALL classification. All classifiers obtained a balanced accuracy above 0.85. Specificity values were a little higher than those of sensitivity in all classifiers. In this case, C5.0 performed the worst from all classifiers. Table 6 shows MF vs ALL classification. Only C5.0 obtained an average accuracy above 0.80. Specificity was a little higher than sensitivity in all classifiers, although in this case the difference between them was much higher than in previous cases. This classification obtained the lowest results of all classification cases.

Figure 3 shows the median ROC curve values of all the classifiers in OVA classification. That is, from each OVA classification results across 30 runs, the median ROC curve value was selected of each classifier. The median value was selected to illustrate the fairest result across 30 runs. The highest median ROC curve values were obtained in AMAN vs ALL classification, ranging from 0.9083 to 0.9167. Second best was AMSAN vs ALL, ranging from 0.8867 to 0.8993. The worst values were obtained in MF vs ALL, ranging from 0.7236 to 0.8486. In short, the best results were

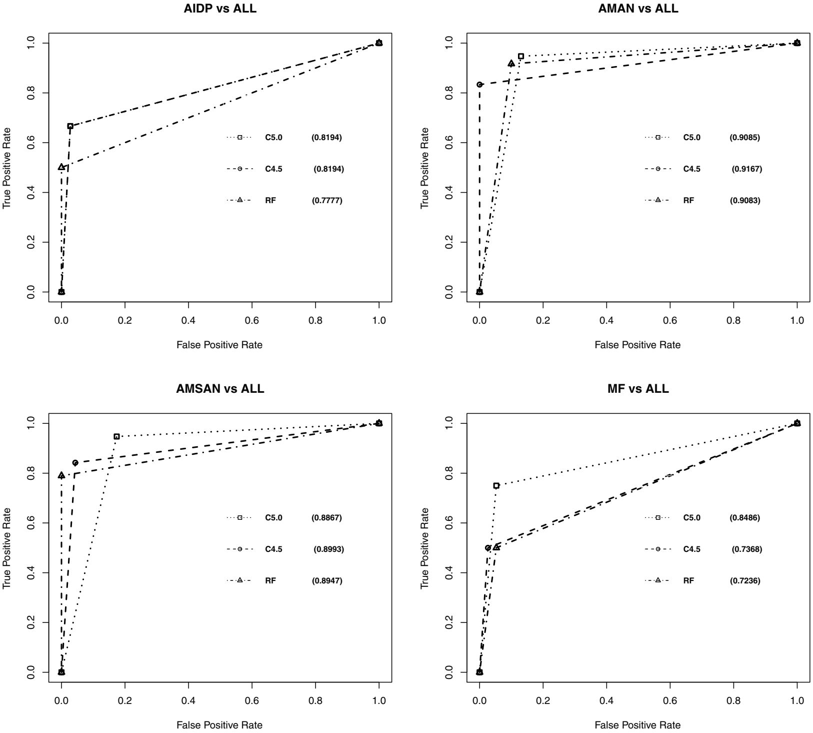


Fig. 3 Median ROC curves in OVA classification.

obtained in the classification cases with the two majority classes present, AMAN vs ALL and AMSAN vs ALL.

Experimental results show the three classifiers are comparable. Nevertheless, it is noteworthy than in three of four OVA classifications, random forest had the lowest performance. Finally, all classification models resulted more specific than sensitive.

Table 5 Average results across 30 runs in AMSAN vs ALL classification.

Method	Balanced Accuracy	Accuracy	Sensitivity	Specificity	Kappa	AUC
RF	**0.8924**	**0.8960**	**0.8544**	**0.9304**	**0.7889**	**0.8924**
	0.0429	0.0417	0.0814	0.0621	0.0849	0.0429
C4.5	**0.8852**	**0.8865**	**0.8719**	**0.8986**	**0.7703**	**0.8852**
	0.0486	0.0450	0.1031	0.0515	0.0929	0.0486
C5.0	**0.8782**	**0.8810**	**0.8491**	**0.9072**	**0.7588**	**0.8782**
	0.0494	0.0480	0.0779	0.0532	0.0978	0.0494

Table 6 Average results across 30 runs in MF vs ALL classification.

Method	Balanced Accuracy	Accuracy	Sensitivity	Specificity	Kappa	AUC
C5.0	**0.8048**	**0.9167**	**0.6667**	**0.9430**	**0.5601**	**0.8048**
	0.1329	0.0487	0.2653	0.0484	0.2145	0.1329
C4.5	**0.7691**	**0.9127**	**0.5917**	**0.9465**	**0.5002**	**0.7691**
	0.1481	0.0416	0.3043	0.0423	0.2489	0.1481
RF	**0.7463**	**0.9254**	**0.5250**	**0.9675**	**0.4973**	**0.7463**
	0.1519	0.0311	0.3104	0.0246	0.2599	0.1519

4 Conclusions

In four GBS subtype classification, we obtained an average accuracy $>= 0.90$ with all the classifiers investigated. In OVA classification, the two majority classes obtained the best classification results, that is AMAN vs ALL and AMSAN vs ALL. Besides, the analysis performed in this work provides insight about the best classifiers for each classification case. From the medical point of view, the reduced number of features used to predict the four GBS subtypes could guide physicians to design a faster, simpler and cheaper diagnosis of the case. However, a predictive model using only clinical variables would be more effective for both patients and physicians since additional studies would be avoided. Currently, we are investigating the building of such model.

References

1. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
2. Canul-Reich, J., Hernández-Torruco, J., Frausto-Solís, J., Méndez-Castillo, J.J.: Finding relevant features for identifying subtypes of guillain-barré syndrome using quenching simulated annealing and partitions around medoids. International Journal of Combinatorial Optimization Problems and Informatics **6**(2), 11–27 (2015)
3. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2012)
4. Kuwabara, S.: Guillain-barré syndrome. Drugs **64**(6), 597–610 (2004)
5. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. **8**(1), 3–30 (1998)
6. Quinlan, J.R.: C4.5: Programs for Machine Learning (1993)
7. Solokova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information processing and management **45**, 427–437 (2009)
8. Uncini, A., Kuwabara, S.: Electrodiagnostic criteria for guillain-barré syndrome: A critical revision and the need for an update. Clinical neurophysiology **123**(8), 1487–1495 (2012)
9. Witten, I.H., Frank, E.. Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann (2011)
10. Zhou, Q.-F., Zhou, H., Yong-Peng, N., Yang, F., Li, T.: Two approaches for novelty detection using random forest. Expert Systems with Applications **42**, 4840–4850 (2015)

Deep Neural Network Architecture Implementation on FPGAs Using a Layer Multiplexing Scheme

Francisco Ortega-Zamorano, José M. Jerez, Iván Gómez and Leonardo Franco

Abstract In recent years predictive models based on Deep Learning strategies have achieved enormous success in several domains including pattern recognition tasks, language translation, software design, etc. Deep learning uses a combination of techniques to achieve its prediction accuracy, but essentially all existing approaches are based on multi-layer neural networks with deep architectures, i.e., several layers of processing units containing a large number of neurons. As the simulation of large networks requires heavy computational power, GPUs and cluster based computation strategies have been successfully used. In this work, a layer multiplexing scheme is presented in order to permit the simulation of deep neural networks in FPGA boards. As a demonstration of the usefulness of the scheme deep architectures trained by the classical Back-Propagation algorithm are simulated on FPGA boards and compared to standard implementations, showing the advantages in computation speed of the proposed scheme.

Keywords Hardware implementation · FPGA · Supervised learning · Deep neural networks · Layer multiplexing

1 Introduction

Neural Networks models have been successfully applied to a wide range of domains in clustering and classification problems in the last three decades [1, 2]. In particular, regarding supervised problems included in the broad area of pattern recognition,

F. Ortega-Zamorano(✉) · J.M. Jerez · I. Gómez · L. Franco
Department of Computer Science, ETSI Informática, Universidad de Málaga, Malaga, Spain
e-mail: {fortega,jja,ivan,lfranco}@lcc.uma.es

F. Ortega-Zamorano
School of Mathematics and Computer Science, University of Yachay Tech,
San Miguel de Urcuquí, Ecuador

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_9

most of the strategies have been based on the utilization of feed forward neural network architectures (FFNN) trained by versions of the well known Back-Propagation algorithm (BP) [3, 4]. One important issue at the time of the implementation of FFNN models is the choice of an adequate architecture [5], that essentially consists of deciding how many hidden layers and neurons to include. It has been observed that the performance of the BP algorithm decreases when a large number of hidden layers are used and so the standard strategy before the irruption of Deep Learning strategies [6] have been to use single hidden layer architectures. Deep Learning is a relatively new technique belonging to the artificial intelligence and machine learning areas that have achieved state-of-the-art results in several recent competitions [7]. There are several approaches for their implementation, as training is a complex process, but in all cases the new characteristic in relationship to previous FFNN approaches is the fact that large (deep and wide) neural network architectures are used. Just to give some numbers, a typical deep learning architecture might include from 5 to 15 hidden layer of neurons with a number of neurons in each hidden layer in the order of the several hundreds or thousands [8]. Training these large networks using standard BP is computationally intensive but also faces the problem of the vanishing gradient problem [9] that makes the training process even slower. To improve the training performance under Deep Learning schemes several strategies have been developed, most of them based on some pre-training phase used to find good starting point synaptic weights from which apply the final supervised phase.

Current implementations of Deep Learning models require the use of parallel strategies to speed up the training process. In this sense alternatives based on cluster computing, GPUs and FPGAs are sensible strategies, each of them having their benefits and drawbacks [10, 11]. Field Programmable Gate Arrays (FPGA) [12] are reprogrammable silicon chips, using prebuilt logic blocks and programmable routing resources that can be configured to implement custom hardware functionality. The main advantage of FPGAs in comparison to PC implementations lies on their intrinsic parallelism but with the disadvantage over PCs and GPUs that they are programmed using VHDL that usually is harder and time consuming. FPGA implementations of neural networks have been analyzed in several studies [13, 14, 15]. Even if recent advances in the computational power of these boards have permitted an increase in the size of the architectures that can be implemented, they are still limited, and in general, the number of layers in the architecture should be prefixed before its application. For this reason, we introduce in this work a layer multiplexing scheme for the on-chip training of deep feedforward neural architectures using the BP algorithm, in which only a single layer of neurons is physically implemented , but this layer can be reused any number of times in order to simulate architectures with several hidden layers, the on-chip learning implementations includes both training and execution phases of the algorithm [15, 16]. Regarding this type of approach, Himavathi et al. [17] have used it previously for neural network training but under an off-chip learning scheme, in which only the synaptic weights of the final model are transmitted to the FPGA that acts as a hardware accelerator.

2 FPGA Layer Multiplexing Scheme Implementation of the BP Algorithm

We describe in this section the layer multiplexing scheme for the Back-Propagation algorithm, which will be divided in 3 different sequential processes: the computation of the neuron output values (S), the calculation of the deltas of each neuron (δ), and the update of synaptic weights. Given the logic of the Back-Propagation algorithm, in which the S values are obtained in a forward manner (from the input towards the output) while the deltas are computed backwards, and that finally the weights updating is executed with the values previously obtained, the three processes are sequentially implemented.

The S values of every layer are obtained as a function of the S values of the previous layer neurons except for those from the first hidden layer which processes the information of the current input pattern. On the contrary, the δ values are computed backwardly, i.e., the δ values associated to a neuron belonging to a hidden layer are computed as a function of the δ values of the a deeper hidden layer, except for the last hidden layer which computes its δ values as a function of the error committed on the current input pattern. The updating process is carried out with the S and δ values of every layer, so it is necessary to store these values when they are computed to be used for the system when they are required. Thus, the structure of the Back-Propagation algorithm allows the whole process to be implemented using a layer multiplexing scheme but nothing that forward and backward phases should be considered separately, as S and δ values cannot be computed in a single forward phase.

The deep design of the Back-Propagation algorithm is based on a layer multiplexing scheme in which only one layer is physically implemented, being reused $3 \times N$ times in order to simulate a whole neural network architecture containing N hidden layers. Fig. 1 shows as in a layer multiplexing scheme the same whole process is carried out but by reusing the structure of the single implemented layer.

The implementation of the layer multiplexing scheme requires a precise control of the layer that is simulated in every moment, and, for this reason, a register called "$CurrentLayer$" is used. For each pattern, the process starts with the forward phase in which the outputs of the neurons are computed in response to the input pattern. This first phase starts by introducing an input pattern in the single multiplexing layer and by setting the variable "$CurrentLayer$" set to 1. Then the neurons' outputs are computed, stored in the distributed RAM memory and transmitted back to the input to calculate the following layer outputs, and thus the variable "$CurrentLayer$" is increased. The same process is repeated sequentially until the "$CurrentLayer$" value is equal to the maximum number of layers, previously defined by the user and stored in the "$MaxLayer$" register. When the last layer is reached the neurons output is computed together with the error committed in the pattern target estimation and these error values are stored in a register for its use in the second phase. The second phase involves the backward computation of the delta values, and the first computation involves the calculation of the delta values of the last layer. Once these

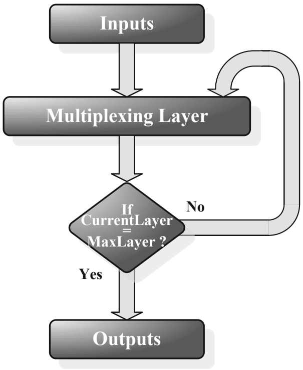


Fig. 1 Layer multiplexing scheme for the simulation of deep feed-forward neural network architectures.

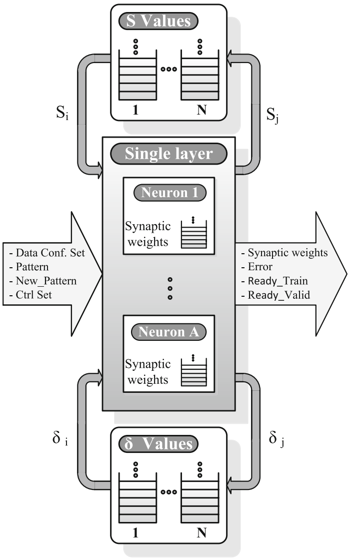


Fig. 2 Schematic representation of the layer multiplexing procedure used for the implementation of the BP algorithm.

values are obtained, they are backwardly transmitted to the previous layer in order to compute the delta values for these set of neurons. With these delta values a recurrent process is used to obtain the delta values of the rest of the layers until the input layer values are obtained ("$CurrentLayer = 1$"). At this point the third phase is carried out in order to update the synaptic weights, and finishing one pattern iteration of the process.

Table 1 Main specifications of the Xilinx Virtex-5 XC5VLX110T FPGA board.

Device	Slice Registers	Slice LUTs	Bonded IOBs	Block RAM
Virtex-5 XC5VLX110T	69,120	69,120	34	148

The Fig. 2 shows a scheme of the architecture block that performs the layer multiplexing procedure for physically implementing a single layer of neurons. This single layer is composed of A neurons blocks implemented in order to compute the neuron‘s output (S) and the δ values, that will later be used for the update of the synaptic weights. The value of A (limited by the board resources) will be the maximum number of neurons for any hidden layer. The neuron blocks manage their own synaptic weights independently of the rest of the architecture, and thus they require a RAM block attached to them. The architecture block also includes memory blocks to store the S and δ values computed for every layer and also for the different input and output signals that are described below.

The input signals are the pattern to be learned, the signal that indicates a new pattern is introduced ($New\_pattern$), the configuration and control data sets, including also the S and δ values. The configuration data set includes the parameters set by the user to specify the neural network architecture, including the number of hidden layers, the number of neurons in each of these layers, learning parameters, etc. The control data set are signals that the control block needs for managing the process of the algorithm to activate the right procedure in every moment. The output signals comprise the output (S) and the δ values for every layer, the training error of the current pattern, and the ready signals for the validation and training processes which are integrated in the control data set.

3 Results

We present in this section results from the implementation of both algorithms (BP and C-Mantec) in a Xilinx Virtex-5 board. Table 1 shows some characteristics of the Virtex-5 XC5VLX110T FPGA, indicating its main logic resources.

Several test cases were analyzed to verify the correct FPGA implementation of the model, comparing the results with those obtained from a PC and with previously published results. These tests were carried out using a 50-20-30 splitting for the training, validation and generalization sets respectively, with a learning rate (η) value fixed to 0.2 in all experiments, and using data from the well-known Iris set.

Table 2 shows the generalization ability obtained for several architectures with different numbers of hidden layers for PC and FPGA implementations. The first column indicates the number of hidden layer present in the architecture, the second column shows the generalization obtained using the PC implementation (mean computed over 100 independent runs), while third and fourth columns shows the results for

Table 2 Generalization ability for the Iris data set for neural network architectures with different numbers of hidden layers for PC and FPGA implementations.

Layers	Type Implementation		
	PC	FPGA	
		Layer Multiplexing	Fixed Layers
1	0.9376	0.9391	0.9406
2	0.9516	0.9442	0.9471
3	0.9518	0.9493	–
5	0.9333	0.9371	–
7	0.8702	0.8842	–
10	0.5273	0.5998	–
15	0.3064	0.3120	–

Table 3 Computation times expressed as a function of the number of hidden layers (X) in the neural architectures for the PC and layer multiplexing FPGA implementations for the cases of including 5 and 20 neurons in each of the layers.

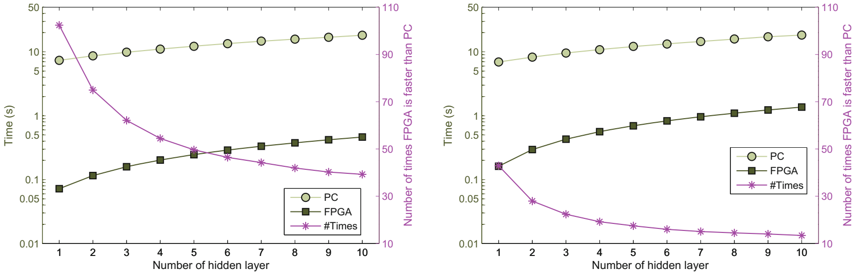
Device	# Neurons	
	5	20
PC	$1.11 \cdot X + 6.25$	$1.26 \cdot X + 5.75$
FPGA	$0.044 \cdot X + 0.028$	$0.134 \cdot X + 0.029$

two different FPGA implementations: the layer multiplexing scheme proposed in this work and the fixed layer scheme utilized in Ref. [18] (only available for architectures with one and two hidden layers). The number of neurons in each of the hidden layers was fixed to five and the number of epochs set to 1000. The maximum number of layers shown in the table is 15 because from this number of hidden layers on the obtained generalization is approximately one third, that is the expected value for random choices for a problem with three classes.

From the results shown in Table 2 it can be seen that the obtained values for generalization are approximately similar for the three implementations considered, and that regarding the number of hidden layers present in the neural architectures the performance of the BP algorithm is relatively stable for architectures with up to 5 hidden neuron layers point from which the generalization accuracy start to decrease to reach the level expected for random choices for a number of layers equal to 15.

Fig. 3 shows the whole learning procedure execution time (in seconds and in logarithmic scale (right Y-axis)) for PC and FPGA implementations a function of the number of hidden layers present in the architecture with five (a) and twenty (b) neurons per layer. The graph also shows a third curve that indicates the number of times (#Times, in linear scale) that the FPGA implementation is faster than the PC one. The number of epochs used was of 1000.

Table 3 shows the results of a linear fitting for the computation time for a variable number of hidden layers, indicated by X in the equations shown. The fitted values were obtained from the cases shown in Fig. 3 for the FPGA and PC implementation in which the number of neurons in each of the hidden layers are fixed to five and twenty.



(a) (b)

Fig. 3 Time and number of times that the FPGA is faster than the PC as a function of the number of hidden layers of the architecture (a) 5 neurons and (b) 20 neurons.

4 Discussion and Conclusions

We have introduced in this work an FPGA implementation for deep neural network architectures using a layer multiplexing scheme. The layer multiplexing scheme used permits to simulate a neural network with several hidden layers by only implementing physically a single hidden layer of neurons. Main advantages of this approach are that very deep neural network architectures can be analyzed through a simple and flexible framework with a very efficient FPGA resource utilization. The implementation has been tested and compared to an existing PC one, obtaining that for a large number of hidden layers the FPGA implementation is approximately 20 to 30 times faster than the PC one. The layer multiplexing scheme used permits in principle the simulation of very deep networks with any number of hidden layers, but memory resource constraints limit the current implementation to approximately hundred hidden layers, that from the point of view of existing Deep Learning models is quite large. The on-chip implementation carried out includes also a validation phase to avoid overfitting effects. Using the Back-Propagation algorithm for training the several hidden layers architectures shown that the performance of the standard BP algorithm starts to degrade when 10 or more hidden layers are present in the architectures, so additional strategies are needed in order to improve the training. In this sense, we believe that the present implementation will facilitate the study of this and related issues, helping to understand very deep neural networks.

Acknowledgements The authors acknowledge support from Junta de Andalucia through grants P10-TIC-5770, from CICYT (Spain) through grant TIN2014-58516-C2-1-R, and from the Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech (all including FEDER funds). And thanks Yachay Tech for financial support for science research.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall PTR, Upper Saddle River (1998)
2. Reed, R.D., Marks, R.J.: Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, Cambridge (1998)
3. Werbos, P.J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University (1974)
4. Rumelhart, D., Hinton, G., Williams, R.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
5. Gómez, I., Franco, L.: Neural network architecture selection: Can function complexity help? Neural Processing Letters **30**, 71–87 (2009)
6. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
7. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61**, 85–117 (2015)
8. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep, big, simple neural nets for handwritten digit recognition. Neural Computation **22**(12), 3207–3220 (2010)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 2010). Society for Artificial Intelligence and Statistics, pp. 249–256 (2010)
10. Suresh, S., Omkar, S.N., Mani, V.: Parallel implementation of back-propagation algorithm in networks of workstations. IEEE Trans. Parallel Distrib. Syst. **16**(1), 24–34 (2005)
11. Huqqani, A.A., Schikuta, E., Ye, S., Chen, P.: Multicore and {GPU} parallelization of neural networks for face recognition. Procedia Computer Science **18**, 349–358 (2013). 2013 International Conference on Computational Science
12. Kilts, S.: Advanced FPGA Design: Architecture, Implementation, and Optimization. Wiley-IEEE Press (2007)
13. Le Ly, D., Chow, P.: High-performance reconfigurable hardware architecture for restricted boltzmann machines. IEEE Transactions on Neural Networks **21**(11), 1780–1792 (2010)
14. Kim, L.W., Asaad, S., Linsker, R.: A fully pipelined fpga architecture of a factored restricted boltzmann machine artificial neural network. ACM Trans. Reconfigurable Technol. Syst. **7**(1), 5–23 (2014)
15. Ortega-Zamorano, F., Jerez, J., Franco, L.: Fpga implementation of the c-mantec neural network constructive algorithm. IEEE Transactions on Industrial Informatics **10**(2), 1154–1161 (2014)
16. Dinu, A., Cirstea, M., Cirstea, S.: Direct neural-network hardware-implementation algorithm. IEEE Transactions on Industrial Electronics **57**(5), 1845–1848 (2010)
17. Himavathi, S., Anitha, D., Muthuramalingam, A.: Feedforward neural network implementation in fpga using layer multiplexing for effective resource utilization. IEEE Transactions on Neural Networks **18**(3), 880–888 (2007)
18. Ortega-Zamorano, F., Jerez, J., Urda Munoz, D., Luque-Baena, R., Franco, L.: Efficient implementation of the backpropagation algorithm in fpgas and microcontrollers. IEEE Transactions on Neural Networks and Learning Systems **PP**(99), 1–11 (2015)

Tracking Users Mobility Patterns Towards CO_2 Footprint

João C. Ferreira, Vítor Monteiro, José A. Afonso and João L. Afonso

Abstract This research work is based on the development of a mobile application and associated central services for tracking users' movements in a city, identifying the transportation mode and routes performed. This passive tracking generates useful data about users' habits, which are then associated with the CO_2 emission in the form of a mobility invoice, with the goal of enabling the users to understand their carbon footprint resulting from the users' mobility process in the city. The performance of the developed system is validated through experimental tests based on data collected during six months from more than 2500 mobility experiences.

Keywords Mobile application · Personalized data · Geographical system · Intelligent public transportation · Carbon footprint

1 Introduction

CO_2 emissions in big cities due to transportation systems raise the need to improve the sustainability and accessibility of collective transport, while simultaneously promoting the use of more environmentally friendly transportation systems. In this sense, it is important to make available adequate and updated information regarding the mobility options offered by transport operators and users. One important

J.C. Ferreira(✉) · V. Monteiro · J.L. Afonso
Centro ALGORITMI, University of Minho, Guimarães, Portugal
e-mail: jferreira@deetc.isel.ipl.pt, vmonteiro@dei.uminho.pt

J.A. Afonso
CMEMS-UMinho, Guimarães, Portugal
e-mail: jose.afonso@dei.uminho.pt

J.C. Ferreira
ADEETC at ISEL, Lisbon, Portugal

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_10

research work is to create a tool to measure the mobility of people in a city, identifying passively the transportation mode, routes performed and associated times. Information and communication technologies (ICT) have the potential to effectively change the way people live and their mobility and energy consumption. Nowadays, mobile devices incorporates many diverse and powerful sensors, like GPS, cameras, microphones, light, temperature, direction (i.e., magnetic compasses) and acceleration (i.e., accelerometers). Accelerometers with GPS can be used to perform activity recognition [1], a task which involves identifying the physical activity of a user. Activity inference provides the 'what' of a user's context, whereas location sensors (such as cell-tower/Wi-Fi localization and/or GPS) provide the 'where'. This 'what' and 'where' information can be used by a number of mobile phone applications, including physical fitness and health monitoring [2], recommendation systems, and the study of environment and personal behavior. In this work, we explore the potential of a mobile device application designed to track users' habits in a customized way, using integrated accelerometers and GPS information, with the goal to create a monthly user invoice related with their movements and used transportation modes (e.g., bus, train, car, bike, walk, etc.). This approach allows the discovery of mobility habits of millions of users passively, just by having them carry mobile phones in their pockets. From the collected sensor data, it is possible to identify the user's transportation mode and also characterize the traffic conditions. In association with the travelled distance, we can track the CO_2 emissions resulting from the user mobility and provide the information to the user in the form of a monthly invoice related with the concept of carbon footprint [3].

The rest of the paper is organized as follows. Section 2 describes the tasks associated to data acquisition and classification. Section 3 describes the developed transportation mode identification system and the corresponding performance evaluation results. Section 4 concerns to the user mobility patterns, while Section 5 presents the proposed mobility invoice system. Section 6 presents the mobility advisor to reduce the carbon footprint. Finally, Section 5 presents the conclusions and future work.

2 Data Acquisition and Classification

The applied methodology is composed by a set of sequential processes. The first process, identified as data collection, varies from case to case, being responsible for the collection of huge amounts of data (big data). For testing purposes, we used mobile device sensor data from 50 Lisbon area users, in a period corresponding to the first six months of 2013. This phase involves the identification of outliers and the removal of inconsistent data to reduce the number of records [4]. The second process consists on data transformation into predefined classes (Fig. 1). This is a process that is specific to each case study. Taking into account the mobile device sensor data, Fig. 1 shows the 20 predefined classes (C_1 to C_{20}). These classes are created based on accelerometer measurements in the three orientation axis, as well as additional GPS data. The accelerometer data is divided into three dimensions: Z (vertical) is the upright direction; Y (longitudinal) is the direction of movement and X (transversal) is the steering direction. The transformation process to merge original data into these predefined classes uses information of two consecutive accelerometer data measurements, where the data value difference is classified into four scales: 1 - straight, when the measurement changes in the module are less than 0.1 ms^{-2}; 2 - smooth, when it is more than 0.1 ms^{-2} and less than 0.5 ms^{-2}; 3 - rough, when it is between 0.5 ms^{-2} and 2 ms^{-2}; 4 - very rough, when it is more than 2 ms^{-2}.

3 Transportation Mode Identification

From the mobile device sensor data, it is possible to identify the transportation mode that the user takes to go from A to B. Transportation mode detection has been explored by [5-7], among others. All of these approaches use past data to build a classification model that identifies the transportation mode and most of these approaches use a combination of GPS and accelerometer data from the three axes. From this data, it is possible to calculate the speed and the position. We developed this work based on a discrete approach using predefined data classes and a training set of 250 cases, representing car travel (60), bus (50), train (30), underground (40), walking (25), boat (20) and motorcycle (25). Table 1 shows an

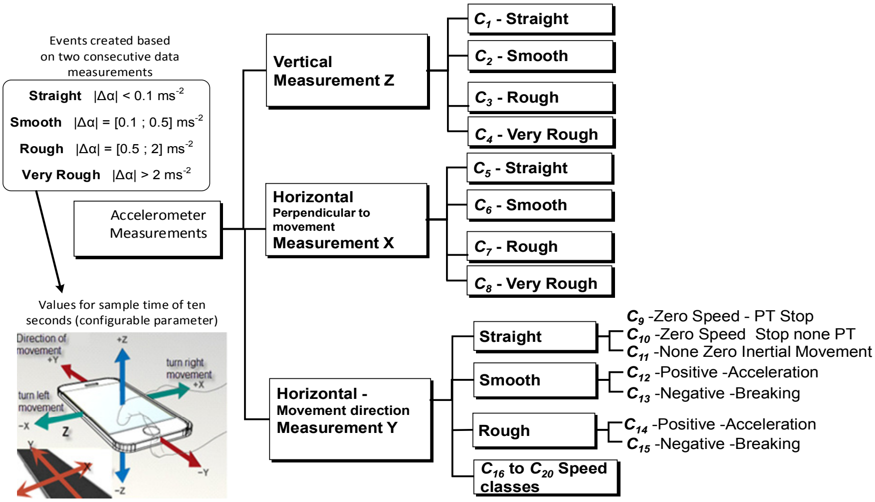

Events created based on two consecutive data measurements
Straight |Δα| < 0.1 ms-2
Smooth |Δα| = [0.1 ; 0.5] ms-2
Rough |Δα| = [0.5 ; 2] ms-2
Very Rough |Δα| > 2 ms-2
Accelerometer Measurements
Values for sample time of ten seconds (configurable parameter)
Direction of movement +Y
+Z
turn right movement +X
turn left movement -X
Z
Y
X
-Z
-Y
Vertical Measurement Z
C1 - Straight
C2 - Smooth
C3 - Rough
C4 - Very Rough
Horizontal Perpendicular to movement Measurement X
C5 - Straight
C6 - Smooth
C7 - Rough
C8 - Very Rough
Horizontal - Movement direction Measurement Y
Straight
C9 -Zero Speed - PT Stop
C10 -Zero Speed Stop none PT
C11 -None Zero Inertial Movement
Smooth
C12 -Positive -Acceleration
C13 -Negative -Breaking
Rough
C14 -Positive -Acceleration
C15 -Negative -Breaking
C16 to C20 Speed classes

Fig. 1 Predefined data classes.

Table 1 Sample of data training set with three transportation modes. All values represent number of events after the second process.

Data Class	Car (12 km)	Bus (7.5 km)	Metro (2.3 km)
C_1	23	314	2313
C_2	2111	1350	112
C_3	3312	4175	2
C_4	34	65	0
C_5	3402	3603	967
C_6	1501	2421	1223
C_7	425	165	121
C_8	126	6	0
C_9	9	454	367
C_{10}	390	420	8
C_{11}	1234	1242	634
C_{12}	1476	1580	1021
C_{13}	1021	1374	32
C_{14}	521	412	53
C_{15}	631	556	41
C_{16}	254	2860	430
C_{17}	532	642	267
C_{18}	264	1210	412
C_{19}	1375	1340	810
C_{20}	96	0	0

example of event count per class for three transportation modes (other modes are omitted for simplicity purposes). Since we want a generic approach, the main effort to perform is the transformation of raw data into these predefined classes, as Table 1 shows. These classes can be increased to cover new situations, and when not used should be treated as empty fields.

Our approach for public transportation mode identification is based on the value of $P(TM_j|C_k)$, which means the probability of transportation mode j (TM_j) given C_k measurements are discrete classes defined from accelerometer data as illustrated on Figure 1. We use classes C_1 to C_{20}, using the Bayes theorem:

$$P(TM_j|C_k) = \frac{P(C_k|TM_j)}{P(C_k)} \propto P(C_k|TM_j)P(TM_j) \qquad (1)$$

where, TM_1 = Car, TM_2 = Bus, TM_3 = Train, TM_4 = Metro, TM_5= Walk, TM_6 = Boat and TM_7 = Motorcycle. We calculate $P(TMj)$ as the number of cases for transportation mode j divided by the number of cases in the training set, for example, $P(TM_1) = 60/250 = 0.24$. The same approach is used to calculate the other values. For the probability $P(C_k|TMj) = P(\{C_1,C_2,..,C_{20}\}|TMj)$, we assume the independence of events, therefore:

$$P(C_k|TM_j) = \prod_{k=1}^{20} P(C_k|TM_j)\, P(TM_j) \qquad (2)$$

where, $P(C_k|TMi)$ is based on the training data set. Since the number of events varies with the sampling time and the route distance, we perform a normalization using the percentage. For example, for X-axis accelerometer data from car samples, we have 23 C_1 events, 2111 C_2 events, 3312 C_3 events and 34 C_4 events. This totals 5480 events, so we have 0.4% of events in class C_1, 38.5% in class C_2, 60.5% in class C_3 and 0.6% in class C_4. To avoid zero probability, since (2) is a multiplication, we always introduce an offset of one in the counting process. This is a similar process used in text classification through NB [8]. Taking into consideration that the 60 cases correspond to trips performed by car, in the calculation of the average of these values we reached the value of 0.5% for $P(C_1|TM_1)$. From other values of the training set we have $P(C_1|TM_2) = 10\%$ (this is higher because of the waiting time at bus stations, where the user is immobile, which means that C_1 events are being collected), $P(C_1|TM_3) = 95\%$, $P(C_1|TM_4) = 91\%$, $P(C_1|TM_5) = 15\%$, $P(C_1|TM_6) = 5\%$, and $P(C_1|TM_7) = 2\%$. This collection process with the class identification allows the determination of the transportation mode because some events are characteristic of some transportation modes in particular. For example, events in the C_1 class are common in railroad or underground transportation, which implies very smooth changes in Z-axis acceleration. On the other hand, if we have the majority of events in class C_4, this indicates a motorcycle or a boat. Meanwhile, the study of the acceleration shape allows differentiating the boat from the motorcycle. C_3 means several speed limitation bumps or potholes in the road in a vehicle or bus. Again, through the pattern of acceleration, it is possible to distinguish a pothole from a speed bump. Acceleration data from the X-axis (C_8 and C_7 classes) can be helpful to identify aggressive or drunk drivers.

Speed information from GPS is used to differentiate among walking, bicycling, boat, and other transportation modes. Periodic stops are used to differentiate among car/motorcycle and underground, bus or train. Motorcycle is better discerned from car transportation in high traffic periods, because the average speed is higher and the position pattern is different. In order to distinguish metro from train, we use the following heuristics: (1) Underground usually runs below ground without a GPS signal; (2) Distances between stops in underground transportation are usually smaller; (3) Altimetry information. Given the sensor information and GPS traces, we predict the transportation mode among the available modes. This is done calculating the probability $\underline{P(TM_j|D_k)}$, with ($j = 1,\ldots 7$) and ($k = 1,\ldots 20$), and choosing the highest value. Table 2 presents performance results in a confusion matrix. Results are available using 6 month data in more than 2500 recorded mobility experiences. From this data we used 250 for training purposes and evaluated 500 cases based on precision measurement (number of correct cases classified for that transportation mode divided by the transportation mode cases available). We achieved high precision identification values for walking (97%) and car (95%), having lower values for train (70%) and underground (72%). In 24% of cases there was a classification change from underground to train.

Table 2 Confusion matrix for the transportation mode identification.

	TM_1	TM_2	TM_3	TM_4	TM_5	TM_6	TM_7
TM_1	95%	3%	1%	0%	0%	0%	1%
TM_2	3%	89%	3%	2%	1%	0%	2%
TM_3	1%	3%	70%	24%	1%	0%	1%
TM_4	0%	2%	24%	72%	1%	1%	0%
TM_5	0%	1%	1%	1%	97%	0%	0%
TM_6	0%	0%	0%	1%	0%	82%	17%
TM_7	1%	2%	1%	0%	0%	17%	79%

4 User Mobility Patterns

Other knowledge that can be extracted from the collected mobile device sensor data is related with the information of the distinct locations where the users spend their time throughout the day (e.g., home, work, shopping centers, restaurants, etc.). From the data that we have collected, we are particularly interested in identifying locations where people spend a great deal of time, and associating these locations with in-formation about the environment, obtained from geographic information system data sources. We have all GPS data and time stored in a user mobility profile, in a cloud database, with the information about time and routes (XML graph with time and GPS coordinates). It is possible to present the route representation for that month with associated information of the transportation mode, the number of times the route was performed, and also the temporal periods. Thus, it is possible to represent the time that a user spent in each location. User mobility profile is stored in central server. Based on the information shown in Fig. 2, it is possible to produce a monthly invoice.

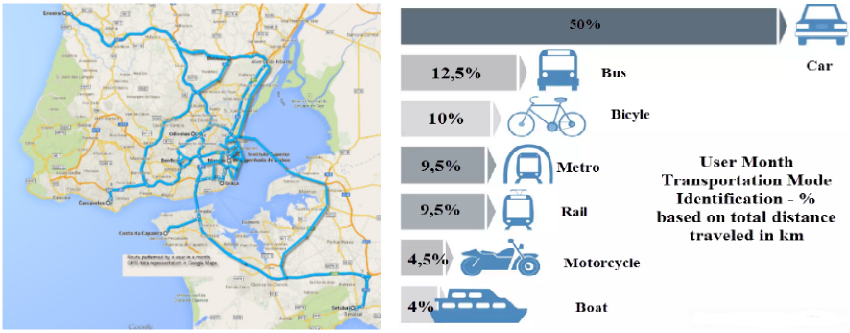
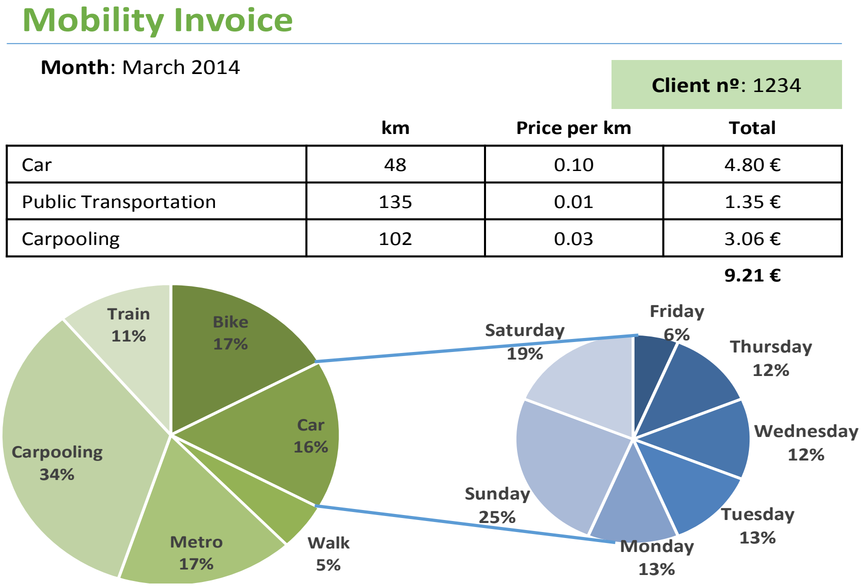


Fig. 2 User movement in a monthly period and associated transportation mode.



Mobility Invoice

Month: March 2014

Client nº: 1234

	km	Price per km	Total
Car	48	0.10	4.80 €
Public Transportation	135	0.01	1.35 €
Carpooling	102	0.03	3.06 €
			9.21 €

Fig. 3 Example of a mobility invoice, based on transportation mode identification and associated distance.

5 Mobility Invoice System

Presently, carbon footprint is the most popular measure of environmental impact, and it is used to refer the amount of greenhouse gas emissions that are produced during the mobility process of a user. Using the knowledge discovery of transportation mode detection and associated distances, we are able to generate a monthly report with the associated CO_2 emissions (see Fig. 3), where a price per km is defined for each transportation mode [9][10]. From the tracking of user movements, it is possible to improve public transportation routes and timing. Our intention is to show an important output, which is a result of the passive data collection from mobile devices, and the consequent application of knowledge discovery to sustainable transportation. It is possible to develop a suggestion system to reduce this invoice and, consequently, reduce the CO_2 emissions in the city. This reduction can be achieved based on: (1) Carpooling system to reduce car usage; (2) Public transportation information integration [11]. For this purpose, a mobile application, developed based on current user position and taking into account the user habits (past route performed), provides personalized advice to go from current position to final destination based on public transportation availability (user receives route and scheduling information). An example of a mobility invoice is showed in Fig. 3, where users travel distance by transportation mode are used to get a value, related with CO_2 emissions price based on values identified in Fig 4.

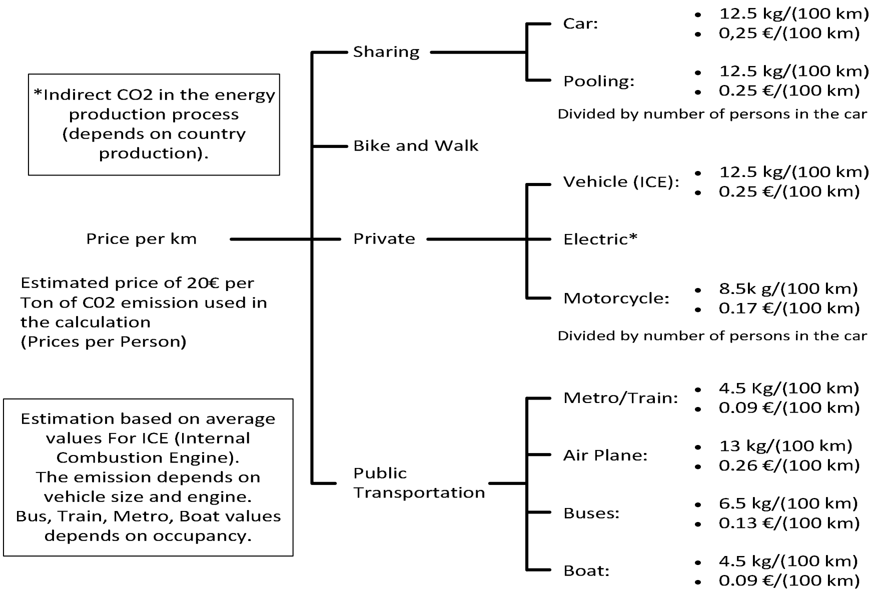

*Indirect CO2 in the energy production process (depends on country production).
Price per km
Estimated price of 20€ per Ton of C02 emission used in the calculation (Prices per Person)
Estimation based on average values For ICE (Internal Combustion Engine). The emission depends on vehicle size and engine. Bus, Train, Metro, Boat values depends on occupancy.
Sharing
Bike and Walk
Private
Public Transportation
Car:
• 12.5 kg/(100 km)
• 0,25 €/(100 km)
Pooling:
• 12.5 kg/(100 km)
• 0.25 €/(100 km)
Divided by number of persons in the car
Vehicle (ICE):
• 12.5 kg/(100 km)
• 0.25 €/(100 km)
Electric*
Motorcycle:
• 8.5k g/(100 km)
• 0.17 €/(100 km)
Divided by number of persons in the car
Metro/Train:
• 4.5 Kg/(100 km)
• 0.09 €/(100 km)
Air Plane:
• 13 kg/(100 km)
• 0.26 €/(100 km)
Buses:
• 6.5 kg/(100 km)
• 0.13 €/(100 km)
Boat:
• 4.5 kg/(100 km)
• 0.09 €/(100 km)

Fig. 4 Identification of the price per km of CO_2 emissions based on transportation mode.

6 Mobility Advisor to Reduce the Carbon Footprint

The main idea is to provide advice to reduce the carbon footprint based on historical user mobility data, through the matching of this data with public transport services using the information of available routes and schedules. This approach can be complemented with data integration of different information sources, such as: car and bike sharing systems [11][12] and also carpooling [13]. These systems are important to the growing interest of the sustainable transport systems, and to reduce the growth of energy use, noise, air pollution and traffic congestion, due to a decrease in the number of cars and to an increase in the shared use of electric or low pollutant vehicles. Also, bike sharing systems permit door-to-door ride features and allow to access areas of the city that are forbidden for other kind of vehicles. Also, the integration of public transportation data can be used as complement. Through previous work, we integrate the access to several public transportation databases using a semantic approach described in [13] and the START project [http://www.start-project.eu], where the integrated information available in mobile devices increases the usage of public transportation among different experiences in European project partners (England, France, Spain and Portugal). With this, it is possible to have information about the schedule and the price of public transportation. With this information in a previous project we build a route planner that integrates routes and public transportation [11].

7 Conclusions

Tracking of user activity and associated mobility is now possible at low costs through the use of mobile device sensor information. The data generated using this approach is in the class of big data and has great impact in the study of user mobility habits. . In this work we show the application of this approach to the tracking of user mobility in a city through the identification of the used transportation modes, routes and times. This information can be transformed in an informative CO_2 invoice, which allows users to be aware of their carbon footprint. Along with that, associated measures and suggestions to reduce this invoice are then provided. With a considerable number of users, this passive tracking of data from citizens could generate useful data about mobility habits and be used to improve the citizens' mobility. Moreover, public transport operators can use the processed data to improve their transportation offers towards the users' needs.

References

1. Khan, W.Z., Xiang, Y., Aalsalem, M.Y., Arshad, Q.: Mobile phone sensing systems: A survey. Comm. Surveys & Tutorials **15**, 402–427 (2013)
2. Pantelopoulos, A., Bourbakis, N.G.: A survey on wearable sensor-based systems for health monitoring and prognosis. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **40**(1), 1–12 (2010)
3. Luckow, P., Stanton, E.A., Biewald, B., Fisher, J., Ackerman, F., Hausman, E.: 2013 Carbon Dioxide Price Forecast. Synapse – Energy Economics, Inc., November 2013
4. Ferreira, J., Almeida, J., Silva, A.: The Impact of Driving Styles on Fuel Consumption: A Data Warehouse and Data Mining based Discovery Process. IEEE Transactions on Intelligent Transportation Systems **16**(5), 2653–2662 (2015)
5. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Using Mobile Phones to Determine Transportation Modes. ACM Trans. on Sensor Networks **6**(2), 13–40 (2010)
6. Patterson, D.J., Liao, L., Fox, D., Kautz, H.: Inferring high-level behavior from low-level sensors. In: UbiComp 2003: Ubiquitous Computing, vol. 2864, pp. 73–89, October 2003
7. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y.: Understanding mobility based on GPS data. In: International Conf. on Ubiquitous Computing, pp. 312–321, September 2008
8. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: Workshop on Learning for Text Categorization, vol. 752, pp. 41–48. AAAI Press (1998)
9. Ackerman, F., Stanton, E.A.: Climate Risks and Carbon Prices: Revising the Social Cost of Carbon. Economics: The Open-Access, Open-Assessment E-Journal **6**, 1–25 (2012)
10. Ferreira, J.C., Silva, A.R., Afonso, J.L.: EV-Cockpit – mobile personal travel assistance for electric vehicles. In: Microsystems for Automotive Applications. Smart Systems for Electric, Safe and Networked Mobility. Springer (2011)

11. Ferreira, J.C., Monteiro, V., Afonso, J.A., Afonso, J.L.: Mobile Cockpit System for Enhanced Electric Bicycle Use. IEEE TII **11**(5), 1017–1027 (2015)
12. Ferreira, J.C., Trigo, P., Filipe, P.: Collaborative car pooling system. In: International Conf. on Sustainable Urban Transport and Environment, Paris, pp. 1–5, June 2009
13. Ferreira, J.C., Filipe, P., Martins, P.M.: Cooperative transportation infrastructure. In: ICITIS Int. Conf. on Civil, Transport and Env. Eng., Malaysia, pp. 1–5, August 2013

Methodology for Knowledge Extraction from Mobility Big Data

João C. Ferreira, Vítor Monteiro, José A. Afonso and João L. Afonso

Abstract The spread of mobile devices with several sensors, together with mobile communication, provides huge volumes of real-time data (big data) about users' mobility habits, which should be correctly analysed to extract useful knowledge. In our research we explore a data mining approach based on a Naïve Bayes (NB) classifier applied to different sources of big data. To achieve this goal, we propose a methodology based on four processes that collects data and merges different data sources into pre-defined data classes. We can apply this methodology to different big data sources and extract a diversity of knowledge that can be applied to the development of dedicated applications and decision processes in the area of intelligent transportation systems, such as route advice, CO_2 emissions reduction through fuel savings, and provision of smart advice for public transportation usage.

Keywords Big data · Data mining · Naïve bayes · Mobile device · Sensor information

1 Introduction

Knowledge discovery (KD) from the big data available in databases has been explored with success in several areas, and usually is better known through the more popular term "data mining" [1]. Data mining (DM) allows the search for

J.C. Ferreira(✉) · V. Monteiro · J.L. Afonso
Centro ALGORITMI, University of Minho, Guimarães, Portugal
e-mail: jferreira@deetc.isel.ipl.pt, vmonteiro@dei.uminho.pt

J.A. Afonso
CMEMS-UMinho, Guimarães, Portugal
e-mail: jose.afonso@dei.uminho.pt

J.C. Ferreira
ADEETC at ISEL, Lisbon, Portugal

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_11

knowledge in large volumes of data [2] and it has been applied to different areas to discover hidden profitable patterns in databases (business opportunities). It is also applied in major research areas, as the example of application in bioinformatics, in genomics and proteomics identification [3].

Some works on this novel research area include traffic analysis and prediction [4], road accidents analysis [4], driver behaviour prediction [6], and range prediction for electric vehicles [7]. This paper is oriented to establish a working methodology that can reuse processes and algorithms in many cases of KD from the big data available from tracking users' mobility activity. This goal results from our experience in several mobility projects, where the DM processes can be reused, avoiding the costs associated to the development of solutions to new problems from scratch. The intention is to create a tool tailored to the increasing data available in the mobility area, which can be used with little effort in different application cases.

This information can be easily collected from mobile devices or based on commercial products. Our work on mobile devices uses GPS and accelerometer sensor data to passively track the users' mobility activity, as described in Section 2. This is a personalized data tracking process from where we can extract useful knowledge related with user mobility habits, like the transportation mode (bus, train, car, bike, walking or other) and carrier number associated, identify main mobility patterns, and generate a mobility invoice. Passive tracking of user activity using mobile devices [8] has been used in a diversity of studies applied to activity recognition [9] and transportation mode detection [10], among others [11][12]. Through the proposed approach, our goal is to establish a common process to handle the collected data into a diversity of KD, in order to facilitate the development of dedicated applications for intelligent transportation systems (ITS).

This approach also allows the joining and aligning of the work of field experts on generating the mobility big data with mining experts, in order to better extract knowledge from the collected data. This can be achieved by data discretization in predefined classes, performed by these field experts. This approach can also be applied for non-professional cases, such as personal data tracking, allowing its extension to the mobility habits of millions of users just by carrying cell phones in their pockets. User data privacy from this tracking activity is an important concern that increases in the context of big data. Thus, all personalized information concerning user mobility is stored in a central database with security parameters (login, encryption). Although we have individual data regarding users' movements, the only personalized information stored is the user email to exchange information.

The main contribution this paper is a common approach of Naïve Bayes (NB) application for KD, based on the normalization of different collected mobility data into predefined classes and sub-classes, which can be used for a diversity of mobility applications developed in a mashup approach. This approach makes possible an increase in the number of available, dedicated or personalized ITS applications, because the development is facilitated by the use of common parts and processes.

The next section presents the proposed work methodology. Section 3 presents examples of application development based on KD from mobile device data. Finally, in Section 4 we highlight the main conclusions.

2 Work Methodology

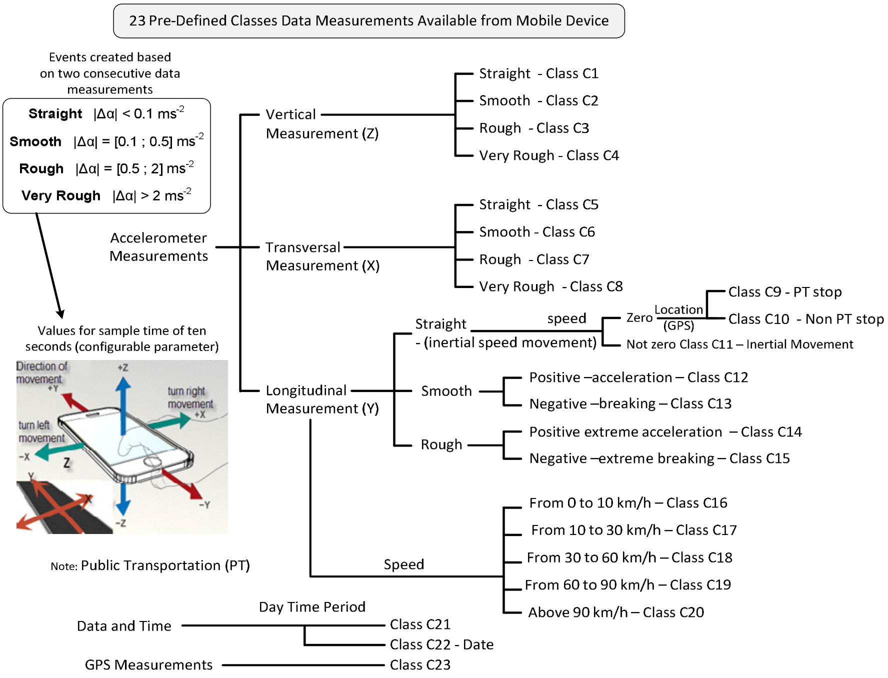
To extract knowledge we need a proper methodology to work with the big data generated by different sources. Due to analytical needs and the huge amount of data generated, this is a field in which it is possible to apply KD [13] based on a DM approach. The work methodology proposed in this paper involves four main processes:

1 – Data collection process associated with data cleaning (outliers identification and removal), using a common approach based on a dedicated data transformation;

2 – Data discretization process into predefined classes with the goal of data uniformization among different data collection conditions;

3 – Data discretization into sub-classes;

4 – KD based on a DM process.



23 Pre-Defined Classes Data Measurements Available from Mobile Device
Events created based on two consecutive data measurements
Straight |Δα| < 0.1 ms-2
Smooth |Δα| = [0.1 ; 0.5] ms-2
Rough |Δα| = [0.5 ; 2] ms-2
Very Rough |Δα| > 2 ms-2
Accelerometer Measurements
Vertical Measurement (Z)
Straight - Class C1
Smooth - Class C2
Rough - Class C3
Very Rough - Class C4
Transversal Measurement (X)
Straight - Class C5
Smooth - Class C6
Rough - Class C7
Very Rough - Class C8
Values for sample time of ten seconds (configurable parameter)
Direction of movement
+Y
+Z
turn right movement
+X
turn left movement
-X
Z
-Y
-Z
Longitudinal Measurement (Y)
Straight - (inertial speed movement)
speed
Zero
Location (GPS)
Class C9 - PT stop
Class C10 - Non PT stop
Not zero Class C11 – Inertial Movement
Smooth
Positive –acceleration – Class C12
Negative –breaking – Class C13
Rough
Positive extreme acceleration – Class C14
Negative –extreme breaking – Class C15
Speed
From 0 to 10 km/h – Class C16
From 10 to 30 km/h – Class C17
From 30 to 60 km/h – Class C18
From 60 to 90 km/h – Class C19
Above 90 km/h – Class C20
Note: Public Transportation (PT)
Data and Time
Day Time Period
Class C21
Class C22 - Date
GPS Measurements
Class C23

Fig. 1 Information about the creation of predefined classes and sub-classes based on a discretization process, with details for mobile device sensor data events creation based on 23 predefined data classes.

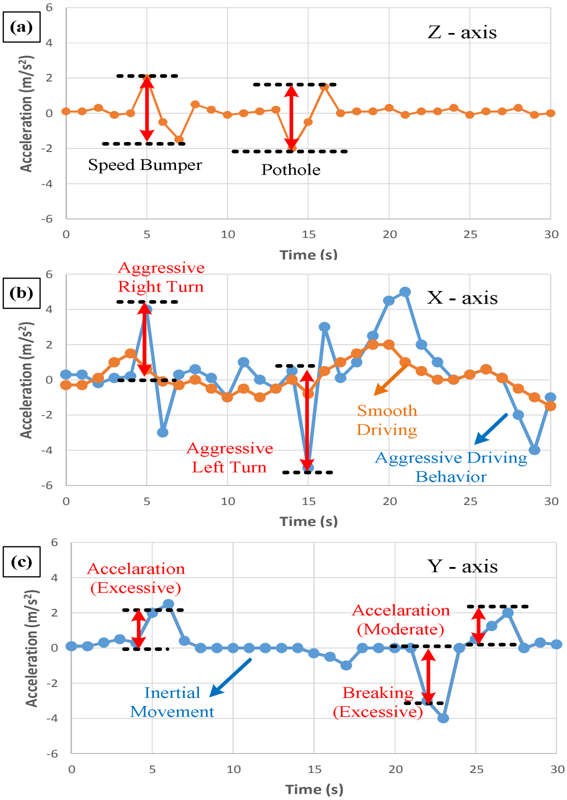


Fig. 2 Accelerometer data transformation process from collected data to predefined classes: (a) *C1* to *C4*, based on X-axis accelerometer measurements; (b) *C5* to *C8*, based on X-axis accelerometer measurements (data from an aggressive driver and a smooth driver); (c) *C9* to *C15*, based on Y-axis accelerometer measurements.

The first process, which varies from case to case, is responsible for the collection of huge amounts of data (big data). To show the application of the proposed methodology we use data from mobile device sensors. For testing purposes, we use data from 50 Lisbon area users, in a period corresponding to the first six months of 2013. The first phase involves the identification of outliers to reduce the number of records. Inconsistent data is also removed during this phase [14], where we remove distant data points based on statistical measurements. Process 2 consists on data transformation into predefined classes. This is a process that is specific to each case study. Taking into account the mobile device sensor data, Fig. 1 shows the 23 predefined classes (C1 to C23). These classes are created based on accelerometer measurements in the three orientation axis, as well as additional GPS data. The accelerometers' data is divided into three dimensions: Z (vertical) is the upright direction; Y (longitudinal) is the direction of movement and X (transversal) is the horizontal direction. The transformation process to merge original data into these predefined classes uses information of two consecutive accelerometer data measurements, where the data value difference is classified into four scales (as shown in the left box of Fig. 1). This number of scales (four) is a compromise between diversity and complexity, as selected based on an analysis of several tested cases (ranging from two to nine scales). Figure 2 shows the creation of data events (measurements performed by the tracking device or application in a sampling period). These data events can be allocated to a specific route or not. All of these data events are allocated into predefined classes. Classes *C1* to *C4* are based on accelerometer data on the Z-axis. This data can be used to identify, together with the GPS coordinates, the position of potholes and speed bumps. Classes *C5* to *C8* are based on accelerometer data on the X-axis. Fig. 2 (b) shows data from an aggressive driver, with aggressive left and right turns. For the Y direction (Fig. 2 (c)), we divide these data events into thirteen classes: *C9* when the speed is close to zero and the GPS coordinates match a public transportation stop; *C10* when the speed is close to zero and the GPS coordinates do not match a public transportation stop; *C11* when the speed is different from zero, which means inertial movement. The other events are used to identify moderate or aggressive acceleration and braking events. Speed values are taken from:

$$V_y[k+1] = V_y[k] + ta_y[k]\,, \tag{1}$$

where, t is the elapsed time between samples, k is the sampling time and a_y is the measurement of acceleration in the Y-axis. The speed data is transformed into classes *C9*, *C10*, *C11* and *C16* to *C20*.

GPS data is also transformed in this process to route graph data, due to our interest in the route trajectories. This process eliminates the error associated to the GPS measurements and allows the saving of storage space. Each road is represented by its corners, and we match the GPS coordinates against the nearest road corner. As shown in Fig. 3, for the first GPS data collected we need to look for the nearest road corner located inside a circle centered on the current GPS position. We calculate the distances to these six corners (1 to 6) and choose the smallest one, which in this case is the corner number 1. This process is repeated several times, forming a route. After that, we use this route in a graph to perform the matching against other routes. The developed system adds more GPS nodes than those shown in Fig. 3, because each road corner generates a node, but for simplification purposes we show only nodes from the main roads. With this process, we can store route trajectories in a graph for future processing, where a circular distance function with a predefined radius is used from the current user position to route the graph trajectory.

Process 3 consists in the class discretization into sub-classes. This step can be applied to continuous or discrete data with the aim of dividing the collected data into sub-classes. Due to the diversity of application cases we present multiple options to the user so he may decide the best one to apply. This discretization process can be based on Heuristics, applied based on [15]: (1) Predefined criteria; (2) Equal area clustering method; (3) Data population division based on percentage.

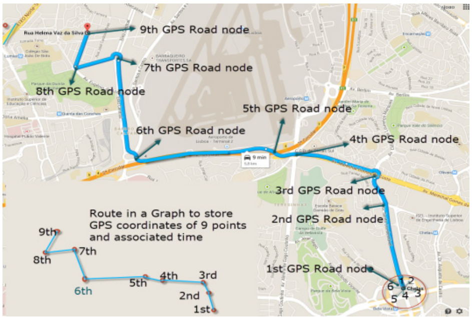


Fig. 3 Simplified process to generate route paths in geo-referenced graphs.

Process 4 is oriented to KD based on the predefined sub-classes, to use a common approach developed as a tool for easy usage. Since the classes and related sub-classes are similar from case to case, the same approach is applied to different cases of knowledge extraction. We implement a NB algorithm that we use to extract different knowledge based on big data allocated in our predefined class. In both cases, the same NB approach is applied using the predefined class and sub-classes. This tool was developed on top of Microsoft SQL Server 2008R2 with SSIS (SQL Server Integration Services), SSAS (Microsoft SQL Server Analysis Services) and SEMMA (Sample, Explore, Modify, Model and Assess), but others platforms are possible, such as the following open source platforms: Rapid Miner [16], Weka [17] or R [18].

3 Application Based on KD from Mobile Device Data

Mobile devices are a rich source of mobility data because users carry them all the time. In this section we present the developed applications based on this data acquisition process and the proposed KD methodology. Technical development details are omitted because we are interested in the idea expressed by the proposed methodology, the usage of a common approach and the reuse of processes:

1) **User Mobility Patterns:** Other knowledge can be extracted from this mobile device sensor data, like the information of where persons spend their time at several distinct locations throughout the day: home, work, shopping centers, restaurants, etc. From the data that we have collected, we are particularly interested in identifying locations where people spend a great deal of time, and associate these locations with information about the environment obtained from geographic information system data sources. All GPS data is stored in a user mobility profile, in a cloud database, with the information about time and routes (XML graph with time and GPS coordinates). It is possible to present the route representation for that month with associated information of the transportation mode, the number of times the route was performed, and also the temporal periods. Thus, it is possible to represent the time that a user spent

in a month in these locations. This information represents the user mobility activity captured by the mobile device sensors.

2) **Traffic Information - My Traffic Info App**: Based on the speed and location information (GPS data), it is possible to identify traffic situations as conditions that occur on road networks as the number of vehicles increase and are characterized by slower speeds and longer trip times. From the information of a current road (matching of position against available routes), it is possible to check traffic situations using the relation of current speed divided by the maximum road speed (we call this ratio *Vt*). The system tries to classify this traffic into four classes: (1) Green if the user's average *Vt* is above 0.25; (2) Yellow for average *Vt* between 0.1 and 0.25; (3) Red when the average *Vt* is below 0.1; (4) Black when average *Vt* is zero. We disregard $Vt = 0$ at public transportation stops if transportation mode is a public transportation, as well as at road intersections, because we assume (for simplification purposes) the existence of a traffic light on each road intersection. To avoid this we would need the GPS coordinates of the traffic lights, which are taken from road graph information. There may be times where the number of online users is small, which results in a reduction of information in the user database. Thus, it is necessary to use external information from traffic web services. The current average speed information of the users in a given route is used as input in the route advisor. Taking into account the traffic KD and since we have all users' positions in a graph, it is possible, based on a Dijkstra's algorithm [19], to propose alternative routes in a real-time approach, where we minimize the time between all possible graph nodes to reach the final destination. Since we stored past user routes based on the current user position, it is possible to generate personalized traffic alerts when traffic happened on users' past taken routes. We illustrate the app usage with a small example: The user A in the morning period always follows a certain route to go to work. As he starts the driving process, the system knows his position by GPS information, and a traffic situation is detected three blocks away in his usual route. The app alerts the user on his mobile device, and the application shows an alternative route. We are aware of the danger of the use of a mobile device while driving, but this problem can be overcome with upcoming device interoperability standards that offer integration between a Smartphone and a car's infotainment system, such as MirrorLink [20]. The result is My Traffic Info, is an Android App that uses traffic KD and the information about a past route driven, in order to present traffic alerts to the driver and, based on a developed real-time route planner [19], propose an alternative route.

3) **Road Classification Application and Potholes Identification:** Accelerometer vertical measurement (Z-axis) can give out important information about road conditions. It is possible to identify potholes and, with GPS coordinates, mark them in a road map. This application ensures road surface quality assessment in a continuous monitoring process, and can be important information for road authorities and drivers. Road quality can be used as an input parameter of router planning. Basically, this detection process uses Z data from accelerometer (see Fig 2 (a)) and looks for an initial decrease of this Z-acceleration of more than 1.5 ms-2 (this value was selected based on the

study of several cases) followed by an increase. It is possible to classify the severity of a pothole based on this Z-axis acceleration value and duration. When these events are detected, the GPS coordinates are stored for geographic representation. It is possible to identify the road side associated to the pothole based on the movement direction, and the GPS measurement error is reduced because we use data from multiple users. This application classifies the roads based on predefined criteria, like the number of potholes, and these are divided into three severity scales (small, medium and big) using the Z-axis accelerometer difference between two consecutive measurements.

4 Conclusions

Tracking user activity and associated mobility is available at low costs through the possibility of using mobile device sensor information. The data generated is huge and has great impact in the study of user mobility habits. In this research we show different applications of this big data tracking using a common mining approach for knowledge discovery (KD), which can be used for specific applications, based on predefined classes and sub-classes. The proposed work methodology provides a bridge between field experts in data collection and data mining (DM). This framework needs to be improved with more case experiences, but it is one of the first steps towards the establishing of a semi-automatic approach to KD in big data and a mashup approach for intelligent transportation systems web applications. Another important issue is the connection among these applications and others, in order to share knowledge and data. User tracking data is important for public transportation planning, and even for advertisement applications, because the information can be automatically personalized based on location and time. Other DM algorithm approaches can be applied, but the Naïve Bayes (NB) classifier has a simplified common approach application. The proposed methodology can be applied to different big data sources. Another important issue is the diversity of mobility applications that can be easily developed using all this mobile device data. One example is a mobility invoice system, as a tool for citizens to be aware of their carbon footprint impacts, along with associated measures or suggestions to reduce this invoice. The passive tracking of data from citizens (with a considerable number of users) could generate useful data about mobility habits and be used to improve citizens' mobility.

References

1. Mariscal, G., Marbán, Ó., Fernández, C.: A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review **25**(2) (2010)
2. Weiss, S.M., Indurkhya, N.: Predictive Data Mining: A Practical Guide (The Morgan Kaufmann Series in Data Management Systems), 1st edn. MK Publ., San Francisco, August 1997

3. Li, J., Wong, L., Yang, Q.: DM in Bioinformatics, IEEE Intelligent System. IEEE CS (2005)
4. Baldi, M., Baralis, E., Risso, F.: Data mining techniques for effective and scalable traffic analysis. In: IEEE Int. Symposium on Integrated Network Management, pp. 105–118 (2005)
5. Fogue, M., Garrido, P., Martinez, F.J., Cano, J.-C., Calafate, C.T., Manzoni, P.: Using data mining and vehicular networks to estimate the severity of traffic accidents. In: Management Intelligent Systems, vol. 171, pp. 37–46. Springer, Heidelberg (2012)
6. Reiter, U.: Modeling the driving behaviour influenced by information technologies. In: Inter. Symposium on Highway Capacity. Highway Capacity and Level of Service, pp. 309–320 (1991)
7. Ferreira, J.C., Monteiro, V., Afonso, J.L.: Dynamic range prediction for an electric vehicle. In: EVS27 Int. Electric Vehicle Symposium & Exhibition, Barcelona, Spain (2013)
8. Khan, W.Z., Xiang, Y., Aalsalem, M.Y., Arshad, Q.: Mobile phone sensing systems: A survey. Communications Surveys & Tutorials **15**, 402–427 (2013)
9. Turaga, P., Pavan, R.C., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. IEEE Trans. on Circuits and Systems for Video Technology **18**(11), November 2008
10. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Using Mobile Phones to Determine Transportation Modes. ACM Transactions on Sensor Networks **6**(2), February 2010
11. Patterson, D.J., Liao, L., Fox, D., Kautz, H.: Inferring high-level behavior from low-level sensors. In: UbiComp 2003: Ubiquitous Computing, vol. 2864, pp. 73–89 (2003)
12. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y.: Understanding mobility based on GPS data. In: International Conf. on Ubiquitous Computing, pp. 312–321, September 2008
13. MacLennan, J., Tang, Z., Crivat, B.: Data mining with Microsoft SQL server 2008. Wiley Publishing Inc., Indianapolis (2009)
14. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining. AA for Artificial Intelligence, USA (1996)
15. de Almeida, J., Ferreira, J.C.: BUS public transportation system fuel efficiency patterns. In: Int. Conf. on Machine Learning and Computer Science, Kuala Lumpur, Malaysia, pp. 4–8 (2013)
16. Hofmann, M., Klinkenberg, R.: RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman & Hall/CRC DM & Knowledge Discovery Series). CRC Press (2013)
17. Waikato ML Group. User Manual Weka: The Waikato Environment for Knowledge Analysis. Department of Computer Science, University of Waikato (New Zealand), June 1997
18. Bunn, A., Korpela, M.: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org
19. Ferreira, J.C.: Green route planner. In: Nonlinear Maps and their Applications, Selected Contributions from the NOMA 2011 Int. Workshop, vol. 57, pp. 59–87. Springer (2013)
20. Huger, F.: User interface transfer for driver information systems: a survey and an improved approach. In: Int. Conf. on Automotive User Interfaces & Interactive Vehicular Applications (2011)

Smells Classification for Human Breath Using a Layered Neural Network

Sigeru Omatu and Mitsuaki Yano

Abstract Progress of sensor technology enables us to measure smells although it is based on chemical reactions. We have developed the smell classification for various subjects using layered neural networks by training a special smell. But we must learn many smells by repeating the same process and it is endless jobs since too many smells exist in the world. In case of a breath smell, several molecules are mixed. Therefore, if we can train basic components of the breath and mixtures are estimated by combining the basic components which consist the breath, it is preferable. In this paper, we develop mixed smell classification after training a neural network for each component by using a genetic algorithm to find a reduction factor from the measurement data which show the maximum value of the output of a layered neural network.

Keywords Smell measurement · Smell of breath · Layered neural network · Classification of breath

1 Introduction

Smell is one of five senses of human being and its study has been paid attention since it has been recognized for various applications in human life and industrial sectors. Based on the progress of medical research on the system of olphactory organs, artificial electronic nose (E-nose) systems have been developed. The various E-nose systems from technical and commercial viewpoints are developed in the field of quality control of food industry [1], public safety [2], and space applications [3].

In this paper we propose a new E-nose system which can separate two kinds of smells of mixed molecuels by using a layered neural network. Historically, J. Milke [4]

S. Omatu(✉) · M. Yano
Osaka Institute of Technology, Osaka 535-8585, Japan
e-mail: omtsgr@gmail.com, yano@elc.oit.ac.jp
https://www.oit.ac.jp

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_12

proved that two kinds of metal-oxide semiconductor gas sensor(MOGS) could have the ability to classify several sources of fire more precisely compared with a conventional smoke detector. However, his results achieved only 85% of correct classification by using a conventional statistical pattern classification. An E-nose has been developed for smell classification of various sources of fire such as household burning materials, cooking smells, the leakage from the liquid petroleum gas (LPG) in [5],[6] by using neural networks of layered type.

The purpose of this paper focuses on mixed smells classification. Various smells have been mixed naturally in our living environment. If a poisonous smell exists in our living environment, it must be recognized immediately by humans. Therefore, it is necessary to classify mixed smells into each component of mixed smells. However, sensors which classify smells of several types do not exist currently. The research purpose is to classify various mixed smells and construct the system that can be used repeatedly and inexpensively. We attempt to measure mixed smells using commercially available sensors and classify mixed smells of various density using a neural network.

2 Smell Sensors and Sensing System

Gas sensors using a tin oxide was produced in 1968 [5],[6]. There are tin, iron oxide, and tungsten oxide as typical gas sensing substances which are used in metal oxide semiconductor gas sensors (MOG sensor). In this case, an electrical resistance of the tin becomes low. Using the above oxidation and reduction process, we could measure whether a gas appears or not.

In this paper, MOG sensors are used to measure the single smell and mixed smell. Table 1 shows the five sensors in the experiment.

Table 1 Gas sensors used in the experiment made by FIS Corporation.

Sensor number (Type)	Gas detection types	Applications
Sensor 1 (SB-AQ8)	volatile, organic compound	air quality
Sensor 2 (SB-15)	propane, butane	flammable gas
Sensor 3 (SB-42A)	freon	refrigerant gas
Sensor 4 (SB-31)	alcol, organic solvent	solvent
Sensor 5 (SB-30)	algohol	alcohol detection

3 Neuron Model

We show the neural network fundamental structure on Fig. 1. The left hand side denotes inputs $(x_1, x_2, x_3, \ldots, x_n)$, the right hand side is the output y_j where

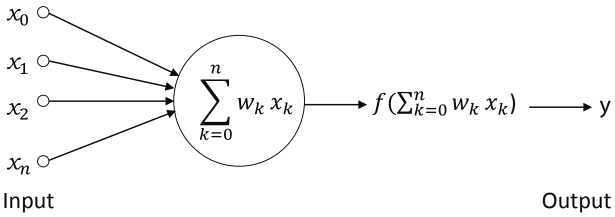


Fig. 1 Neuron model.

$x_i, i = 0, 1, \ldots, n$ denotes input data, $w_k, k = 0, 1, \ldots, n$ show weighting coefficients, $f(\cdot)$ is output function of the neuron, and y shows output. Note that $x_0 = -1$ and $w_0 = \theta$ where θshows a threshold of the neuron. Although there are several types of the output function $f(\cdot)$, we adopt the sigmoid function.

4 Error Back-Propagation Method

We use layered neural networks which are mainly used in pattern recognition or learning control. It has an input layer, a hidden layer, and an output laye as shown in Fig. 2.

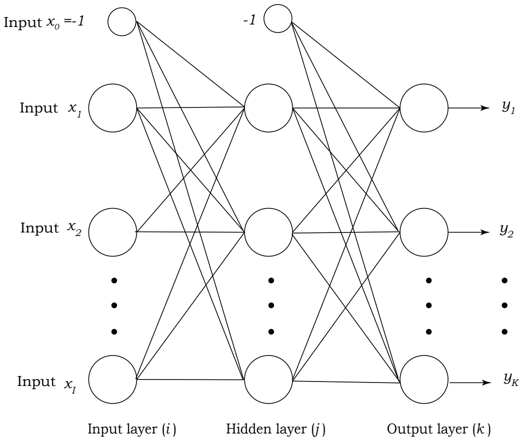


Fig. 2 Layered neural network.

5 Measurement of Smell Data

We have measured four types of smells as shown in Table 2. The sampling frequency is 500 [ms], that is, the sampling frequency is 0.5 [s]. The temperatures of smell gases were 18∼24 [°C], and the humidities of gases are 20∼30 [%]. We show sample paths of each smell of Table 1 in Figs. 3 and 4. Note that those factors in Table 2 are main smells included in our breath. Bad smell of our breath will make the neighboring persons unpleasant although the person himself might not notice it.

Furthermore, the bad smell of persons may reflect a kind of diseases and the measurement of human breath and smell classification will be profitable to predict a disease of persons in advance. Since these gasses exists very thin density of ppm or ppb order, we make those gasses of 1 ppm∼5 ppm order from 1% pure gasses. Using those gasses of densities, we have made mixing gasses such as 1:1, 1:5, and 5:1 mixing rates where a:b means a ppm and b ppm gasses are mixed. In case of four sources, there are six combination cases to mix two gasses. In this paper, we have mixed only two gasses of methyl-mercaptan (C) and hydrogen sulfide (D) since they are main components in our breath. Those sample paths are shown in Figs. 5–7 for different mixing rates of methyle-mercaptan (C): hydrogen sulfide (D) such as 1:1, 1:5, and 5:1, respectively.

Table 2 Training data.

Label	substance	samples
A	acetaldehyde	3
B	ethylene	3
C	methyl-mercaptan	3
D	hydrogen sulfide	3

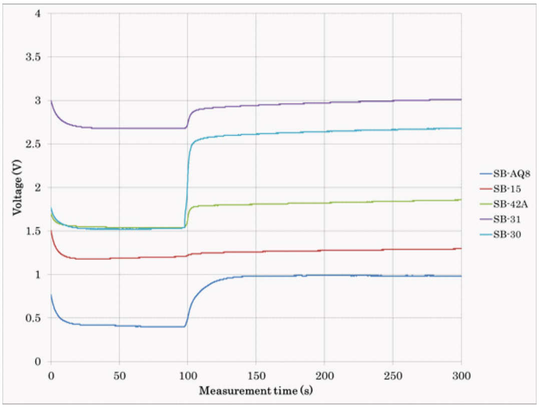


Fig. 3 Methyl-mercaptan.

6 Smell Classification of Breath Data

We must extract the features of the sample paths. Those sample paths of Figs. 4, 5, 6, 7 show time [s] to voltage [V] where at t=0, dry airs blow and at 100 seconds later gasses are injected. We can see that after about 60 seconds since dry airs blew in and after about 60 seconds later since smell gasses were injected at t=100 seconds later all sensors would become almost steady. Therefore, we decided the changes from t=60 to t=160 for each sensor reflect the features of the gasses and take the difference values at t=60 seconds and t=160 seconds for each sensor as features.

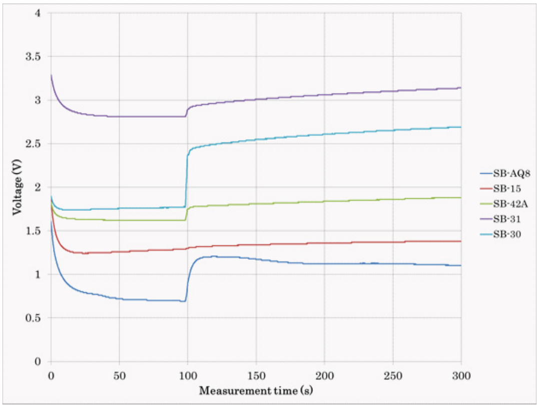


Fig. 4 Hydrogen sulfide.

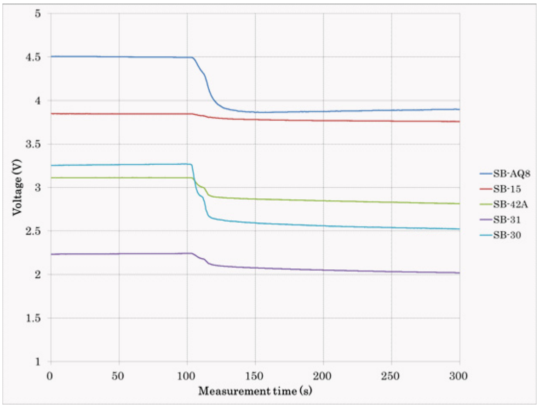


Fig. 5 Mixing rate 1:1.

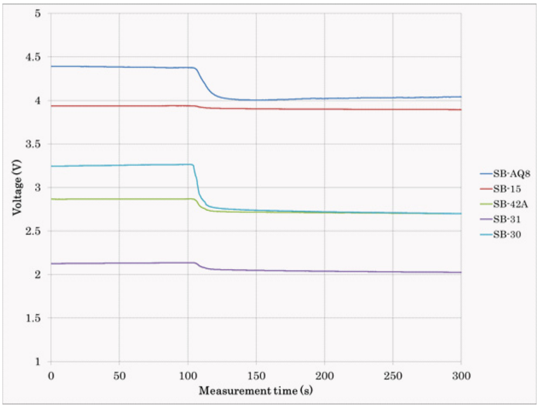


Fig. 6 Mixing rate 1:5.

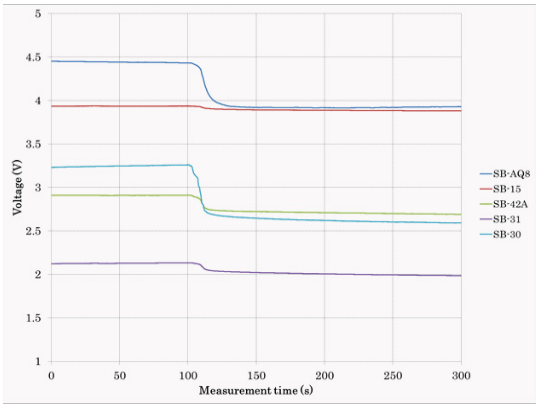


Fig. 7 Mixing rate 5:1.

7 Training for Classification of Smells

In order to classify the feature vector, we allocate the desired output for the input feature vector where it is five–dimensional vector as shown in Table 3 since we have added the coefficient of variation to the usual feature vector to reduce the variations for smells. The training has been performed until the total error becomes less than or equal to $.5 \times 10^{-2}$ where $\eta = .2$. Note that the training data set is the first sample data among three repeated data of A, B, C, and D. After that, the second sample data are selected as the training data set and the third samples data are selected as the training data set. The test data sets are selected as the remaining data sets except for the training data set.

Table 3 Training data set for acetaldehyde (A), ethylene (B), methyl-mercaptan (C), and hydrogen sulfide (D).

Labels	A	B	C	D
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

8 Smell Classification Results

8.1 Single Smell Classification Results

Using a layered neural network with five inputs and four output, we trained the neural network for one training data until the total error becomes 0.001. After training, we checked the remaining two data set. For two test data, correct classification rates became 100% and using the leave-one-out cross validation check, correct classification rates were100%.

8.2 Mixed Smell Classification Results

Using the trained neural network, we tested the classification results whether the original two smells could be classified or not. The classification results are shown in Table 4. From this results, the classification results are not so good since there are many misclassification results came out. Especially, C&D has been classified acetaldehyde (A) because hydrogen sulfide (D) is too strong chemical reaction even if the very thin density levels as well as acetaldehyde (A). Thus, we must reconsider our method to classify the mixed smells classification. One of the approaches is to reduce the maximum values of smell data in order to prevent a sole winner case.

Therefore, we use the following formula to reduce the effect of a sole winner.

$$z = x - \alpha y \tag{1}$$

where x is a feature of the smell (a maximum value of measurement data), y is the top value of each row Table 4 and α is a constan which has been determined by using a genetic algorithm. In this case we have obtained α=0.3. Using this value, we have had the results of Table 5.

Table 4 Classification results for mixed smells of methyl-mercaptan (C), and hydrogen sulfide (D).

Mixing rates	A	B	C	D
1:1 (C&D)	4	3	6	6
1:5 (C&D)	9	0	0	9
5:1 (C&D)	9	0	0	9

Table 5 Classification results when we reduce the largest smell data effect by (1).

Mixing rates	A	B	C	D
1:1 (C&D)	7	2	7	2
1:5 (C&D)	7	0	2	9
5:1 (C&D)	6	0	3	9

9 Conclusions

We have proposed a classification method of mixed smell sensing data using a layered neural networks. The mixed smells classification is not so simple as single smell classification cases. One way seems to use the mixed smells directly as mixed smell of two substances. In that case we need the classification boundary more and more big world. Therefore, we have proposed one way to classify the mixed smells classification from single classification neural network.

Acknowledgment This research has been supported by Grant-in-Aid for Scientific Research for (B) No. 24360141 and Grant-in-Aid for Scientific Research for Challenging Exploratory Research No.25630180, Japan Society for the Promotion of Science (JSPS) and we wish to thank JSPS for their support.

References

1. Norman, A., Stam, F., Morrissey, A., Hirschfelder, M., Enderlein, D.: Packaging effects of a novel explosion-proof gas sensor. Sensors and Actuator B. **114**, 287–290 (2003)
2. Baric, N., Bucking, M., Rapp, M.: A novel electronic nose based on minimized saw sensor arrays coupled with same enhanced headspace analysis and its use for rapid determination of volatile organic compounds in food quality monitoring. Sensors and Actuator B. **114**, 482–488 (2006)
3. Young, R., Buttner, W., Linnel, B., Ramesham, R.: Electronic nose for space program applications. Sensors and Actuator B. **93**, 7–16 (2003)
4. Milke, J.: Application of neural networks for discreminating fire detectors. In: 1995 International Conference on Automatic Fire Detection, AUBE 1995, Duisburg, Germany (1995)
5. Charumpom, B., Yoshioka, M., Fujinaka, T., Omatu, S.: An e-nose system using artificial neural networks with an effective initial training data set. IEEJ. Trans. EIS **123**, 1638–1644 (2003). Japan
6. Fujinaka, T., Yoshioka, M., Omatu, S., Kosaka, T.: Intelligent electronic nose systems for fire detection systems based on neural networks. Int. J. of Adv. in Intelligent Systems **2**, 268–277 (2009)

Constraint Solving-Based Itineraries for Mobile Agents

Ichiro Satoh

Abstract Itineraries of mobile agents among multiple nodes seriously affect the availability and performance of mobile agent-based processing. Such itineraries tend to be complicated, for example, the order of the nodes that agents should visit may be alternative or commutable. This paper proposes a framework for specifying constraints on the itineraries of agents and solving the itineraries that can satisfy the constraints. The contribution of this framework is to automatically generate the itineraries of mobile agents among computers from application-specific constraints. A prototype implementation of this framework and its application were built on a Java-based mobile agent system.

Keywords Mobile agent · Constraint satisfaction problem · Agent route · Agent itinerary

1 Introduction

Mobile agents are autonomous programs that can travel from computer to computer in a network under their control [5]. The state of the running program is saved, by being transmitted to the destination. The program is resumed at the destination continuing its processing with the saved state. They can provide a convenient, efficient, and robust framework for implementing distributed applications to reduce the latency and bandwidth of client-server applications and reducing vulnerability to network disconnection. Although not all applications for distributed systems will need mobile agents, many other applications will still find mobile agents the most effective technique for implementing all or part of their tasks.

I. Satoh(✉)
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
e-mail: ichiro@nii.ac.jp

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_13

This paper addresses the itineraries of mobile agents, because existing mobile agent platforms explicitly or implicitly assume that, when developing mobile agents that visit multiple computers, developers must specify the static and straight routes of the agents. However, such routes among multiple computers are complicated and dependent on their applications and network topologies. The itineraries of mobile agents greatly affect their efficiencies in addition to their achievement. It is also almost impossible to define efficient itineraries among multiple nodes, without having any knowledge of the network.

The contribution of this paper is to enable mobile agents to generate their concrete itineraries from application-specific requirements on the orders of their movements and to migrate the agents among computers through their itineraries. The framework proposed in this paper uses as a constraint satisfaction problem (CSP) into mobile agents. It allows us to specify partial conditions that agents require as constraints on their itineraries to achieve their applications and dynamically generate itineraries that can satisfy the conditions. Our current implementation is built on a Java-based mobile agent system, but the framework itself can be used in other mobile agent platform, including Aglets [3], because our constraint solver is available outside the platform and the interpreter of agent itineraries generated based on the framework is constructed as a Java library, which agents can carry.

2 Basic Approach

Suppose mobile agents for accessing remote database systems as example scenarios.

- If a mobile agent can travel among multiple database servers to query and aggregate interesting data from the servers without writing on any of them, the order of its movement is independent of its achievement and the servers (the upper of Fig. 1).
- If a mobile agent queries and carries data from a database server and reflects the data on other database servers, the order of its movement affect the content of the servers (the lower of Fig. 1).

Therefore, in the second the order is imposed on the itinerary of such an agent as a constraint, in the sense that the results of the application-specific behavior of the agent is often dependent on the agent's itineraries. However, existing mobile agent systems explicitly or implicitly assume that the itinerary of each mobile agent is defined as just the address of a computer or a sequential and static list consisting of the addresses of computers. It is almost impossible for developers to specify agent itineraries that can satisfy application-specific requirements in addition to network topologies, whose nodes and connections may dynamically change.

To solve this problem, we use the notion of CSP as an approach to generating agent itineraries from the constraints corresponding to application-specific requirements on the migration of agents. As mobile agents are defined with general-purpose programming languages, such as Java, it is almost impossible to exactly extract only their itineraries from their programs. Our framework defines two languages.

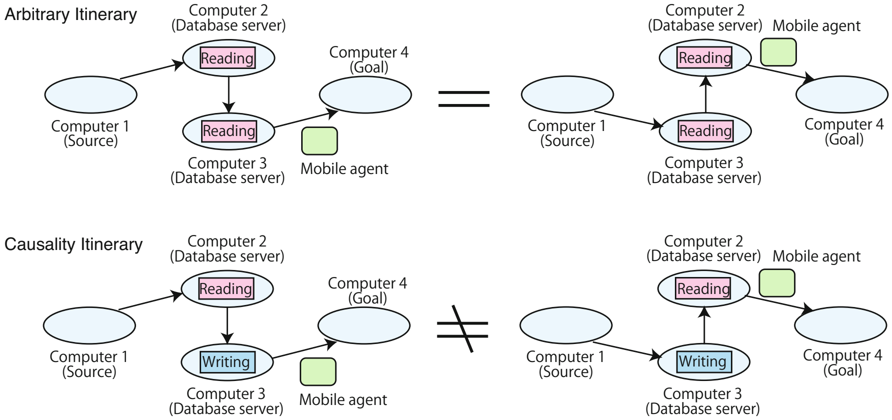


Fig. 1 Agent itineraries and constraints

The first is defined to specify constraints imposed on the itineraries of mobile agents. The second is defined to specify agent itineraries. The framework generates agent itineraries specified in the second according to the constraints specified in the first by using techniques for CSP.

3 Constraint-Based Agent Itinerary Generation

This section defines two languages for specifying agent itineraries and constraints, called C and E. To accurately express such itineraries, we need to define a specification language based on a process calculus such as CCS [4].

Definition 1. The set $\mathcal{E}$ of expressions of the language, ranged over by $E, E_1, E_2, \ldots$ is defined recursively by the following abstract syntax:

$$E ::= \mathtt{0} \quad | \quad \ell \quad | \quad E_1 \,;\, E_2 \quad | \quad E_1 + E_2 \quad |$$

where $\mathcal{L}$ is the set of location names, ranged over by $\ell, \ell_1, \ell_2, \ldots$. We often omit $\mathtt{0}$.

□

Intuitively, the meaning of the terms is as follows:

- $\mathtt{0}$ represents a terminated itinerary.
- ℓ represents agent migration to a node whose name or network address is ℓ.
- $E_1 \,;\, E_2$ denotes the sequential composition of two constraints E_1 and E_2. If the migration of constraint E_1 terminates, then the migration of E_2 follows that of E_1.
- $E_1 + E_2$ represents an agent moving according to either E_1 or E_2, where the selection can be explicitly performed by the processing of the agent.

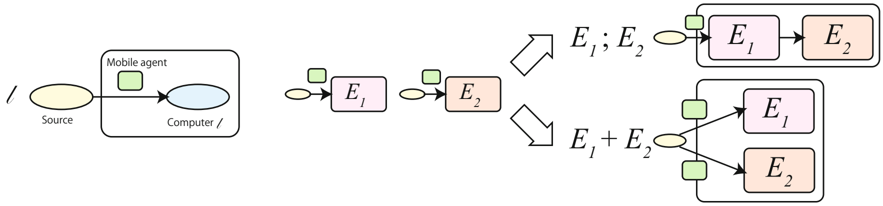


Fig. 2 Agent itinerary language

Figure 2 shows basic agent itineraries specified in the $\mathcal{E}$ language.

Example 1

- ℓ_1 ; ℓ_2 ; ℓ_3 means that an agent migrates to ℓ_1 and then to ℓ_2. Finally, it migrates to ℓ_3
- ℓ_1 ; $(\ell_2 + \ell_3)$ means that an agent migrates to ℓ_1 and then to one of either ℓ_2 or ℓ_3.

The semantics of the language is defined by the following labeled transition rules:

Definition 2. The language is a labeled transition system $\langle\, \mathcal{E},\ \mathcal{L}\,\{ \xrightarrow{\ell} \subseteq \mathcal{E} \times \mathcal{E} \mid \ell \in \mathcal{L}\,\}\,\rangle$ is defined as the induction rules below:

$$\frac{-}{\ell \xrightarrow{\ell} 0} \qquad \frac{E_1 \xrightarrow{\ell} E_1'}{E_1 \;;\; E_2 \xrightarrow{\ell} E_1' \;;\; E_2} \qquad \frac{E_1 \xrightarrow{\ell} E_1'}{E_1 + E_2 \xrightarrow{\ell} E_1'} \qquad \frac{E_2 \xrightarrow{\ell} E_2'}{E_1 + E_2 \xrightarrow{\ell} E_2'}$$

where 0 ; E is treated as being syntactically equal to E. □

The framework provides a library for interpreting the itineraries of agents and migrating the agents to their itineraries. Next, we define a language for specifying constraints on agent itineraries.

Definition 3. The set $\mathcal{C}$ of expressions of the language, ranged over by C, C_1, C_2, ... is defined recursively by the following abstract syntax:

$$C ::= 0 \quad | \quad \ell \quad | \quad C_1 \succ C_2 \quad | \quad C_1 \,\#\, C_2 \quad | \quad C_1 \,\&\, C_2 \quad | \quad C_1 \,\%\, C_2$$ □

Intuitively, the meaning of the terms is as follows:

- ℓ means that an agent itinerary need to contain a node whose name or network address is ℓ, as one of its destinations.
- $C_1 \succ C_2$ denotes that agent itinerary must satisfy C_2 after satisfying C_1, where C_1 and C_2 are constraints.
- $C_1 \# C_2$ means that an agent itinerary has to satisfy at the least one of either C_1 or C_2, where C_1 and C_2 are constraints.
- $C_1 \& C_2$ means that an agent itinerary has to satisfy two constraints specified as C_1 and C_2, although iterates for C_1 and C_2 constraints can be interleaved.
- $C_1 \% C_2$ means that an agent itinerary need to satisfy at the least one of either C_1 before C_2 or C_2 before C_1 in its itinerary, where C_1 and C_2 are constraints.

Example 2

- *Causality.* An agent travels between two database servers, named ℓ_1 and ℓ_2, and to read data from the ℓ_1 server and then write the data to the ℓ_2 server. The agent has to migrate to ℓ_1 before ℓ_2 as follows:

$$\ell_1 \succ \ell_2$$

- *Data Aggregation.* An agent travels between two database servers, named ℓ_1 and ℓ_2, to aggregate data from the servers and then reflects the data to another server, named ℓ_3

$$(\ell_1 \parr \ell_2) \succ \ell_3$$

- *Independent Causality.* An agent has to do two independent tasks, where the first task is to read data from the ℓ_1 database server and then write the data to the ℓ_2 database server and the second task is to read data from the ℓ_3 database server and then write the data to the ℓ_4 database server.

$$(\ell_1 \succ \ell_2) \,\&\, (\ell_3 \succ \ell_4)$$

Expressions in the $\mathcal{C}$ language can be transformed into the structures of directed graphs according to the rules proposed in this paper, where each directed graph is defined by a set of operations (vertices) and dependencies (edges). The operation is an equation with the left side consisting of a dependency (acting as a short-term name for the result of the operation), and the right side being the application of an operator to zero or more dependencies. An operation is a source for the dependency on the left side of the equation and a sink for dependencies listed on the right side. A dependency has exactly one source but can have many sinks. Our rules map constraints into graphs. The itinerary of each agent is generated as a path on the graph from at most one source node to destination nodes.

Example 3

- The first constraint of Example of 2 is transformed into agent itinerary: $\ell_1 \; ; \; \ell_2$.
- The second constraint of Example of 2 is transformed into two agent itineraries: $\ell_1 \; ; \; \ell_2 \; ; \; \ell_3$ or $\ell_2 \; ; \; \ell_1 \; ; \; \ell_3$.
- The third constraint of Example of 2 is transformed into an agent itinerary: $(\ell_1 \; ; \; \ell_2 \; ; \; \ell_3 \; ; \; \ell_4) + (\ell_1 \; ; \; \ell_3 \; ; \; \ell_2 \; ; \; \ell_4) + (\ell_1 \; ; \; \ell_3 \; ; \; \ell_4 \; ; \; \ell_2) + (\ell_3 \; ; \; \ell_1 \; ; \; \ell_2 \; ; \; \ell_4) + (\ell_3 \; ; \; \ell_1 \; ; \; \ell_4 \; ; \; \ell_2) + (\ell_3 \; ; \; \ell_4 \; ; \; \ell_2 \; ; \; \ell_3)$

Next, we describe the process of CPS-based automatic generation of agent itineraries.

- **Step 1:** A mobile agent is loaded from storage and initialized, where its application-specific requirement on its movement among computers is specified in the $\mathcal{C}$ language.
- **Step 2:** The requirement is solved by using the graph-based CSP approach mentioned in this section. As a result, an agent itinerary described as an expression of

the $\mathcal{E}$ language that can satisfy the constraints corresponding to the requirement is automatically generated and is given to the agent.

– **Step 3:** The agent is migrated among computers through its generated itinerary.

4 Implementation

There have been many mobile agent platforms so far. Although the framework itself is independent of any platforms, the current implementation is constructed as our original mobile agent platform. As shown in Figure 3, the platform consists of three parts: runtime system, agent itinerary interpreter, and constraint solver. The first is responsible for migrating mobile agents and executing their application-specific tasks defined as Java programs, the second for interpreting the itinerary of each agent, and third for solving constraints on agent itineraries.

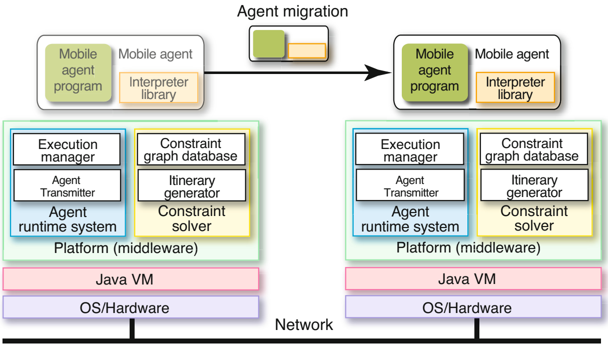


Fig. 3 System structure

4.1 Runtime System

Runtime systems are executed on computers for executing and migrating agents to other runtime systems. Each runtime system is built on the Java virtual machine (Java VM), which conceals differences between the platform architectures of the source and destination computers. Each runtime system governs all the agents inside it and maintains the life-cycle state of each agent. When the life-cycle state of an agent changes, e.g., when it is created, terminates, or migrates to another computer, the runtime system issues specific events to the agent. Each runtime system can exchange agents with another system through a TCP channel using mobile-agent technology. When an agent is transferred over the network, not only the code of the agent but also its state is transformed into a bitstream and then the bit stream is transferred to the destination. Therefore, the arriving agents can continue to execute its processing at the destination. The runtime system on the receiving side receives and unmarshals the bit stream.

4.2 Agent Itinerary Interpreter

The framework provides a Java-library for interpreting expressions in $\mathcal{E}$ based on the semantics of Definition 2. Each agent is equipped with the library to evaluate the expressions. The library invokes API for agent migration with the next destination to migrate the agent to there. Each runtime system periodically monitor connection to neighboring runtime systems and the availability of the systems by using a peer-to-peer manner through multicasting UDP. The $+$ operator is evaluated as one of either sub-itineraries by this current runtime system according to the connectivity and availability of the destinations of the sub-itineraries. When there are two or more possible sub-itineraries, one of the sub-itineraries is selected by the runtime systems. Since the size of the library is smaller than 10 KB, the cost of migrating the library can be ignored in most practical applications.

4.3 Constraint Solver for Agent Itinerary

Each constraint is specified on a dependency graph. Constrained path problems have to do with finding paths in graphs subject to constraints. One way of constraining the graph is by enforcing reachability between nodes. For instance, it may be required that a node reaches a particular set of nodes by respecting some restrictions like visiting a particular set of nodes or edges in a given order specified in our constraint language. Our constrain solving system is responsible for generating agent itineraries as paths that can satisfy constraints. Our framework has three arguments for solving constraints: (1) a directed graph, i.e., a directed graph with a source node; (2) the relation graph on nodes and edges of one or more directed graphs; and (3) the transitive closure of the directed graphs. The second represents operators in $\mathcal{C}$ and the third is to find paths that can satisfy multiple constraints connected with the relations in the second. Agent itineraries are generated through the reachability on the transitive closure of the directed graphs by finding a simple path in a directed graph containing a set of mandatory nodes. A simple path is a path where each node is visited once, i.e., given a directed graph, a source node, a destination node, and a set of mandatory nodes, we want to find a path in the graph from the source to the destination, going through mandnodes and visiting each node only once.

4.4 Evaluation

The framework is constructed and available on Java version 7 or later. Its current implementation was not built for performance, but a basic agent migration experiment was done using eight computers (Xeon E5 3.5 GHz), named $\ell_1,\ldots,\ell_8$, connected through a giga-ethernet network. We measured the cost of migrating a null agent (a 5-KB agent, zip-compressed) between two computers to be 28 ms. The cost of agent migration included that of opening TCP transmission, marshaling the agents, migrating the agents from their source computers to their destination computers, unmarshaling the agents, and verifying security. The costs of solving three kinds of constraints were 7 ms, 8 ms, and 10 ms, where the first was specified as $(\ell_0 \succ \ell_1) \,\&\, (\ell_1 \succ \ell_2) \,\&\, \cdots \,\&\, (\ell_7 \succ \ell_8)$, the second as $(\ell_0 \succ \ell_8) \,\#\, (\ell_1 \succ \ell_8) \,\#\, \cdots \,\&\, (\ell_7 \,\%\, \ell_8)$, and the third as $(\ell_0 \succ \ell_8) \,\&\, \cdots \,\&\, (\ell_7 \succ \ell_8)$. We already implemented typical applications of mobile agents with the proposed framework.

5 Application

Remote information retrieval is one of the most traditional applications of mobile agents. An agent migrates to four database servers and queries about certain data from the servers, named ℓ_1, ℓ_2, and ℓ_3, and then carries the results of their queries to a specified computer, e.g., its client, named ℓ_4. The order of visiting the servers are arbitrary if their queries have no side effect to the servers. Therefore, a constraint on such an agent is described as follows:

$$(\ell_1 \,\%\, \ell_2 \,\%\, \ell_3) \succ \ell_4$$

The framework generated the following itinerary that could satisfy the constraint.

$$(\ell_1 \,;\, (\ell_2 \,;\, \ell_3 + \ell_2 \,;\, \ell_1) + \ell_2 \,;\, (\ell_1 \,;\, \ell_3 + \ell_3 \,;\, \ell_1) + \ell_3 \,;\, (\ell_1 \,;\, \ell_2 + \ell_2 \,;\, \ell_1)) \,;\, \ell_4$$

The generated itinerary consisted of six alternative paths. One of them was selected according to the reachability of ℓ_1, ℓ_2, and ℓ_3. In fact, the mobile agent migrates among nodes through the itinerary with keeping its constraints.

6 Related Work

Many mobile agent systems have been developed over the last twenty years. Most of these studies explicitly or implicitly assume that the itineraries of mobile agents were defined in the development phases of their agents. Like our framework, there have been several attempts to define theoretical foundations into mobile agents.

The ambient calculus [1] allows mobile agents (called ambients in the calculus) to contain other agents and to move as a whole with all its subcomponents. The Seal calculus [6] is similar to the mobile ambients and ours in its expressiveness of hierarchical structure of mobile agents, but its main purpose is to reason about the security mechanism of mobile agents. The Polis language [2] is a theoretical framework for specifying and analyzing mobile entities, including mobile codes and mobile agents, which can contain other entities inside them. However, it is not executable and needs a kind of shared memory over remote nodes, whereas our framework can operate reusable mobile agents for network management in a decentralized manner. Unlike software agents, which are not mobile, there have been a few attempts to merge CSP and mobile agents.

7 Conclusion

We presented a framework for enabling the itineraries of mobile agents to be specified as constraints and the agents to be migrated through paths that could satisfy the constraints. As a result, mobile agents could migrate among multiple nodes to perform their tasks at each of the visited nodes. The contribution of this framework is to automatically generate the itineraries of mobile agents among computers from application-specific constraints. In fact, it is useful in operating mobile agent-based applications, because the itineraries of agents seriously affect the availability and performance of their processing. This paper presented a framework for specifying constraints on the itineraries of mobile agents and solving the itineraries that could satisfy the constraints. Since the framework is independent of the underlying mobile agent platforms, it is useful to other platforms.

References

1. Cardelli, L., Gordon, A.D.: Mobile ambients. In: Foundations of Software Science and Computational Structures. LNCS, vol. 1378, pp. 140–155 (1998)
2. Ciancarini, P., Franze, F., Mascolo, C.: Using a coordination language to specify and analyze systems containing mobile components. ACM Transactions on Software Engineering and Methodology **9**(2), 167–198 (2000)
3. Lange, B.D., Oshima, M.: Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley (1998)
4. Milner, R.: Functions as Processes. Mathematical Structures in Computer Science **2**(2), 119–141 (1992)
5. Satoh, I.: Mobile agents. In: Handbook of Ambient Intelligence and Smart Environments, pp. 771–791. Springer (2010)
6. Vitek, J., Castagna, G.: Seal: a framework for secure mobile computations. In: Proceedings of Internet Programming Languages. LNCS 1686, pp. 47–77 (1999)

δ-Radius Unified Influence Value Reinforcement Learning

J. Alejandro Camargo and Dennis Barrios-Aranibar

Abstract Nowadays Decentralized Partial Observable Markov Decision Process framework represents the actual state of art in Multi-Agent System. Dec-POMDP incorporates the concepts of independent view and message exchange to the original POMDP model, opening new possibilities about the independent views for each agent in the system. Nevertheless there are some limitations about the communication.

About communication on MAS, Dec-POMDP is still focused in the message structure and content instead of the communication relationship between agents, which is our focus. On the other hand, the convergence on MAS is about the group of agents convergence as a whole, to achieve it the collaboration between the agents is necessary.

The collaboration and/or communication cost in MAS is high, in computational cost terms, to improve this is important to limit the communication between agents to the only necessary cases.

The present approach is focused in the impact of the communication limitation on MAS, and how it may improve the use of system resources, by reducing computational, without harming the global convergence. In this sense δ-radius is a unified algorithm, based on Influence Value Reinforcement Learning and Independent Learning models, that allows restriction of the communication by the variation of δ.

Keywords Multi-Agent Systems, Artificial Intelligence, Markov Decision Process

1 Introduction

It is important to mention that for single agent problems the convergence to the solution is directly related to the exploration and approximation to the solution; thus,

J.A. Camargo(✉) · D. Barrios-Aranibar
San Pablo Catholic University, Arequipa, Peru
e-mail: alejandro.camargo@gmail.com

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_14

the learningn process is guided by the exploration process. Instead, the MAS solution is not about the independent process of convergence for each agent, it is about the capability of a group of agents to cooperate and found the system solution as a whole.

The MAS cooperation capability is described in [2] as coordination and collaboration, necessary actions that describe different kind of problems in MAS, but both are based on the idea of establishing cooperation between agents.

With coordination is possible to avoid overlapping tasks and/or the interruption of process by other agents [6]. To be able to coordinate in a group of agents it is necessary to establish communication between them.

The communication, as described in [7][8] , has different levels about what and when the information must be shared:

1. Instantaneous information sharing
2. Episode information sharing
3. Share the learned decisions policies

The instantaneous information sharing is directly related to the interaction of each agent with the world. This means that the agent will communicate its $sensations$ about the world, its own actions or the rewards from the world.

In the episode information sharing, the agent do not share immediately the information. Is necessary to complete the collection of information until the end of the episode of actions to share the information.

Finally, the policies sharing is the only case in which the agent shares the information related to itself. The inner agent information may be its partial or complete knowledge.

As [7] shows, the knowledge sharing affects the convergence in different ways for each of the exposed cases. The instantaneous information sharing affects the convergence only if the shared information is relevant enough to the problem solution. The episode information sharing is, in this case, the assurance of relevance by grouping the information and the details though the episode sequence; this communication kind speeds up the system convergence. Finally the policies sharing creates an interesting difference at the final results, it allows the agents to develop new behaviors. Is important to remark that new behaviors are uniques, although, in some cases they do not generate a reward improve for the agents as individual beings, but it improves the system performance.

With the new behavior exploration for MAS as an objective, we select the Influence Value Reinforcement Learning as the base for the δ-Radius Communication Model. IV-Q-learning is a Q-Learning adaption for MAS that explores and explodes the possibility of a full agent communication system.

At this point is important to remark some differences with Dec-POMDP framework. First of all the present proposal is not focused on the message communication content as some Dec-POMDP variations are[4] [5]. Our focus is “whom should a certain agent must share information with?”. In this sense the investigation will show the impact of the way the agents construct their communication relationship.

2 Multi-Agent Systems and Markov Decision Process

Formally proposals on MAS are commonly defined in base of Markov Decision Process, a formal model that allows the definition of a world environment for the agents as a set of states and actions. This basic structure is a graph model world in which is possible the exploration of new states through the execution of actions of the agents, the graph transitions.

For MDP, every agent interaction in the world is directly related and applied to the actual world state, the agents decisions are unaffected by any interactions with other agents in the same environment. At this point is clear that only the restrictions on the world (its own configuration) and the rewards on it will affect the agents behavior.

This simple model allows, in manny ways, the definition of several problems, even some simple MAS problems, but it is restricted at the same time about possibility to define or establish the interaction of the agents in the same environment.

A more general model for MAS is the based in stochastic games, it allows the definition of independent states from the world states, they are connected by transitions with a probabilistic model as an underground for each agent to model the agents behavior. Even this model is not enough to ensure a complete representation of MAS with the coexistence of the agents in the same environment.

The main problem about the classic representations of MAS as MDP or stochastic games are about the world interaction with the agents and between them. Both models allow the interaction of agents with the world without taking into account the coexistence of the others and the changes that other agents may provoque in the world.

As shows in [3][1] the use of POMDP supports the coexistence of agents into the same environment with independent actions and the capability to modify the world state independently to the other agents perception.

The capability to modify or change the world independently to the other agents perception state is one of the most important capabilities of POMDP, because, even when an agent modifies a world state other agents could or not perceive it.

Depending if they need or not that information or even if they are in the capability to perceive the change. The capability to perceive change in the world state is modeled in POMDP through a Y function which restricts the vision of the world for the agents, based in this, it is possible to model the world as a non deterministic problem for each agent.

This perception is restricted for Dec-POMDP as $O : \Omega \times A \times S$, the probability for an agent to perceive a change given an action on A, that results on a state of S.

3 Dec-POMDP

POMDP introduced by Åström [11] solves problems where not all the information is available or is not completely trustworthy. Taking into account both points, Dec-POMDP was created as a frame that supports the exploration of an uncertainly world with black spaces or noise[1].

This original framework defines a different point of view ω_i for each agent in the system. It limits the agents capabilities to perceive changes in the real world state and at the same time establish a formal difference between what can each agent perceive.

Another important contribution of Dec-POMDP is the definition of O, the set of observation probabilities, that defines if it is possible for an agent to perceive the environment changes from an state and/or action. The most remarkable contribution about Dec-POMDP is related to the original POMDP, the agent's limitations on the environment perception and the establishment of the restriction on the agent's decision system, that is based only on each agent own vision perception of the environment.

Definition 1. Decentralized Control of Partially Observable Markov Decision Processes

- Æ , a finite set of agents {æ$_0$, æ$_1$, ..., æ$_n$}.
- S, a finite set of states with designated initial state distribution b_0 .
- A$_i$, a finite set of actions for each agent, æ$_i$ with A = $\times_i A_i$ the set of joint actions, where $\times$ is the Cartesian product operator.
- P(s,s',a) , a state transition probability function, P : S $\times$ S $\times$ A $\rightarrow$ [0, 1], that specifies the probability of transitioning from states s $\in$ S to s' $\in$ S when these to factions $\overrightarrow{a}$ $\in$ A are taken by the agents. Hence, P(s,s, $\overrightarrow{a}$) = Pr(s'| $\overrightarrow{a}$, s).
- R, a reward function: R : S $\times$ A $\times$ R, the immediate reward for being in state s $\in$ S and taking the set of actions $\overrightarrow{a}$ $\in$ A.
- Ω_i, a finite set of observations for each agent, i, with $\Omega = \times_i \Omega_i$ the set of joint observations.
- O, an observation probability function: O : $\Omega \times A \times S \rightarrow$ [0, 1], the probability of seeing the set of observations $\overrightarrow{o}$ $\in \Omega$ given the set of actions $\overrightarrow{a}$ $\in$ A was taken which results in state s' $\in$ S, Hence O($\overrightarrow{o}$, $\overrightarrow{a}$,s') = Pr($\overrightarrow{o}$ | $\overrightarrow{a}$, s').
- h, the number of steps until the problem terminates, called the horizon.

4 δ-Radius Communication Model

As previously described, the true form of MAS is not the convergence based on the singular work of the agents, is the group work as a whole. To talk about group work is always necessary talk about cooperation, that is the true form of MAS, its edge stone.

Once the cooperation problem has been described as the edge stone of MAS convergence is important to establish a communication model for agents that includes a formal communication definition. δ-radius communication model is our proposed model to describe the communication relationship on MAS.

Definition 2. δ-radius Communication Model

- P, a hyperplane with base = $\{\overrightarrow{x_0}, \overrightarrow{x_1}, \ldots, \overrightarrow{x_m}\}$.
- Æ, a finite set of agents with coordinates $æ_{Coord} \subseteq P$
- γ_i, a set of agent pairs $\{(æ_i, x) \mid x \in Æ)\}$ related to the agent $æ_i \in Æ$, with $\eta = \times_i \gamma_i$, the joint set of communication relationships between agents in Æ, where $\times$ is the Cartesian product operator.
- δ, the max distance allowed to establish a relationship between two agents. $\delta \in \mathbb{R}$.
- f($æ_x$, $æ_y$), a distance function, f : Æ × Æ → $\mathbb{R}$. f = f' | f': P × P → $\mathbb{R}$. The definition of f could be asymmetric if the problem requires.

Using δ-radius model it is possible to define a relationship for a set of agents Æ with base in their own characteristics. This characteristics are formal named as P, an hyperplane in which f' is defined as complex distance function. f determines if two different agents will be able to establish communication using δ as a threshold limitation.

Finally an extended definition of η and γ is required to describe the communication rules.

Definition 3. γ_i : Relationships set for the agent $æ_i$. Let η , the set of relations in Æ defined as:

$$\eta = \{(\, æ_i, æ_j) \mid æ_i, æ_j \in Æ; æ_i \neq æ_j \,\}$$

Then the γ_i is defined as :

$$\gamma_i = \{(\, æ_i, x) \mid æ_i, x \in Æ\ ; æ_i \neq x;\ f(æ_i, x) \leq \delta\}$$

5 δ-Radius Unified IVRL

Based on IVRL, the definition of a communication system with limitations allows us to define a more restrictive and controlled model for reinforcement learning in MAS. The new δ-radius IVRL.

The original IVRL proposal is about to adapt RL algorithm for MAS to improve the positive results with base in the collaboration between the agents. This improve is based on the definition of the influence value, a calculated variable from the *opinion* of the agents about the executed actions of a specific agent.

The IV for an agent i is calculated as:

$$IV_i = \sum_{j \in \{1:n\}-i} B_i * Op_j(i)$$

Where $Op_j(i)$ is the opinion evaluation function and B_i is the influence coefficient for the agent i over the actual agent. The opinion value is calculated as:

$$OP_i(j) = \begin{cases} ((r_i) - Q(a_i)) * OI(a_j) & if(r_i - Q(a_i)) < 0 \\ ((r_i) - Q(a_i)) * (1 - OI(a_j)) & if(r_i - Q(a_i)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Where a_j is the last choosen action, r_i the reward from that action execution, $Q(a_i)$ the inner state of the agent related to the action a_i, and finally Oi is the Occurrence Index of the action a_i.

The adaption of the new proposal is focused in the IV definition in which the restriction on j changes from $j \in \{1 : n\} - i$ to $(\alpha_i, j) \in \gamma_i$, where γ_i is the set of relations of the agent α_i.

Definition 4. $\delta\ Influence\ Value$

$$\delta IV_i = \sum_{(j,\alpha_i)\in\gamma_i} B_i * Op_j(i)$$

This simple adaption of the main IV formula restricts the scope action of the influence value itself; and in consequence reduces the computational cost from n^2 to $c * n$.

Algorithm 1. δ-radius Influence Value Reinforcement Learning Initialization

```
procedure δ- RADIUS IVRL INITIALIZATION
    $\eta = \{\}$
    for all $æ_i \in Æ$ do
        $\gamma_i = \{\}$
        for all $æ_j \in Æ - æ_i$ do
            $d = f(æ_i, æ_j)$
            if $d \leq \delta$ then
                $\gamma_i = \gamma_i \cup \{(æ_i, æ_j)\}$
        $\eta = \eta \cup \{\gamma_i\}$
```

It is important to remark that the new δ-radius IVRL is a unified model of the original IVRL and Independent Learning. Similar to the λ in Temporal Differences, δ variation in the new algorithm changes the algorithm itself from Independent Learning to IVRL and $viseversa$. Both cases are particular values for δ; if δ is a negative value, then is not possible establish any relation between agents, because there is no negative distances, this is Independent Learning. By other hand, if $\delta = \infty$ then each agent will establish a relation with every other agent in the system, this is the case of IVRL. The capability to modify δ creates new possibilities to explore the agents behavior in base to the communication restrictions.

Algorithm 2. δ-radius Influence Value Reinforcment Learning Step

```
procedure $\delta$- RADIUS IVRL STEP
    for all $\ae_i \in \AE$ do
        Based on the policy $Q_i$ select an action $a_i$ from the state $s_i(t)$
        Execute $a_i$
        Go to state $s_i(t+1)$
        Update the reward $r_i(t+1)$
        $RV_i = r_i(t+1) + maxQ(s_i(t+1), a_i) - Q(s_i(t), a_i(t))$
        for all $\ae_j \in \AE - \ae i$ do
            $OP_i(j) = \begin{cases} ((r_i) - Q(a_i)) * OI(a_j) & if(r_i - Q(a_i)) < 0 \\ ((r_i) - Q(a_i)) * (1 - OI(a_j)) & if(r_i - Q(a_i)) > 0 \\ 0 & \text{otherwise} \end{cases}$
        $\delta IV_i = \sum_{(j,\alpha_i)\in\gamma_i} B_i * Op_j(i)$
        $Q_i(s(t), a_i(t)) = Q_i(s(t), a_i(t)) + \alpha(r_i(t+1) + \sigma maxQ(s(t+1), a_i) - Q_i(s(t), a_i(t)) + \delta IV_i)$
        $t = t + 1$
```

6 Experiments

6.1 *The Convergence Problem*

In manny different game theory problems only one answer or goal exists, but for certain problems there are more than one right answer. One of this special problems is the prisoner's dilemma. A game theory problem in which is possible establish two convergence points. The first is related to Nash Equilibria and the second one related to Pareto Optimum.

As a competitive definition, Nash equilibria is focused on the maximization of the own reward. If no player can benefit by changing strategies while the other players keep their unchanged, then they are on Nash Equilibria.

In the other hand, Pareto Strategy is focused on the best reward of the team as a whole, in this sense, Pareto allows both players to be the better off, without making the other player worse off in comparison with other scenarios.
The rules for the basic two player prisoner's dilema are:

- The reward for each agent is based on the answer of both and each agent reward is independent.
- If A and B stays in silent, they will serves 1 year in prison.
- if A and B betray each other, they will serves 3 years in prison.
- If the agent A stays silent and B betrays A, then A will serves 6 years in prison and B will be free, and vice versa.

A simple extension of the game is possible for n-players as:

- If all players are in silent, then they will serve in prison 1 years

- If all players betray the group, then they will serve in prison 3 years
- Otherwise, the players who choose to betray the group will be free, and the others will serve for 6 years.

The convergence to the Nash Equilibria of the Prisoner's game is under the betray option for all agents, 3 years on jail are better than 6 and if some other agent change its option to stays silence then the actual agent will be free.

The case in which all are silent is more complicated to achieve, because if at least one player chooses to betray the others, then he will be free but the others will stay in prison for 6 years. This scenario corresponds to Pareto Optimum.

6.2 Results

The following experiment is focused on the impact of the communication between the agents. We want to show how the convergence conditions can be easily altered by the influence of other agents on the same environment.

The experiment is an adaption of the original Prisoner's dilema with only two players, as previously described the game rules have been adapted and ten agents are now involved in the experiment. The agents are distributed in a circular list, the use of a radius 1 means that the agent n will have a communication channel with the agents $n-1$ and $n+1$. With radius 2 the agent n will have a communication channel with the agents $n-2$, $n-1$, $n+1$ and $n+2$. The same process for every agent in the system for each radius tested from $\delta = 0$ until $\delta = 5$.

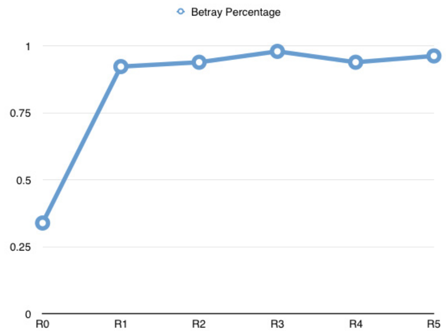


Fig. 1 Average Percentage of Agent's Choice

As the figure 1 shows, initially the agent's convergence is over the silent option, Pareto Optimum; but once the minimal communication is established the agent's choice turns to Nash Equilibria or betray option. To produce this behavior change is only necessary the instance of $\delta = 1$, the minimal communication.

Another important results are about computational cost of the algorithm. As previously mentioned the cost of maintain a communication system for each agent is equal to the cost of the calculous of IV, for δ-Radius is m^2.

For the original IVRL m is equal to n, which means that the calculus cost is n^2. Instead the value of m for δ-Radius is a constant. In conclusion, is possible to achieve the same agent system behavior with a constant computational cost.

In the figure 2 the tendency to betray or Nash Equilibria is evident on how the agent's behavior evolves to a more competitive when a communication is established. Is important to mention that the tendency is taken after 125 iterations. After 1000 or more iterations all the agents in the system with at least a radius of communication equal to one will choose betray. Only the agents with 0 radius have tendency to be silent.

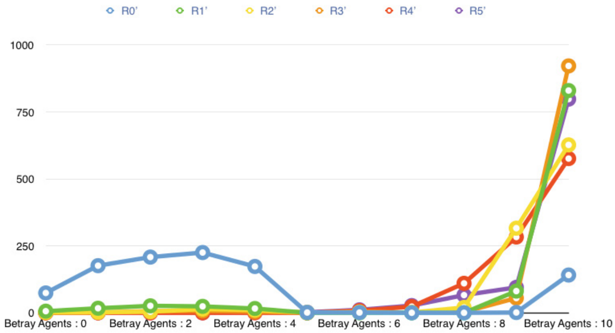


Fig. 2 Agent's Betray Tendency

And finally,in figure 3, the variation of β factor in the adaption of IVRL is showed to ensure that the behavior changes are only a consequence of the communication variations. It shows 3 bar groups, with 6 bars each. Each group represents the percentage of agents in betray option under different radius and different ß values, ß =0.02 , 0.04 and 0.1.

7 Conclusions

In this paper we expose the problem of the communication under a new model, δ-radius, with the intention of showing how the communication on MAS can improve and change the behavior of the agents in different ways.

In contrast with other proposals IVRL use the opinion of other agents as the message in the communication, this is a real world base of how we can learn; but it is necessary to understand its real impact. Our proposal explores the impact of opinion

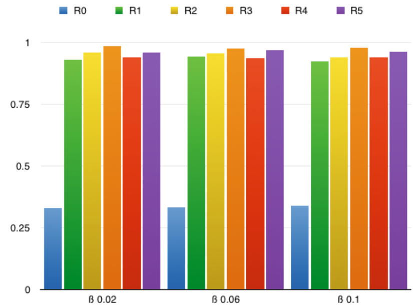


Fig. 3 Agent's Betray Percentage with the Variation of β factor for the IV

message sharing and the limitation of it under the δ variable. This simple change allows the integration of IVRL with Independent learning, a special case with $\delta = 0$.

Depending of the problem the δ variable can improve the convergence or change the agents behavior as in Prisoner's Dilema. In some cases the cost of communication is high, and its limitation is necessary, in this cases is possible explore which configuration of δ-radius can maintain the same behavior. For example, in our problem, after one thousand iterations the behavior or the agents with a full communication system, $\delta = 5$, is the same as the group trained with $\delta = 1$. This results are special interesting for solving problems with a high computational cost.

Normally the computational cost of a MAS with communication is equal to the cost of calculate the message per the number of agents in the system, for each agent in the system: $Message\ cost * n^2$, where n is the number of agents. The possibility to limit the communication reduces the n^2 to $n * m$, where $m \leq n$. m must be enough low so it can be taken as a constant, reducing the computational cost.

Some future works are related to an adaptive variation of the problem to take into account a degradation of β in base of the distance f. This adaption will introduce the concept of trust, with base in closeness for MAS.

Acknowledgements This work was supported by grant FONDECYT 011-2013(Master Program) from the National Council for Science,Technology and Technological Innovation (CONCYTEC-Peru)

References

1. Amato, C., Chowdhary, G., Geramifard, A., Ure, N., Kochenderfer, M.: Decentralized control of partially observable markov decision processes. In: 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 2398–2405, December 2013
2. Barrios-Aranibar, D., Gonçalves, L.M.G.: Learning from delayed rewards using influence values applied to coordination in multi-agent systems. In: VIII SBAI-Simpósio Brasileiro de Automaç ao Inteligente (2007)

3. Barrios Aranibar, D., Gonçalves, L.M.G., de Carvalho, F.V.: Aprendizado por Reforço com Valores de Influência em Sistemas Multi-Agente (2009)
4. Goldman, C.V., Zilberstein, S.: Optimizing information exchange in cooperative multi-agent systems. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, pp. 137–144. ACM, New York (2003)
5. Guestrin, C., Venkataraman, S., Koller, D.: Context-specific multiagent coordination and planning with factored mdps. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada, pp. 253–259 (2002)
6. Pini, G., Gagliolo, M., Brutschy, A., Dorigo, M., Birattari, M.: Task partitioning in a robot swarm: a study on the effect of communication. Swarm Intelligence **7**(2), 173–199 (2013)
7. Tan, M.: Multi-agent reinforcement learning: independent versus cooperative agents. In: Proceedings of the Tenth International Conference on Machine Learning (ICML 1993), pp. 330–337. Morgan Kauffman, San Francisco (1993)
8. Whitehead, S.D.: A complexity analysis of cooperative mechanisms in reinforcement learning. In: Proceedings of AAAI 1991, Anaheim, CA, pp. 607–613 (1991)
9. Zhang, C., Lesser, V.: Coordinating multi-agent reinforcement learning with limited communication. In: Ito, J., Gini, S. (eds.) Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, St. Paul, MN, pp. 1101–1108 (2013)
10. Zhang, K., Maeda, Y., Takahashi, Y.: Group behavior learning in multi-agent systems based on social interaction among agents. SCIS & ISIS **12010**, 193–198 (2010)
11. Åström, K.: Optimal control of markov processes with incomplete state information. Journal of Mathematical Analysis and Applications **10**(1), 174–205 (1965)

Personal Peculiarity Classification of Flat Finishing Motion for Skill Training by Using Expanding Self-Organizing Maps

Masaru Teranishi, Shinpei Matsumoto, Nobuto Fujimoto and Hidetoshi Takeno

Abstract The paper proposes an unsupervised classification method for peculiarities of flat finishing motion with an iron file, measured by a 3D stylus. The classified personal peculiarities are used to correct learner's finishing motions effectively for skill training. In the case of such skill training, the number of classes of peculiarity is unknown. An expanding Self-Organizing Maps is effectively used to classify such unknown number of classes of peculiarity patterns.

Experimental results of the classification with measured data of an expert and sixteen learners show effectiveness of the proposed method.

Keywords Self-organizing maps · Unsupervised classification · Motion classificaiton · Technical education

1 Introduction

In the technical education of junior high schools in Japan, new educational tools and materials are in development, for the purpose to transfer many kinds of crafting technology. When a learner studies technical skills by using the educational tools, two practices are considered to be important: (1) to imitate motions of experts and (2) to notice their own "Peculiarity", and correct it by themselves.

However, present educational materials are not yet effective to assist the practices because most materials consist still or motion pictures of tool motions of experts. Even though the learners could read out rough outlines of the movements from these materials, it is difficult to imitate detailed motion due to less information these materials have. Especially, it is most difficult to imitate fine motions and postures of the tool by only looking the expert motions from a fixed viewpoint.

M. Teranishi(✉) · S. Matsumoto · N. Fujimoto · H. Takeno
Hiroshima Institute of Technology, Hiroshima, Japan
e-mail: {teranisi,s.matsumoto.gk,m161507,h.takeno.au}@cc.it-hiroshima.ac.jp

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_15

Furthermore, the learners could not recognize their own "peculiarity" as a difference between the expert's motion and the learner's motion from the materials. To solve the problem, a new assistant system for a brush coating skill has developed [1]. The system presents a learner corrective suggestion by play-backing the learner's motion by using animated 3D Graphics.

The paper describes the development of a new technical educational assistant system [2, 3, 4] that let learners acquire a flat finishing skill with an iron file. The system measures a flat finishing motion of a learner by a 3-D stylus device, and classifies the learner's "peculiarity". The system assists the learners how to correct bad peculiarities based on detected "peculiarity" classified from difference of the motions between the expert and the learner. The paper mainly describes the learner's peculiarities classification method which is implemented in the proposed system. An expanding Self-Organizing Maps(SOM) [5] is proposed to classify learners' "peculiarity" effectively.

2 Motion Measuring System for Flat Finishing Skill Training

Fig. 1(a) shows an outlook of the motion measuring devices of the system for flat finishing skill training. The system simulates a flat finishing task that flatten a top surface of an pillar object by an iron file. The system measures a 3D + time motion data of the file by using a 3D stylus. We use the PHANTOM Omni (SensAble Technologies) haptics device as the 3D stylus component of the system. The file motion is measured by attaching the grip of the file to the encoder stylus part of the haptics device. Assuming that the system will have a force feed back teaching function, we use a light weight mock file made of an acrylic plate which imitates a real 200 mm length iron file [2, 3].

In the measurement task, a learner operates the mock file in order to flatten the top surface of a dummy work whose area has 25 mm width and 25 mm depth, at 80 mm height. The system measures the motion of the mock file. The measured motion is recorded as a time series of the position of the file with X, Y, and Z coordinate values, and the posture with the tilt angles along the three axes with Tx, Ty, Tz as the radians. The spatial axes of the operational space and tilt angles of the file are assigned as shown in Fig. 1(b).

Fig. 2 shows an example of an expert's motion measured by the system. In these plots, the expert operates the file with reciprocating motion 4 times per 20 seconds. The main direction of the reciprocating motion is along X axis. The expert pushes the file to the direction that X reduces, and pull the file to the opposite direction. Since the file works only in the pushing motion, we focused the pushing motion in the classification task in the rest part of the paper.

To classify every file-pushing motion as the same dimensional vector by the SOM, we have to preprocess measured data by two steps: (1) clipping out each

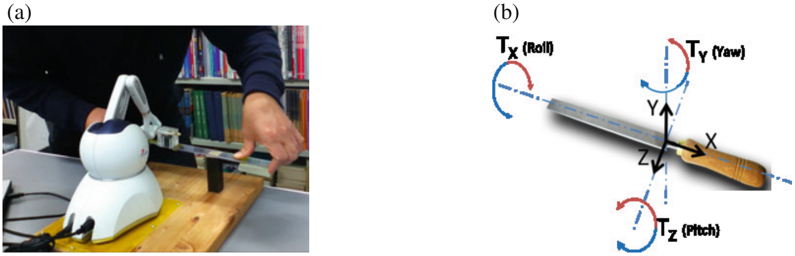


Fig. 1 Flat finishing skill measuring system: (a) outlook, (b) coordinates of measuring.

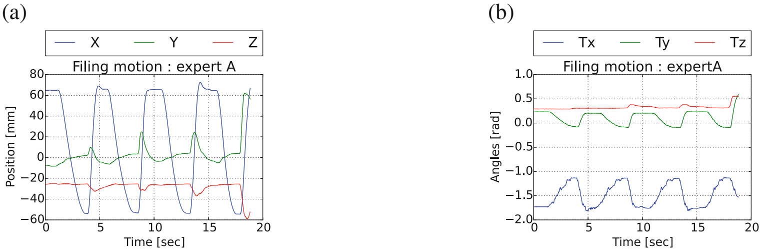


Fig. 2 Filing motion of an expert: (a) tool position, (b) posture of the tool.

time series potion of file-pushing motions, and (2) re-sampling the time series portions. **(1) Clipping:** The clipping is done according to the file-pushing motion range which is defined in X coordinates, beginning with X_{begin}, and ending with X_{end}. **(2) Re-sampling:** The resulted clipped time series of the pushing motions have different time lengths. So we could not use the series directly to the SOM because the dimensions of each series differ. Therefore, we should arrange dimensions of all series to the same number. Every series are re-sampled in order to have the same number of the sampling time points, by using the linear interpolation, as shown in Fig. 3(a). Since the stylus samples the motion enough fast, we can use the linear interpolation in the re-sampling with less lose of location precision. Fig. 3(b) shows the re-sampled result of the expert motion, Fig. 3(c) is that of a learner. In each plot, all four motions are imposed.

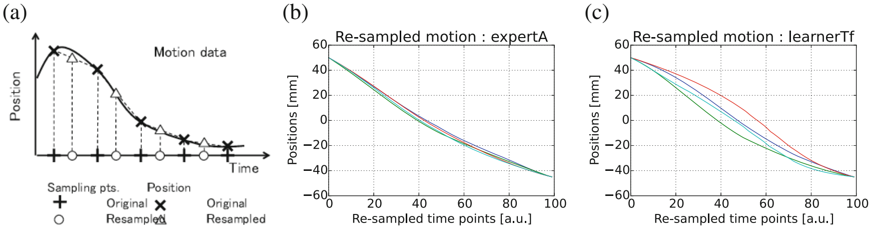


Fig. 3 (a) Re-sampling of a motion pattern, and re-sampled motion:(b) an expert, (c) a learner.

There is relevant differences between the expert's motions and the learner's motions. Every motion of the expert draws almost the same shaped curve, but the learner's don't. Each curves of learner's motion differs each time, in spite of identical person's actions. Additionally, the shape of curve of the expert means that the file is not moved in constant velocity. The expert operates the file by varying its velocity slightly. We call the slight varying velocity "velocity time series (VTS)", and consider it as the important factor of the expert motion.

The difference described above seems to be the main key of the correction in the skill teaching tasks. So we address a classification study of this difference as the starting point of the proposed system. In this paper, we use only the X coordinate values of the measured data for the classification to keep simplicity of data computation. The aim of the classification study is to form the SOM to organize such different motion curves in the map. After that, taking and using code-books of the learner's motion as the "bad peculiarities".

3 Feature Extraction of Filing Motion Based on Velocity Time Series

Velocity Time Series (VTS)

Although the filing motion obtained the former section have useful information about peculiarities, it is difficult to tell the learner their peculiarities exactly by only using the motion curve. Instead, to display the peculiarities on the basis of velocity time series (VTS) is an effective way. In this paper, we obtain a VTS of a re-sampled motion time series by computing local velocities at every re-sampled time point, by using 1st order differential approximation. Fig. 4(a) shows VTSs of an expert, and Fig. 4(b) shows VTSs of a learner. Since a VTS represents how velocities were taken at each time point of a filing motion, then learners could recognize and replay their motions more easily than looking raw motion plots. Additionally, learners also can imitate the expert's "model" motion. The average vector of an expert's VTSs is used as the "model" VTS.

Difference of Velocity Time Series (dVTS)

Although the "model" VTS of Fig. 4(a) is the model motion learners should imitate, a better way is to take difference of VTSs between the model and the learner for a learner to recognize and correct the peculiarities. For the purpose, we use difference of VTS (dVTS) as the feature pattern of the peculiarity. The dVTS is calculated by subtracting "model" VTS from the learners VTS.

Fig. 4(c) shows dVTSs of the expert, Fig. 4(d) shows that of the learner. The proposed system presents a learner dVTS. The dVTS tells the learner the peculiarity and the corrective points of the motion clearly. The dVTS displays a gap of VTSs between the model and the learner: if a learner imitates the model correctly, the dVTS draws a flat line, which means "Your skill is exact. No peculiarity", otherwise, for

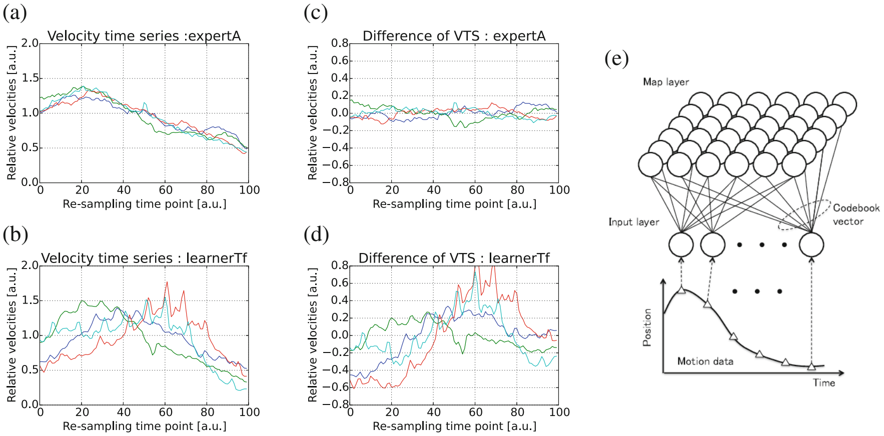


Fig. 4 Velocity time series (VTS) of (a) an expert, (b) a learner, difference of VTS (dVTS) of (c) the expert, (d) the learner, and (e) structure of SOM.

example, if the former part of the dVTS takes negative values, a trainer could point out the corrective point by telling the learner "You should move the file more fast in the former part of the motion".

4 Peculiarity Classification by Expanding SOM

We classify the file motion data by using the Self-Organizing Maps (SOM). Technical issues about the classification of the file motion are considered in three major points: (1)peculiarities are implicitly existing among the motion data, (2) the number of peculiarity variations, i.e., the number of classes is unknown and (3) every motions are not exactly same even though in the same learner, there are some fluctuation in each motion.

4.1 Structure of SOM

SOM is one of effective classification tools for such above patters whose number of classes is unknown and whose classification features are implicit. Fig. 4(e) shows the structure of the SOM. The SOM is a kind of neural networks which have two layers: one is the input layer, the other is the map layer. The input layer has n neuron units, where n is the dimension of input vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$. The map layer consists of neuron units, which is arranged in 2D shape. Every unit of the map layer has full connection to all units of the input layer. The ith map unit u_i^{map} has full connection vector $\boldsymbol{m}_i$ which is called "code-book vector". The SOM classifies the input pattern by choosing the "firing code-book" $\boldsymbol{m}_c$ which is the nearest to the input vector $\boldsymbol{x}$ by the distance defined as follows.

$$\|\boldsymbol{x} - \boldsymbol{m}_c\| = \min_i\{\|\boldsymbol{x} - \boldsymbol{m}_i\|\} \tag{1}$$

The SOM classifies the high dimensional input pattern vector according to the similarity to the code-book vectors. The map units also arranged in two dimensional grid like shape, and neighbor units have similar code-book vectors. Therefore, the SOM is able to "map" and visualize the distribution of high dimensional input patterns into a simple two dimensional map easily. Since the SOM also could form a classification of patterns automatically by using "Self-Organizing" process, we need not to consider the number of classes.

The SOM organizes a map by executing the following three steps onto every input pattern: (1) first, the SOM input a pattern, (2) then it finds a "firing unit" by applying Eq.(1) to every code-book vector $\boldsymbol{m}_i$, (3) and it modifies code-book vectors of the "firing unit" and its neighbors. In the step (3), code-book vectors are modified toward the input pattern vector. The amount of modification is computed by the following equations, according to a "neighbor function" h_{ci} which is defined based on a distance between each unit and the firing unit.

$$\boldsymbol{m}_i(t + 1) = \boldsymbol{m}_i(t) + h_{ci}(t)\{\boldsymbol{x}(t) - \boldsymbol{m}_i(t)\} \tag{2}$$

where t is the current and $t + 1$ is the next count of the modification iterations. The neighbor function h_{ci} is a function to limit modifications of code-book vectors to local map units which are neighbor the firing unit. The proposed method uses "Gaussian" type neighbor function. The Gaussian type modifies code-book vectors with varying amounts that decays like Gaussian function, proportional to the distance from the firing unit as follows:

$$h_{ci} = \alpha(t) \exp\left(-\frac{\|\boldsymbol{r}_c - \boldsymbol{r}_i\|}{2\sigma^2(t)}\right) \tag{3}$$

where $\alpha(t)$ is a relaxation coefficient of modification amount. The standard deviation of the Gaussian function is determined by $\sigma(t)$. We decrease both $\alpha(t)$ and $\sigma(t)$ monotonically as the modification iteration proceeds. The $\boldsymbol{r}_c$, $\boldsymbol{r}_i$ are the locations of the firing unit and the modified code-book vector unit, respectively. The reason why we use the Gaussian function is based on the assumption that the dVTSs distribute continuously in the feature space.

4.2 Expanding SOM

When we use the SOM straightforward, a "Map edge distortion" problem often occurs. The problem is observed as over-gathering of input data to code-books which are located edges of the map. Therefore, the classification performance become worth due to the each of the edge code-books represents more than one appropriate class of the input data. We have confirmed the problem in the early development of the system [3]. The problem is caused based on the fact that there is no code-book outside the edge, therefore the edge code-book can't be modified to appropriate direction in the feature space.

We solve the problem by relocating the edge code-books in order to expand the map in the feature space, artificially. The solution expects the edge distortion problem would be reduced by expanding the SOM after ordinal organizing process. First, we execute the usual self-organizing process for certain iterations. After that, we expand the map by relocating the edge code-books toward outside the span of the map.

In the expanding operation, the edge code-books $\boldsymbol{m}_e$s are modified by the following equation, in order to relocate its location in the corresponding feature space, against the center code-book $\boldsymbol{m}_o$ of the map.

$$\boldsymbol{m}'_e = \boldsymbol{m}_e + r_e\{\boldsymbol{m}_e - \boldsymbol{m}_o\} \tag{4}$$

where $r_e > 0$ denotes the ratio of expansion.

Second, we execute self-organizing process of the SOM again. Then the edge code-books of the first SOM result are virtually moved to inside the map, so the edge distortion would be reduced.

5 Classification Experiment

To evaluate of the effectiveness of the proposed method, we carried out experiments of the filing motion peculiarities classification.

The motion data were measured with an expert and sixteen learners. The expert operated four filing motions, and every learner operated three or four. Totally we used 66 motion data.Each motion data is clipped out as in range $[X_{begin}, X_{end}] = [50, -45](mm)$, and re-sampled at $n = 100$ points.

The SOM consists the input layer with 100 units and the map layer with 49 code-book vectors, which are arranged in 2D formation with 7 by 7, hexa-topology. The first self-organizing process started with initial relaxation coefficient $\alpha = 0.01$, initial extent of the neighbor function $\sigma = 7$, and iterated 10,000 times with the Gaussian type neighbor function.

After the first self-organizing process, each dVTS of an expert and learners are classified as shown in Fig. 5(a). In the Figures, labels beginning with letter 'S' are expert's dVTS, labels beginning with 'T' are learners'. The expert's dVTSs are located left edge of the feature map. The peculiarities are classified into four classes along the edge. In Fig. 5(a), the edge distortion problem occurred. Especially in the left bottom edge, more than ten dVTSs are gathered into a code-book. Fig. 5(b) shows all the dVTSs which are gathered to the left bottom code-book. The dVTSs have different curve types, therefore insufficient classification caused by the edge code-book. Our former work [3] had this problem.

After that, we expanded the map with 2.0 times larger. Then the second self-organizing process started with the expanded map and had 10, 000 iterations. The resulted map is shown in Fig. 5(c). Both over-gatherings at the left bottom and the left upper edges are reduced as shown in Fig. 5(c). The inappropriate gathering problem of the usual SOM shown as Fig. 5(b) was also improved by separating two classes, shown in Fig. 5(d) and (e).

From these results, the expanding SOM seems to work effectively to classify motion peculiarities of the learners, even for a little data set.

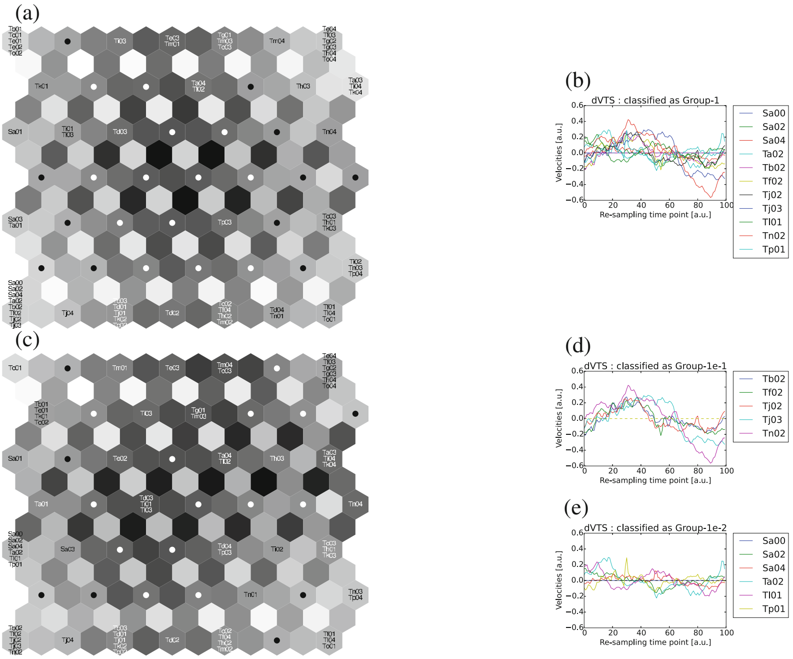


Fig. 5 (a,b)Result of usual SOM [3]: (a) feature map, (b) classified dVTSs at left bottom code-book. (c,d,e) result of the expanded SOM: (c) feature map, (d) classified dVTSs at left bottom code-book, (e) classified dVTSs at left center code-book.

6 Conclusion

The paper proposed a new motion classification method of individual learner's "peculiarities" as a part of the development of the new technical educational tool of flat finishing skill. An expanding Self-Organizing Maps(SOM) has proposed in order to classify the X coordinate motion dVTS curves into an expert's motions and “peculiarities” of learners. The experimental results with measured data of an expert and learners showed the effectiveness of the expanding SOM to cope with “edge distortion” problem of the SOM.

Consideration for evaluation methods of effectiveness of expanding SOM, more effective expansion methods for SOM, further classification evaluation is required with more number of learner person data are remained as future works.

References

1. Matsumoto, S., Fujimoto, N., Teranishi, M., Takeno, H., Tokuyasu, T.: A brush coating skill training system for manufacturing education at Japanese elementary and junior high schools. Artificial Life and Robotics **21**, 69–78 (2016)
2. Teranishi, M., Takeno, H., Matsumoto, S.: Classification of Personal Variation in Tool Motion for Flat Finishing Skill Training by Using Self-Organizing Maps. 2014 Ann. Conf. of Electronics, Information and Systems Society, I.E.E. of Japan, OS1‘1-8, 1144–1147 (2014)
3. Teranishi, M., Matsumoto, S., Takeno, H.: Classification of Personal Variation in Tool Motion for Flat Finishing Skill Training by Using Self-Organizing Maps. The 16th SICE System Integration Division Annual Conference, 3L2-1, 2655–2669 (2015)
4. Matsumoto, S., Teranishi, M., Takeno, H.: A Training Support System of Brush Coating Skill with Haptic Device for Technical Education at Primary and Secondary School. Intl. Sym. on Art. Life and Robotics (AROB 20th 2015), GS10–5 (2015)
5. Kohonen, T.: Self-Organizing Maps. Springer (2001)

Kinect and Episodic Reasoning for Human Action Recognition

Ruben Cantarero, Maria J. Santofimia, David Villa, Roberto Requena, Maria Campos, Francisco Florez-Revuelta, Jean-Christophe Nebel, Jesus Martinez-del-Rincon and Juan C. Lopez

Abstract This paper presents a method for rational behaviour recognition that combines vision-based pose estimation with knowledge modeling and reasoning. The proposed method consists of two stages. First, RGB-D images are used in the estimation of the body postures. Then, estimated actions are evaluated to verify that they make sense. This method requires rational behaviour to be exhibited. To comply with this requirement, this work proposes a rational RGB-D dataset with two types of sequences, some for training and some for testing. Preliminary results show the addition of knowledge modeling and reasoning leads to a significant increase of recognition accuracy when compared to a system based only on computer vision.

Keywords Action recognition · Common sense · Episodic reasoning · Artificial intelligence · Kinect · Computer vision

1 Introduction

Human action recognition has been a major concern for areas such as computer vision, robotics, machine learning, or ambient intelligence. Traditionally, vision methods for action recognition were mainly based on RGB images [16], sometimes enriched

R. Cantarero(✉) · M.J. Santofimia · D. Villa · R. Requena · M. Campos · J.C. Lopez
School of Computing Science, University of Castilla-La Mancha, Ciudad Real, Spain
e-mail: {ruben.cantarero,mariajose.santofimia,david.villa,juancarlos.lopez}@uclm.es,
{Roberto.Requena,Maria.Campos1}@alu.uclm.es

F. Florez-Revuelta · J.-C. Nebel
Digital Imaging Research Centre, Kingston University, London, UK
e-mail: {F.Florez,J.Nebel}@kingston.ac.uk

J. Martinez-del-Rincon
The Institute of Electronics, Communications and Information Technology (ECIT),
Queens University of Belfast, Belfast, UK
e-mail: j.martinez-del-rincon@qub.ac.uk

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_16

with body sensors [14]. However, with the availability of affordable devices such as Microsoft Kinect or ASUS Xtion Pro, depth information (D) has also came into play. This type of devices facilitates the extraction of 3D points of body joints, from which skeletal models can be constructed. Different methods have been employed to perform action recognition from RGB-D data [12]. These approaches are based on the use of depth maps [19] [20] [25], skeleton joints [23][24] or hybrid methods [22] [15] [13].

Despite accuracy improvement in comparison to RGB-based methods [26], more elaborated mechanisms are required to support action recognition. Inspired in how humans tackle this task, different sources of information have to be combined: sensory information (visual, acoustic, smell, etc.), common-sense and context knowledge. Rationality, knowledge, and senses, therefore prove essential for any attempt to replicate human ability for action recognition. This work caters for these three elements in a novel approaches that combines body-pose estimation and common-sense reasoning.

Sensory perception is here limited to vision. RGB-D images recorded from a Microsoft Kinect device are fed to a state-of-art body-pose estimation algorithm. Rationality is achieved by proposing a scenario that has been set up to promote actions aimed to an end. This will allow us to overcome the lack of rationality that most of non-real scenario datasets lack from. Finally, common-sense and context knowledge is here managed by means of an efficient knowledge-base system with reasoning capabilities. This work combines these three elements to propose a five-stage framework for action recognition.

The rest of the paper is organized as follows. First, the methodological approach proposed in this work is detailed in Section 2. Then, Section 3 evaluates the proposed methodology. Finally, Section 4 summarizes the conclusions drawn from this work.

2 Methodology

The method for human action recognition proposed here follows a five-stage approach, as depicted in Figure 1. First stage records data from an RGB-D device such as Microsoft Kinect. Then, these data are organized and made available in a public dataset. The 3D-points of body joints will afterwards go through a parsing process first and a normalization process later in order to homogenize different heights, angles, or perspectives. Once normalized, these data, organised as action files, are fed to a body-pose estimation algorithm known as Bag of Key Poses (BoKP) [3] in charge of identifying the action being performed. The BoKP method returns a list of actions ordered by probability, which are different possibilities of the actual action happening in the file. Finally, it is the role of the reasoning system, implemented in Scone [11], to determine whether the actions provided by the vision system need to be corrected based on knowledge premises. The following subsections provide thorough details of the aforementioned stages.

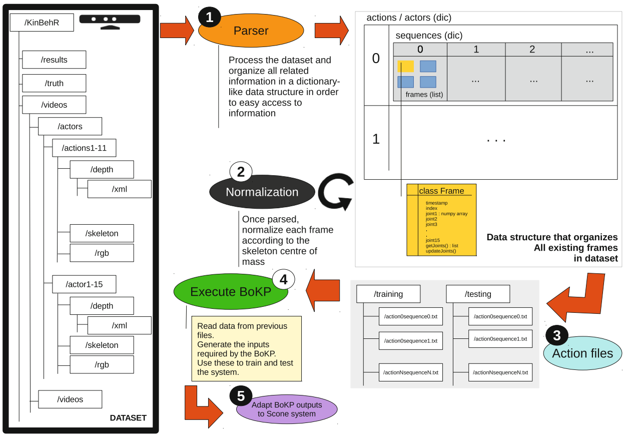


Fig. 1 System-stage overview

2.1 *The Dataset*

Rational behavior is implicit in human's daily life activities, since a person performs an action for a reason or aimed to an end [8]. Lab-recorded and synthetic datasets that involve actors performing isolated and unrelated actions [22] [23] are therefore not representative of real scenarios and not valid for the purpose of this work. Rationality is what entitles the reasoning system to understand an ongoing activity which also brings opportunities to correct the computer vision system estimation. Other more realistic datasets simply do not provide skeleton joint positions [26] [18].

It is the lack of a benchmarking dataset, with the appropriate configuration, what has motivated the recording of a new one. The KinBehR [2] dataset has been recorded using the second generation of Kinect for Windows and its SDK[1].

The recorded information can be divided into five categories: RGB video with a resolution 1920 × 1080 pixels per frame; 512 × 424 resolution depth video; 512 × 424 infrared video; background substracted videos that yield detected users; 560 × 420 video where the 25 joints of each identified and tracked user are represented. Additionally, one XML file is generated per frame. This XML file contains all the relevant information collected by Kinect and the Microsoft capturing algorithm: the position and orientation of the 25 joints of all tracked users, user state, user hand states, etc.

The resulting dataset has been split in two sets of sequences: training and testing. The generation of this dataset involved 18 actors, performing 13 types of actions: watch clock, crossing arms, scratching head, sit down, get up, turn around, walk, wave, punch, kick, point, take something, throw over head, throw down. These 13 actions have been selected to match those of the public dataset IXMAS [21]. In [2] all the information about this dataset is available.

[1] https://dev.windows.com/en-us/kinect

For each actor, two types of sequences were recorded. First, the actor was told what to do to ensure that all the considered actions were performed, and thus recorded. Second, one more sequence was recorded for every actor, in which the actor was not told what to do. In this second type, actors were given the sole direction of behaving normally. Finally, all sequences were manually labelled, indicating the start and the end of actions occurrences, since automatic segmentation of videos is not considered in this study.

In order to ensure rational behavior in the sequences where actors were freely behaving, the recording scenario was equiped with appropriate elements such as a punching ball, a small ball, a book, and a chair. With this scenario configuration, we were ensuring that performing an action such as kicking, was motivated by the fun obtained from interacting with a punching ball.

2.2 Bag of Key Poses

Among the different approaches for human action recognition, this work resorts to a machine learning technique, known as Bag of Key Poses (BoKP), described in [7] [4] and available in [6]. This method relies on a training phase during which salient features are learned. For this phase, training sequences were used where actors were being told what to do. Originally, the BoKP system worked with silhouette-based pose representation, where only the contour points of the silhouettes were used as features. An extension of that work led to consider 3D real-world coordinates for the points comprising the 25 joints of tracked users. Consequently, the number of points to be processed is dramatically reduced [5] while preserving the recognition accuracy by only considering the most representative points of the users in the scene, i.e. their joint points.

However, to ensure the user-characteristic independence, a normalization process has to be carried out. The normalization method employed in this work is described in [1]. During the training stage, skeleton sequences performing an action are provided to the BoKP algorithm. For the skeleton of each frame, the algorithm computes the most similar key pose. Then, at the end of the process, the frame sequence is represented by a sequence of key poses, labeled with the given action. During the testing stage, a similar process is carried out in order to compute the most representative key pose for the skeletons of the sequence. The learned action most similar to that sequence is the recognized one.

The algorithm proposed in [6] has been modified returning a ranking of five recognised actions according to their distance to the trained sequences, instead of returning the most similar one. The ranking might reveal if a test sequence is clearly matched with a particular action or it is similar to several actions. The reasoning system, as explained below, uses that information provided by the five most probable actions at every frame in order to make corrections.

2.3 Common-Sense Reasoning

Finally, the last system stage is intended to select one action, out of the five obtained from the previous stage, in order to complete the understanding of the ongoing activities.

The work in [9] demonstrates the assessment improvement obtained from combining common-sense reasoning with computer vision when trying to recognize sequence of actions that were performed for a reason [10]. In this sense, the representation of this type of sequences, as well as its support for reasoning, is well addressed by the Scone Knowledge-Base [11].

This work employs the knowledge and semantic model presented in [17]. Similarly to that work, rather than considering just one action, the five most probable ones are being considered at the same time. Based on this premise, the reasoning system needs to be capable of simultaneously tracking both the action considered as the most probable and the remaining four. In order to articulate this need, the employed architecture is thoroughly described in [17].

This architecture resorts to the notion of **Action** to represent each of the actions considered in the dataset. The notion of **Belief** encompasses a sequence of actions or episode. Each belief can only hold one action out of the five considered possible. **Expectations** group a set of actions. For example, the expectation used for *reading a book* considers the following sequence of actions: picking up, sitting down, and standing up. The appearance of any of these actions might suggest that an ongoing activity such as reading a book might be taking place. Finally, the **Estimation** concept is used to refer to the explanation standing from the recorded sequence.

These concepts along with the *knowledge about how the world works* and the considered context scenario are modeled and represented in Scone Knowledge-Base.

3 Validation and Preliminary Results

To validate our approach, the previously described KinhBer dataset is used. Figure 2 depicts the different modules involved and their interconnections.

At this stage, the CVS only implements the BoKP algorithm. The extraction of low-level information refers to information that can be potentially extracted from recognizing objects from video sequences in future extensions. The idea is to enhance the AIRS with low-level information regarding objects location in the scene, with regards to the actor. At this stage, only a reduced version of the DSK, WK, and BK is being considered for this prototype. Our recognition process could greatly benefit from such information, since for example, if a sitting down action is being considered but no chair is near the actor, this choice can therefore be rationally discarded.

Table 1 summarizes the recognition rate of the actions performed in each sequence obtained by the implemented prototype. A CVS-only system obtains an average accuracy of 31,9% when the first and most likely action of the BoKP algorithm is

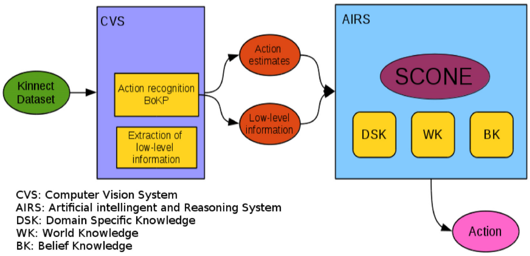

CVS
Kinnect Dataset
Action recognition BoKP
Extraction of low-level information
Action estimates
Low-level information
AIRS
SCONE
DSK
WK
BK
Action
CVS: Computer Vision System
AIRS: Artificial intellingent and Reasoning System
DSK: Domain Specific Knowledge
WK: World Knowledge
BK: Belief Knowledge

Fig. 2 System overview

Table 1 Preliminary results obtained by the prototype implementation

	CVS-only	AIRS
Actor 1	33.33%	33.33%
Actor 2	66.66%	66.66%
Actor 3	42.85%	52.38%
Actor 4	13.33%	23.33%
Actor 5	16.66%	66.66%
Actor 6	11.11%	11.11%
Actor 7	36.84%	42.10%
Actor 8	52.94%	58.82%
Actor 9	18.75%	18.75%
Actor 10	30.76%	30.76%
Actor 11	25.00%	50.00%
Actor 12	33.33%	43.33%
Actor 13	42.85%	50.00%
Actor 14	22.22%	27.77%
Average	31.90%	41.07%

used as the estimated action to compute the accuracy. The AIRS obtains an accuracy of 41,07% after having considered the five most probable actions in the reasoning to select the most likely action as the final estimation. Here, accuracy is measured in terms of the numbers of actions correctly estimated.

This first prototype only considers a reduced version of DSK. When the considered knowledge is relevant for any of the sequences, like in Actor 5 or 11, an accuracy improvement is observed. However, actors were not given behavioral instructions and DSK was neither tailored for the recorded sequences. Results therefore show that, when the encoded knowledge responds to recorded behaviour, the recognition rate is improved. Additional efforts need to be dedicated to model DSK, WK, and BK.

4 Conclusions

This paper presents a novel methodology for human action recognition that combines RGB-D data, captured from a Microsoft Kinect device, and a common-sense reasoning system. In order to validate the proposed methodology a dataset recording rational behaviour has been constructed and released. A prototype of the system shows encouraging results since, with a reduced version of the required knowledge, the system has improved the average accuracy obtained by the computer vision system.

As future work, the system will be extended to consider information about the objects as well as additional domain specific knowledge. This additional information will support, for example, corrections based on if no object is nearby the actor, a *picking up* does not make sense.

Acknowledgements This work has been partly funded by the Spanish Ministry of Economy and Competitiveness under project REBECCA (TEC2014-58036-C4-1-R) and by the Regional Government of Castilla-La Mancha under project SAND (PEII_2014_046_P).

References

1. Chaaraoui, A.A., Padilla-Lopez, J.R., Florez-Revuelta, F.: Fusion of skeletal and silhouette-based features for human action recognition with rgb-d devices. In: The IEEE International Conference on Computer Vision (ICCV) Workshops, December 2013
2. Cantarero, R., Santofimia, M.J., Nebel, J.-C., Revuelta, F.F., del Rincon, J.M., Lopez, J.C.: KinBehR: Kinect for Behaviour Recognition. http://arco.esi.uclm.es/public/prj/KinbehR/KinbehrDataset_v2.zip
3. Chaaraoui, A.A., Climent-Pérez, P., Flórez-Revuelta, F.: An efficient approach for multi-view human action recognition based on bag-of-key-poses. In: Proceedings of the Third International Conference on Human Behavior Understanding, HBU 2012, pp. 29–40. Springer, Heidelberg (2012)
4. Chaaraoui, A.A., Climent-Pérez, P., Flórez-Revuelta, F.: Silhouette-based human action recognition using sequences of key poses. Pattern Recognition Letters **34**(15), 1799–1807 (2013)
5. Chaaraoui, A.A., Florez-Revuelta, F.: Adaptive human action recognition with an evolving bag of key poses. IEEE Trans. on Auton. Ment. Dev. **6**(2), 139–152 (2014)
6. Chaaraoui, A.A., Flórez-Revuelta, F., Climent-Pérez, P.: Bag of key poses (2015). https://github.com/DAIGroup/BagOfKeyPoses
7. Climent-Pérez, P., Chaaraoui, A.A., Padilla-López, J.R., Flórez-Revuelta, F.: Optimal joint selection for skeletal data from RGB-D devices using a genetic algorithm. In: Advances in Computational Intelligence - 11th Mexican International Conference on Artificial Intelligence, MICAI 2012, Revised Selected Papers, Part II, San Luis Potosí, Mexico, October 27 - November 4, pp. 163–174 (2012)
8. Davidson, D.: Actions, reasons, and causes. The Journal of Philosophy **60**(23), 685–700 (1963)
9. del Rincón, J.M., Santofimia, M.J., Nebel, J.-C.: Common-sense reasoning for human action recognition. Pattern Recogn. Lett. **34**(15), 1849–1860 (2013)
10. Edwards, C.: Decoding the language of human movement. Commun. ACM **57**(12), 12–14 (2014)
11. Fahlman, S.E. : The Scone knowledge-base project (2016). http://www.cs.cmu.edu/~sef/scone/ (retrieved on January 20, 2016)
12. Farooq, A., Won, C.S.: A survey of human action recognition approaches that use an rgb-d sensor. IEIE Transactions on Smart Processing and Computing **4**(4), 281–290 (2015)
13. Gonen, M., Alpaydin, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
14. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Communications Surveys Tutorials **15**(3), 1192–1209 (2013)

15. Lei, J., Ren, X., Fox, D.: Fine-grained kitchen activity recognition using rgb-d. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp 2012, pp. 208–211. ACM New York (2012)
16. Mojidra, H.S., Borisagar, V.H.: A literature survey on human activity recognition via hidden markov model. In: IJCA Proceedings on International Conference on Recent Trends in Information Technology and Computer Science 2012, ICRTITCS vol. (6), pp. 1–5, February 2013
17. Santofimia, M.J., Martinez-del Rincon, J., Nebel, J.-C.: Episodic reasoning for vision-based human action recognition. The Scientific World Journal (2014)
18. Shimada, A., Kondo, K., Deguchi, D., Morin, G., Stern, H.: Kitchen scene context based gesture recognition: A contest in ICPR2012. In: Advances in Depth Image Analysis and Applications - International Workshop, WDIA 2012, Revised Selected and Invited Papers, Tsukuba, Japan, November 11, pp. 168–185 (2012)
19. Vieira, A., Nascimento, E., Oliveira, G., Liu, Z., Campos, M.: Stop: Space-time occupancy patterns for 3d action recognition from depth. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. LNCS, vol. 7441, pp. 252–259. Springer, Heidelberg (2012)
20. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7573, pp. 872–885. Springer, Heidelberg (2012)
21. Weinland, D., Özuysal, M., Fua, P.: Making action recognition robust to occlusions and viewpoint changes. In: European Conference on Computer Vision (2010)
22. Wu, Y.: Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, pp. 1290–1297. IEEE Computer Society, Washington, DC (2012)
23. Xia, L., Chen, C.-C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3d joints. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, June 16-21, pp. 20–27. IEEE (2012)
24. Yang, X., Tian, Y.: Effective 3d action recognition using eigenjoints. J. Vis. Comun. Image Represent. **25**(1), 2–11 (2014)
25. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM International Conference on Multimedia, MM 2012, pp. 1057–1060. ACM, New York (2012)
26. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE Multimedia **19**(2), 4–10 (2012)

Ontology-Based Platform for Conceptual Guided Dataset Analysis

Miguel Ángel Rodríguez-García, José Medina-Moreira, Katty Lagos-Ortiz, Harry Luna-Aveiga, Francisco García-Sánchez and Rafael Valencia-García

Abstract Nowadays organizations should handle a huge amount of both internal and external data from structured, semi-structured, and unstructured sources. This constitutes a major challenge (and also an opportunity) to current Business Intelligence solutions. The complexity and effort required to analyse such plethora of data implies considerable execution times. Besides, the large number of data analysis methods and techniques impede domain experts (laymen from an IT-assisted analytics perspective) to fully exploit their potential, while technology experts lack the business background to get the proper questions. In this work, we present a semantically-boosted platform for assisting layman users in (i) extracting a relevant subdataset from all the data, and (ii) selecting the data analysis technique(s) best suited for scrutinising that subdataset. The outcome is getting better answers in significantly less time. The platform has been evaluated in the music domain with promising results.

Keywords Ontologies · Business intelligence · Semantic web · Knowledge-based systems

M.Á. Rodríguez-García
Computational Bioscience Research Center, King Abdullah University of Science and Technology, 4700 KAUST, P.O. Box 2882, Thuwal 23955-6900, Kingdom of Saudi Arabia
e-mail: miguel.rodriguezgarcia@kaust.edu.sa

J. Medina-Moreira · H. Luna-Aveiga
Universidad de Guayaquil, Cdla. Universitaria Salvador Allende, Guayaquil, Ecuador
e-mail: {Jose.medinamo,harry.lunaa}@ug.edu.ec

K. Lagos-Ortiz
Universidad Agraria del Ecuador, Avenida 25 de julio, Guayaquil, Ecuador
e-mail: Klagos@uagraria.edu.ec

F. García-Sánchez · R. Valencia-García(✉)
Departamento de Informática y Sistemas, Universidad de Murcia, 30100 Murcia, Spain
e-mail: {frgarcia,valencia}@um.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_17

1 Introduction

The decision making process in organization is usually carried out in an environment of great uncertainty, mainly due to the lack of business domain information that justifies the decisions made. Uncertainty, which is tightly related to the scarcity of relevant data, is a source of risks. In line with this, there is a widespread understanding about the transformation process from data to information, and from information to knowledge. Thus, it would be desirable for organizations to have as much information as possible about the potential alternatives during the decision analysis to minimize the risk. Currently, organizations generate (and have access to) a huge amount of both structured (e.g. in the form of the content of databases) and unstructured (e.g. in the form of documents or text files) data. According to the study presented by [1], the digital universe will grow by a factor of 300, from 130 exabytes to 40,000 exabytes, or 40 trillion gigabytes from 2005 to 2020. In most cases, organizations cannot cope with the vast amount of data gathered in order to exploit it for creating competitive advantages [2].

Many methods and techniques have been conceived that make use of data analysis algorithms to generate knowledge from large datasets. The ultimate goal is to leverage such tools to effectively manage the available information and turn it into actionable insights. In this scenario, it is very important to place the datasets into context in order to generate useful knowledge and reduce the level of uncertainty in decision making processes. Certainly, for Business Intelligence solutions to work properly it is necessary to first define some business objectives and indicators before the data processing stage could be undertaken. Current tools in this area fail to provide services that guide the selection of the data related to those objectives and the data mining techniques that should be applied accordingly. Furthermore, in the state-of-the-art frameworks the data selection, filtering and interpretation processes require much time and effort.

In this work we propose a semantically-enhanced approach to overcome the aforementioned limitations in the Business Intelligence data analysis. The novel system presented here makes use of knowledge representation and reasoning technologies in order to facilitate different stages in data analysis in a particular domain. More concretely, our system simplifies the data selection phase by using ontologies containing shared knowledge about the application domain under question, what generates a better understanding by experts. Formally, an ontology can be defined as an explicit specification of a conceptualization, that is, it represents the knowledge as a set of concepts and a set of relationships between those concepts. Ontologies can be seen as conceptual models whose vocabulary is closer to both experts and managers. Ontologies are commonly used in artificial intelligence and the Semantic Web for knowledge representation. Currently, ontologies are being applied to different domains such as biomedicine [3; 4], innovation management [5], cloud computing [6, 7] and business intelligence [8].

The paper is structured as follows. The architecture of the platform developed in this work is described in Section 2. Section 3 includes the evaluation of the proposed approach through a use case in the music domain. Finally, conclusions and future work are put forward in Section 4.

2 Platform Architecture

The proposed system architecture (see Fig. 1) is comprised of three different modules, namely, the conceptual mapper module (1), the conceptual selection and enrichment module (2), and the "data processing techniques"-recommendation module (3). The former is responsible for finding a mapping between the domain ontology and the different data sources. The second is in charge of assisting users in selecting the appropriate data to create the subdataset that will be analysed. The latter module guides users in the selection of the specific algorithm or algorithms best suited for the data analysis of that subdataset.

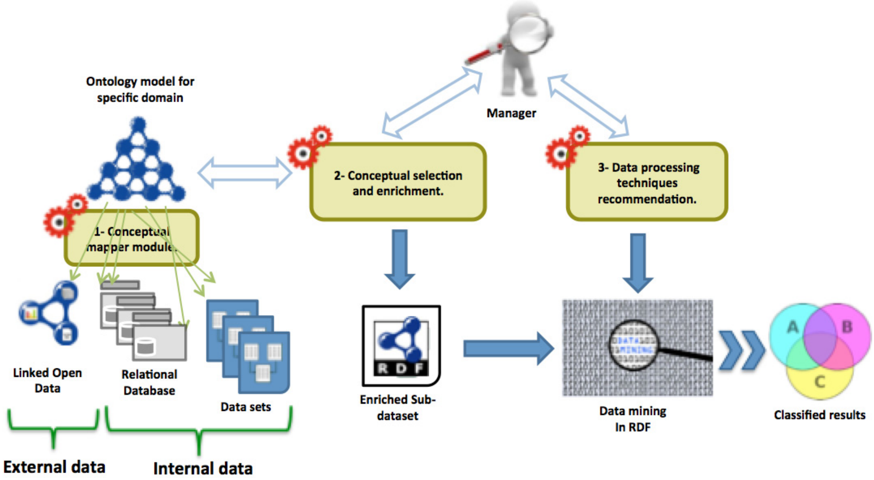


Fig. 1 Platform architecture.

These three modules are explained in detail in the following sections.

2.1 *Conceptual Mapper Module*

The aim of this module is to provide a way to perform a mapping between a given domain ontology and the different types of data sources available. As shown in Fig. 1, Linked Open Data repositories and relational databases are some of the data sources supported by this module. This innovative feature is based on a mapping configuration file containing a list of specific rules in XML. Each rule in the file defines the way in which the information contained in a data source is mapped to the domain ontology.

In essence, the mapping configuration file defines (i) the ontology location, (ii) the data sources where the information is extracted from, and (iii) each mapping between the elements in the data sources and components of the domain ontology. These mappings are based on the ones designed within the D2RQ platform[1], a system for accessing relational databases as virtual, read-only RDF graphs. In order to provide an abstract solution adaptable to different databases, the D2RQ platform makes use of the D2RQ mapping language, a declarative language for mapping relational database schemas to RDF vocabularies and OWL ontologies.

2.2 Conceptual Selection and Enrichment Module

This module assists users in selecting the data from all the datasets mapped by the conceptual mapper module to create a particular subdataset that is going to be further scrutinised. For this, it is necessary an ontology modelling the business domain. The module reads the ontology and displays in the user interface a conceptual abstraction of the domain under question, so that users can choose the concepts, attributes and relationships of the domain that they want the platform to analyse. The selection of the elements in the ontology has to meet these three conditions:

- Users must select at least one class.
- At least one attribute for each selected class has to be selected.
- Each selected class must have at least one relationship with another selected class.

When users choose the features and concepts of the dataset which they aim to make the analysis on, a system function allows them to set restrictions on the values that they want to process. This functionality is not usually present in current state-of-the-art tools but it is our understanding that it might be of great interest for users since it allows to further restrict the area on which they would like to gather more specific information. Once the classes, attributes and relationships have been selected, the module builds the subdataset that can be scrutinised using data mining methods for knowledge acquisition. The generation of this subdataset is output format-independent.

2.3 Data Processing Techniques Recommendation Module

The purpose of this module is to guide users in choosing the most convenient algorithm (or algorithms) for the analysis of the subdataset among those available. The module offers a simple interface showing the list of techniques that can be applied over each subdataset according to its characteristics. A large number of data mining techniques have been included in the platform such as the Apriori

[1] http://d2rq.org/

algorithm, K-Means clustering, EM algorithm, J48 algorithm, or Sequential Minimal optimization (SMO). In particular, our platform takes advantage of the algorithms already available through the Weka 3 library[2]. The results of the data mining algorithm executed can be both saved in a flat text file and shown through the user interface. For validation purposes in the next section the results obtained with the J48 algorithm and SMO are considered.

Many data analysis techniques, such as the Apriori algorithm, require the input data to be discretized, that is, the values each attribute may take have to belong to some specified interval. In order to fulfil such a prerequisite, the "discretize" method has been included in the module.

3 Use Case. The Million Song Dataset

The approach presented in this work has been validated in the music domain. In order to study the effect of the approach in this domain, a dataset related to the music domain had to be chosen. In particular, the MILLION SONG DATASET (MSD)[3] was selected. MSD is a free, public, 273 GB dataset containing a number of features (e.g., title, duration, name of the band, location of the artist and so on) and metadata from a vast amount of the most popular current songs. One of the main reasons for selecting this huge dataset was that it provides a large quantity of numeric features that can be analysed by employing several of the data mining techniques supported in our platform.

MSD is comprised of a set of HDF5-formmated files (HDF stands for "Hierarchical Data Format") where each file contains information about a song and the artist. Since the information provided by this dataset about each artist is insufficient, more details about them were incorporated into a MySQL database through the "musicbrainz.org" REST service[4]. The total size of our test dataset (including both the HDF5 files and the MySQL database) is 9 GB.

Besides, an ontology modelling the music domain has to be used for the purposes of the evaluation in this use case scenario. The Music Ontology [9] provides the main concepts and properties for describing the semantic concepts related to the music domain, such as releases, lives events, albums, artists and tracks. However, most of the information stored in the dataset is not modelled in this ontology, so it was necessary to adapt the music ontology by including new elements to fully model the dataset. The ontology contains five main classes as follows:

- Artist. This class is used to represent a singer or a band in the domain.
- Location. It represents a location. Some of its attributes are continent, country, latitude and longitude.

[2] http://www.cs.waikato.ac.nz/ml/weka/
[3] http://labrosa.ee.columbia.edu/millionsong/
[4] https://musicbrainz.org/doc/Development/XML_Web_Service/Version_2

- Release. This concept is used to represent an album comprised of different songs.
- Track. This class is used to represent a track or a song that is part of an album.
- Analysis. This class constitutes the root class of the hierarchy of analyses that could be applied in this scenario. Each analysis is represented as a subclass of the "Analysis" concept and includes "BeatAnalysis", "SectionAnalysis", "SegmentsAnalysis", "SegmentsPitchesAnalysis" and so on.

To carry out the evaluation, first the domain ontology and the mapping file were developed. Then, ten different experiments were conducted selecting various concepts to prove or obtain alternative hypothesis. The processing time for generating a subdataset takes approximately 1 hour. The data analysis methods executed over these subdatasets do not take longer than five minutes. Furthermore, the information gained (from the results of executing these algorithms over the subdatasets) is reduced but more valuable and more relevant to the domain. Next, two of the specific experiments conducted are described in more detail.

The first experiment is about guessing the sex of an artist from an analysis of the songs he or she sings. For this, the user selected the “SegmentsPitchesAnalysis” class from the “Analysis” class hierarchy. Then, the user selected two classification algorithms, namely, the J48 algorithm and SMO. The Precision, Recall and F-Measure scores of the results that each algorithm obtained are shown in Table 1.

Table 1 Precision (P), Recall (R) and F-Measure (F) obtained from the first experiment.

	Data	**Entire dataset**				**Subdataset**			
	Gender	**P**	**R**	**F**	**Time**	**P**	**R**	**F**	**Time**
J48	Male	80.1	91.4	85.3	5,3 h	78.9	90.9	84.5	5:16 m
	Female	36.3	17.4	23.1		35.4	16.9	22.9	
	Average	69	74.1	74.5		69	74.1	70.5	
SMO	Male	77.3	1	87.6	5,3 h	77.3	1	87.2	4:59 m
	Female	0	0	0		0	0	0	
	Average	59.8	77.3	67.4		59.8	77.3	67.4	

First, it is important to note the barely noticeabledifference between the results obtained from the whole datasets and subdatasets. However, the impact on the execution time is dramatic. While processing the whole dataset took 5,3 hour, the time spent using the subdataset is close to 5 minutes. Tests were made in a Core i7 CPU at 3.50GHz and 8GB of RAM.

The results obtained for classifying male singers are considerably better than the ones got for the female singer classification. Particularly, the F-Measure for male gender classification is 84.5/87.2 using the subdatasets, and 85.3/87.6 using the whole dataset, applying the J48/SMO algorithm, respectively. Therefore, it is possible to conclude that the selected algorithms are able to very accurately determine if a male singer plays a particular song from just the song analysis. However, for female gender classification the results obtained are significantly worse.

Concretely, with the SMO algorithm Precision, Recall, as well as the F-Measure got a score of 0, meaning that for this specific experiment the SMO algorithm is not capable of analysing and learning how to classify the information provided. The results for the J48 algorithm are better but they do not reach the good scores obtained for the male gender classification. In general terms, the average Precision, Recall and F-Measure achieved by the J48 algorithm is higher than those achieved using the SMO algorithm.

The second experiment focused on predicting the duration of a song given its analysis. For this, first the range of possible duration of the songs was divided into 10 different groups using the "discretize" function provided by Weka. For this experiment, the user selected the "BeatAnalysis" class from the "Analysis" class hierarchy. Then, both the J48 and SMO algorithms were executed in in order to verify whether these classification algorithms are capable to categorise the songs into their correct duration group taking into account other features and attributes. Table 2 collects the results obtained for this experiment in terms of the Precision, Recall and F-Measure scores. Since the results for the entire dataset are close to those obtained for the subdataset, only the latter are shown. While the time spent in processing the whole dataset was over 6 hours, the processing time of the subdataset was close to 6 minutes.

As shown in Table 2, the results of this experiment prove that there exists a correlation between the song analysis and its duration. On average, the SMO algorithm achieved the best results with an average F-Measure score of 97.6.

Table 2 Precision (P), Recall (R) and F-Measure (F) collected from the second experiment for J48 and SMO algorithms in subdataset.

Duration range (seconds)	**J48**			**SMO**		
	P	**R**	**F**	**P**	**R**	**F**
(0-138.435465]	98.7	98.5	98.6	99.9	1	99.9
(138.435465-166.85669]	93.9	94.2	94	97.3	98.1	97.7
(166.85669-187.25832]	89.6	89.5	89.6	95	95.7	95.3
(187.25832-205.857505]	87.8	89.5	88.6	96.6	95.5	96
(205.857505-223.568525]	88	86.3	87.1	96.4	95.6	96
(223.568525-240.78322]	88	88	88	95	95.7	95.3
(240.78322-263.013425]	89.7	89.7	89.7	96.9	96.3	96.6
(263.013425-291.77424]	91.6	92.1	91.8	98.7	98.9	98.8
(291.77424-351.3073]	95.7	95.7	95.7	99.8	99.9	99.8
(351.3073-inf)	99	98.5	98.8	1	1	1
Average	92.2	92.2	**92.2**	97.6	97.6	**97.6**

4 Conclusions and Future Work

In this paper, a new knowledge-based business intelligence platform is presented. Its aim is to take advantage of semantic technologies to offer an enhanced, more precise and more efficient data analysis. It uses knowledge representation for building a semantic model that helps to restrict the type of data to be used during the business intelligence analysis. Concretely, this platform is divided into three different modules, "conceptual mapper", "conceptual selection and enrichment", and "data processing techniques recommendation". The platform allows users to select a subset of the whole collection of data available, so data analysis can be more efficiently performed.

The approach has been validated in the music domain. For this, the Million Song Dataset was used and the music ontology was adapted to work with this dataset. With the evaluation performed in this work, it can be concluded that the results obtained with the subdatasets can provide more relevant information in less time, than the ones obtained through the analysis of the whole dataset. Consequently, our approach improves, in general, current state-of-the-art tools for data analysis.

This system can be improved in a number of ways. First, since it should be manually performed, building the mapping configuration file is very time-consuming. We aim to develop a usable, semi-supervised mechanism for specifying the mapping between the domain ontology and the data sources. Second, in its current form the platform does not carry out pre-processing tasks such as class balance pre-processing. We plan to include it along with other specific business intelligence methods to improve the performance in the dataset exploration. On the other hand, the RDF-formatted subdataset obtained will be enriched using reasoning techniques. Besides, we plan to develop ontologies to support decision making at various choice points of the data mining process as suggested in [10]. Meta-learning algorithms will be also included in future releases. Furthermore, we plan to evaluate our platform with different large data sources and linked data repositories. Finally, a new approach for domain ontology-based filtering and displaying the results in a conceptual manner will be developed taking into account some of the ideas and methods proposed in [11].

References

1. Gantz, J., Reinsel, D.: The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. In: McAfee, A., Brynjolfsson, E. (eds.) IDC iView: IDC Analyze the Future, vol. 2007, pp. 1–16 (2012)
2. McAfee, A., Brynjolfsson, E.: Big Data: The Management Revolution. Harvard Business Review **90**(10), 60–68 (2012)
3. Ruiz-Martínez, J.M., Valencia-García, R., Martínez-Béjar, R., Hoffmann, A.G.: BioOnto-Verb: A top level ontology based framework to populate biomedical ontologies from texts. Knowl.-Based Syst. **36**, 68–80 (2012)
4. Rodríguez-González, A., Alor-Hernández, G.: An approach for solving multi-level diagnosis in high sensitivity medical diagnosis systems through the application of semantic technologies. Computers in Biology and Medicine **43**(1), 51–62 (2013)

5. Hernández-González, Y., García-Moreno, C., Rodríguez-García, M.Á., Valencia-García, R., García-Sánchez, F.: A semantic-based platform for R&D project funding management. Comput. Ind. **65**(5), 850–861 (2014)
6. Rodríguez-García, M.Á., Valencia-García, R., García-Sánchez, F., Samper-Zapater, J.J.: Ontology-based annotation and retrieval of services in the cloud. Knowl.-Based Syst. **56**, 15–25 (2014)
7. Rodríguez-García, M.Á., Valencia-García, R., García-Sánchez, F., Samper-Zapater, J.J.: Creating a semantically-enhanced cloud services environment through ontology evolution. Future Gener. Comput. Syst. **32**, 295–306 (2014)
8. Cheng, H., Lu, Y.C., Sheu, C.: An ontology-based business intelligence application in a financial knowledge management system. Expert Systems with Applications **36**(2), 3614–3622 (2009)
9. Raimond, Y., Abdallah, S.A., Sandler, M.B., Giasson, F.: The music ontology. In: ISMIR 2007, pp. 417–422 (2007)
10. Keet, C.M., Ławrynowicz, A., d'Amato, C., Kalousis, A., Nguyen, P., Palma, R., Stevens, R., Hilario, M.: The data mining OPtimization ontology. Web Semantics: Science, Services and Agents on the World Wide Web **32**, 43–53 (2009)
11. Jareevongpiboon, W., Janecek, P.: Ontological approach to enhance results of business process mining and analysis. Business Process Management Journal **19**(3), 459–476 (2013)

Intragroup Density Predicting Intergroup Tie Strength Within Open-Source-Software Collaboration Network

Stefan Kambiz Behfar and Qumars Behfar

Abstract There have been many studies in the literature discussing intra- and inter-cluster ties within sociological systems denoted by strong and weak ties, social and biological systems represented by community structure, and organizational systems signified by strong and loose couplings; where inter-cluster ties are mostly considered weak or loose. Loose couplings lead to dissemination or retrieval of information, spread of viruses, new product adoption, more and their prediction is significant for knowledge management, organizational innovation, epidemics of contagious diseases, and viral product design. Therefore in this paper, we investigate how to predict inter-cluster tie strength, and propose that inter-cluster tie strength can be predicted from determination of intra-cluster density. In the model design section, we provide the hypothesis and discuss logical and analytical reasoning; in the empirical section, we alternatively examine the relationship between intra-group density and inter-group tie strength via examining open-source-software (OSS) project collaboration data collected from SourceForge repository.

Keywords Inter-cluster tie strength · Intra-cluster density · Loose cluster coupling · Open-source-software project collaboration network

1 Introduction

Granovetter [1] proposed a network theory for tying micro and macro levels of sociological theory through an analysis of various types of weak ties bridging

S.K. Behfar(✉)
International Business Department, HEC Montreal, Montréal H3T 2A7, Canada
e-mail: kambiz.behfar@gmail.com

Q. Behfar
Neurology Department, Cologne Univeristy, NRW 50924, Cologne, Germany
e-mail: q_behfar@yahoo.com

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_18

groups otherwise connected by strong ties. Strong ties are relationships with individuals whom we know very well, but weak ties provide bridges over which innovations cross over boundaries of social groups (clusters), which themselves are strongly tied. Weick [3] introduced the concept of loose coupling, then organized a wide-ranging literature on loose coupling, and refocused research on loose coupling toward more difficult and useful interpretations. In continuation with Granovetter definition of weak-strong ties and Weick definition of loose coupling, Newman [2] defined a new property of sociological and biological networks as "community structure" where nodes are joined together in tightly-knit groups which are loosely connected to each other. Within organizational system, Hansen [4] has thoroughly discussed advantages and disadvantages of weak ties versus strong ties in terms of information redundancy and project time competition.

Knowledge management among open-source-software (OSS) developers has been already investigated (Joshi and Sarker [8]). Joshi et al. discussed factors influencing knowledge transfer within development teams. Ojha [10] also discussed knowledge sharing between team members based on similarity-attraction paradigm; where he proposed that knowledge sharing more likely happen between same demographic team members. Other scientists also discussed how OSS communities are formed such that Hahn et al. [11] investigated the personal factors causing a new developer to join a project such as prior collaboration between the new developer and the project initiator or previous experience of the group members. Hinds [12] discussed costs and benefits of community ties, and concluded that "social network structure of open source software has no important effect on community structure". On the other hand, Antwerp and Madey [13] investigated social network structure of open source software, and used long term popularity as the metric developer-developer tie and concluded that "previous ties are generally an indicator of past success and usually lead to future success".

In this study, we investigate how to predict inter-cluster tie strength, and propose that inter-cluster tie strength can be predicted from determination of intra-cluster density. We use both analytical and empirical analysis to prove our hypotheses. In the model design section, we provide hypothesis and discuss logical and analytical reasoning to prove it; whereas in the empirical analysis section, we use a complex network of open source software (OSS) project collaboration as our domain of interest; we examine the relationship between intra-group density and inter-group tie strength via examining OSS data collected from SourceForge repository.

2 Model Design

Initial tie is generated when two clusters are indirectly connected via a third party, or completely in random. All the ties formed after direct cluster inter-connection is called subsequent ties. Loose couplings lead to dissemination or retrieval of information, spread of viruses, new product adoption, and their prediction is significant for knowledge management (Hansen [4]), organizational innovation

(Ruef [5]), epidemics of contagious diseases (Shu et al. [6]), and viral product design (Aral and Walker [7], behfar et al. [8]). On the other hand, strong inter-cluster couplings lead to network cluster growth and performance, and their prediction is of great significance to make right marketing map and development strategy. We often encounter with the question of how to predict inter-cluster tie and its strength. Therefore, we propose that *inter-cluster tie strength can be predicted from determination of intra-cluster density.*

In order to predict inter-cluster tie strength, we used the method of tie generation based on the proposed tie probability model by Huang [14], and provided the probability functions based on the proposed clustering model. Tie generation mechanism, based on the proposed tie probability model by Huang (2006), for a cycle formation of degree k is given by g(k) denoted by c_k ; where g(1) corresponds to random tie formation mechanism. For example in the below figure, the occurrence probability of tie (P_1^1, P_4^1) is determined by three mechanisms of random tie generation g(1), length-2 formation mechanism g(2) and length-3 formation mechanism g(3). Considering that there is either probability of path remains path, or path becomes cycle, therefore either $Pr(P_1^1, P_4^1) = c_k$ or $Pr(i,j) = c_k^m/[c_k^m + (1-c_k)^m]$, where m indicates number of k-length paths, expected to become a cycle. We should now combine multiple mechanisms, e.g. g(1), g(2) and g(3) as:

$$Pr(i,j) = c_1^{m_1} c_2^{m_2} \dots c_k^{m_k} \\ /\left[c_1^{m_1} c_2^{m_2} \dots c_k^{m_k} + (1-c_1)^{m_1}(1-c_2)^{m_1} \dots (1-c_k)^{m_k}\right] \quad (1)$$

In general, the total probability of tie (1, k+1) formation is given by:

$$f(c_1, c_2, \dots, c_k) = \sum_i \Pr(P^i)\, Pr\left(P_1^i, P_{k+1}^i\right) \quad (2)$$

where $\Pr(P^i)$ is the probability of formation for such as sub-cluster. The probability of occurrence of Figure 1 (left) is $(1-p_3)^2$, whereas probability of occurrence of Figure 1 (right) is $p_3(1-p_3)$. Tie (P_1^1, P_4^1) occurrence probability within Figure 1 (left) is obtained as:

$$Pr(P_1^1, P_4^1) = (1-p_3)^2 \frac{c_1 c_3}{c_1 c_3 + (1-c_1)(1-c_3)} \quad (3)$$

Tie (P_1^1, P_4^1) occurrence probability within Figure 1 (right) is obtained as:

$$Pr(P_1^1, P_4^1) = p_3(1-p_3) \quad \frac{c_1 c_2 c_3}{[c_1 c_2 c_3 + (1-c_1)(1-c_2)(1-c_3)]} \quad (4)$$

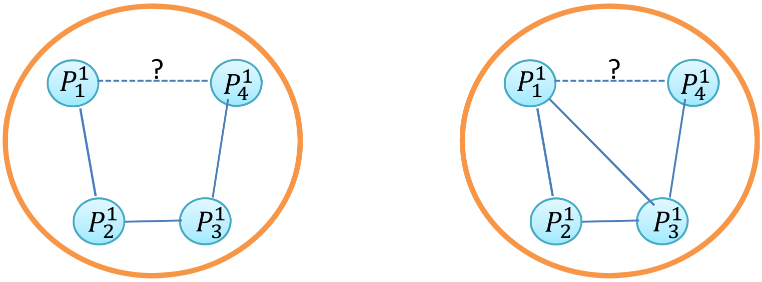


Fig. 1 Illustration of a sub-cluster: tie (P_1^1, P_4^1) occurrence a) with two generation mechanisms of g(1) and g(3) on the left, b) with three generation mechanisms of g(1), g(2) and g(3) on the right.

Here we would like to prove that adding a length-2 tie in Figure 1 (right), as compared to Figure 1 (left), will increase $Pr(P_1^1, P_4^1)$. In order to do that we subtract (4)-(3), rendered as follows:

$$p_3(1-p_3)\frac{c_1c_2c_3}{[c_1c_2c_3+(1-c_1)(1-c_2)(1-c_3)]}-(1-p_3)^2\frac{c_1c_3}{[c_1c_3+(1-c_1)(1-c_3)]}= p_3\frac{(1-c_1)(1-c_3)+2\,c_1c_2c_3}{[c_1c_2c_3+(1-c_1)(1-c_2)(1-c_3)][c_1c_3+(1-c_1)(1-c_3)]}>0 \qquad (5)$$

As easily observed, the $Pr(P_1^1, P_4^1)$ obtained from (4) is greater than the probability obtained from (3). Now we discuss the case, where we investigate tie generation between two clusters as:

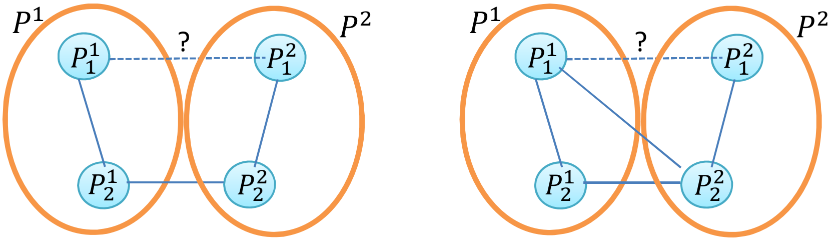


Fig. 2 Illustration of two sub-clusters: tie (P_1^1, P_1^2) occurrence a) with two generation mechanisms of g(1) and g(3) on the left, b) with three generation mechanisms of g(1), g(2) and g(3) on the right.

Tie formation probability $Pr(P_1^1, P_1^2)$ in Figure 2 includes also the probability of selecting cluster P^1equal to Pr (P^1), and probability of selecting cluster

P^2equal to Pr (P^2). In addition, the probability of formation for such a connected sub-cluster as shown in Figure 2 (left) is as:

$$\Pr(random\ sub - cluster) = 1 / \left(\begin{pmatrix} n \\ 2 \end{pmatrix} \atop m\right) \tag{6}$$

where n represents possible number of nodes, m shows possible number of ties within the connected sub-cluster. Finally the probability of $Pr(P_1^1, P_1^2)$ for Figure 2 (left) is given below:

$$Pr(P_1^1, P_1^2) = \Pr(P^1)\Pr(P^2) / \left(\begin{pmatrix} n_{P^1} + n_{P^2} \\ 2 \end{pmatrix} \atop n_{P^1}\, n_{P^2} - 1\right) \frac{c_1 c_3}{[c_1 c_3 + (1 - c_1)(1 - c_3)]} \tag{7}$$

where n_{P^1} represent number of nodes within cluster P^1, and where n_{P^2} represent number of nodes within cluster P^2. Probability of $Pr(P_1^1, P_1^2)$ for Figure 2 (right) is given below:

$$\begin{aligned} Pr(P_1^1, P_1^2) &= \Pr(P^1)\,\Pr(P^2) \\ &/ \left(\begin{pmatrix} n_{P^1} + n_{P^2} \\ 2 \end{pmatrix} \atop n_{P^1}\, n_{P^2} - 2\right) \frac{c_1 c_2 c_3}{[c_1 c_2 c_3 + (1 - c_1)(1 - c_2)(1 - c_3)]} \end{aligned} \tag{8}$$

3 Empirical Analysis

3.1 Data and Measures

We collected data from SourceForge.net, which is the largest repository of OSS projects. Access to the, data was given to us by Computer Science & Engineering, University of Notre Dame. At a time it contained more than 150,000 projects and more than 1,600,000 project developers (as indicated by Howison and Crowston [15]). SourceForge.net website has categorized open source software (OSS) into several categories such as Audio and Video, Business and Enterprise, Communications, Development, Home and Education, Games, Graphics, Science and Engineering, Security and Utilities, System Administration. In addition, SourceForge.net website gives group$_{\text{id}}$ as an identifier for each project. We downloaded data (user$_{\text{id}}$, group$_{\text{id}}$, group project$_{\text{id}}$) for 10,000 developers (user$_{\text{id}}$ <10,000) for the period between Jan 2006-Jan 2014 from SourceForge repository based on multidimensional table. Different developers (users) intra and inter groups are connected by groupprojects, where each developer contributes in different projects within and between groups. Each group$_{\text{id}}$ includes some project$_{\text{id}}$ and some user$_{\text{id}}$, assigned by SourceForge Administration to provide better search possibility.

In the empirical analysis section, we use a complex network of open source software (OSS) project collaboration as our domain of interest. In this network, developers represent the nodes and a developer contributing to a group project represents a network tie. In addition, a group (including team of developers and projects) is represented by group$_{id}$ assigned by SourceForge. As previously mentioned, Intragroup tie represent developers (user$_{id}$) contributing to different projects within a single group; whereas Intergroup tie represent developer (user$_{id}$) contributing to projects from two different groups. Intergroup tie strength can be a function of sum of intragroup ties (or density), as can be also inferred from Figure 3 and Figure 4. One can also consider number of projects and number of developers per each group as control variables.

$$\begin{aligned} &intergroup\ tie\ stength \\ &\qquad = f(\#intragroup\ tie, \#project_{id}, \#user_{id} | group_{id}) \end{aligned} \tag{9}$$

In order to show relationship between intergroup tie strength and intragroup density, we use regression modeling. Intergroup tie strength is calculated by sum of number of ties between two group at time t denoted by S_t . Then the regression model can be shown as:

$$\begin{aligned} S_t\,(intergroup\ tie) &= a_0 + a_1 \cdot S_t\,(intragroup\ tie | group_{id}) \\ &+ a_2 \cdot S_t\,(user_{id} | group_{id}) + \ a_3\ \cdot S_t\,(project_{id} | group_{id}) \end{aligned} \tag{10}$$

As given above, the dependent variable is number of ties between two groups represented by $S_t\,(intergroup\ tie)$. We used regression model based on 1) OSS data over 9 years jan 2006 – jan 2014, 2) Sum of intergroup ties with respect to intragroup density, 3) Sum of intergroup ties with respect to sum of intragroup ties and number of projects and number of users within each group.

3.2 Empirical Results

Before we show the results obtained by applying the regression model, we demonstrate why relationship between intergroup tie strength and intragroup density makes sense. As easily seen in Figure 3, there seems to be a linear relationship for data in Jan 2014. In the next step, we average over data obtained for 9 years (Jan 2006 – Jan 2014), and calculate the intergroup tie strength and intragroup density, and illustrate them in Figure 4.

In order to prove our hypothesis, we do the statistical analysis. As shown in Table 1, intra-group density has positive and significant influence on inter-group tie strength (a_1 = 0.371, and p-value = 0.000). In addition, relationship between number of developers and inter-group tie strength is negative and significant (a_2 = -0.083, and p-value = 0.005). This makes sense, because as the number of developers within a group increases, it has initial impact on number of intragroup ties rather than intergroup tie strength. Perhaps it is because developers would rather work on their own projects within the group rather than working on projects from other groups. Finally, number of projects within a group has insignificant influence on inter-group tie strength (a_3 = 0.001, and p-value = 0.920). This is intuitive, because our complex network is based on developers as nodes and a developer working on a project as tie, and developer working on projects from different groups create intergroup tie; therefore, increasing number of projects rather than developers do not have a direct influence on intergroup tie strength.

As we saw in Table 1, independent variable (#projects per group$_{id}$) do not have significant influence on the dependent variable (intergroup tie strength); therefore, it could be eliminated from the regression model. However, once intragroup tie sum is divided over number of users per group which gives intragroup density, then the two independent variables within the regression model can be merged; therefore, *intergroup tie strength* could be directly proportional to *intragroup density* as shown in Figure 3 and Figure 4.

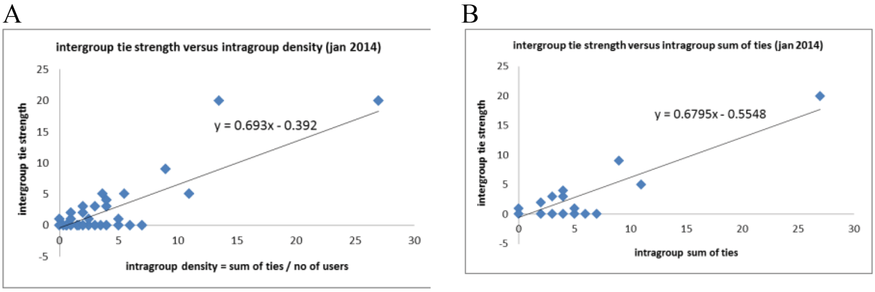


Fig. 3 Illustration of intrergroup tie strength versus intragroup density (left figure), and versus sum of intragroup ties (right figure) per group$_{id}$ for Jan 2014.

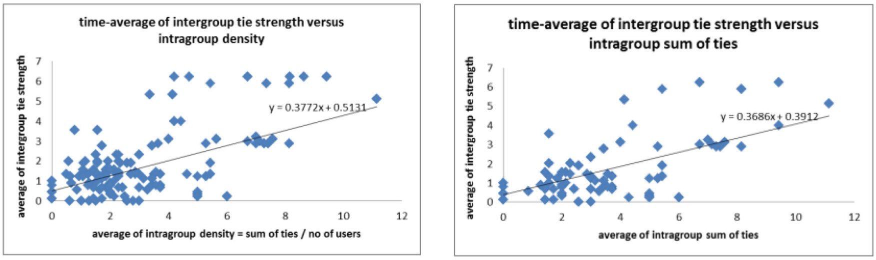


Fig. 4 Illustration of intrergroup tie strength versus intragroup density (left), and versus sum of intragroup ties (right) per group$_{id}$ for data averaged over Jan2006 - Jan2014.

Table 1 calculation of coefficients based on regression model

	Coefs	*Std*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.495	0.045	10.946	0.000	0.406	0.584
Intragroup tie sum	0.371	0.009	42.098	0.000	0.354	0.389
#users per group$_{id}$	-0.083	0.029	-2.826	0.005	-0.140	-0.025
#projects per group$_{id}$	0.001	0.009	0.100	0.920	-0.016	0.018

4 Conclusion

In this paper, we investigated how to predict inter-cluster tie strength, and proposed that inter-cluster tie strength can be predicted from determination of intra-cluster density. We used both analytical and empirical analysis to prove our hypothesis. To predict inter-cluster tie strength, we used the method of tie generation based on the proposed tie probability model by Huang [14], and provided the probability functions based on the proposed clustering model. In the empirical analysis section, we used a complex network of open source software (OSS) project collaboration as our domain of interest, and did statistical analysis to prove the hypothesis. As shown in Table 1, intra-group density has positive and significant influence on inter-group tie strength ($a_1 = 0.371$, and p-value = 0.000), which support our hypothesis. In addition, relationship between number of developers and inter-group tie strength is negative and significant ($a_2 = -0.083$, and p-value = 0.005). Finally, number of projects within a group has insignificant influence on inter-group tie strength ($a_3 = 0.001$, and p-value = 0.920).

We collected OSS data from SourceForge.net, which is the largest repository of OSS projects. We appreciate access to this repository given by prof. Greg Madey at the department of Computer Science & Engineering, University of Notre Dame.

References

1. Granovetter, M.: The Strength of Weak Ties. American Journal of Sociology **78** (1973)
2. Newman, M.E.J.: Scientific collaboration networks. ii. Shortest paths, weighted networks and centrality. Phys. Rev. E. **64** (2001)
3. Weick, K.E.: Educational Organizations as Loosely Coupled Systems. Administrative Science Quarterly **21**(1), 1–19 (1976)
4. Hansen, M.T.: The Search-Transfer Problem: The Role of Weak Ties in Sharing Knowledge across Organization Units. Administrative Science Quarterly **44** (1999)
5. Ruef, M.: Strong ties, weak ties and islands: structural and cultural predictors of organizational innovation. Industrial and Corporate Change **11**(3) (2002)
6. Shu, P., Tang, M., Gong, K., Liu, Y.: Effects of Weak Ties on Epidemic Predictability in Community Networks. Statistical Mechanics (2012)
7. Aral, S., Walker, W.: Creating Social contagion through viral product design: A randomized trial of peer influence in networks. Management Science **57** (2011)

8. Behfar, K., Turkina, E., Cohendet, P., Animesh, A.: Analytical Modeling of Social Network Growth Using Multilayer Network Projection. IJCSN **2**(6) (2013)
9. Joshi, K.D., Sarker, S.: Knowledge transfer within information system development teams: Examining the role of knowledge source attributes (2006)
10. Ojha, A.K.: Impact of team demography on knowledge sharing in software project teams. South Asian Journal of Management **12**(3), 67–78 (2005)
11. Hahn, J., Moonm, J.Y., Zhang, C.: Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. ISR **19** (2008)
12. Hinds, D., Lee, R.M.: Social Network Structure As a Critical Success Condition for Virtual Communities. Proceeding of Int. Conf. on System Sciences HICSS (2008)
13. Antwerp, M.V., Madey, G.: The importance of social network structure in the open source software developer community. In: 43rd Hawaii Int. Conf. on Sys. Sci. (2010)
14. Huang, Z.: Tie Prediction Based on Graph Topology: The Predictive Value of the Generalized Clustering Coefficient. ACM SIGKDD, Philadelphia, USA (2006)
15. Crowston, K., Howison, J.: The Social Structure of Free and Open Source Software development. Peer-Reviewed Journal of Internet **10**(2–7) (2004)

Soft-Sensors for Lipid Fermentation Variables Based on PSO Support Vector Machine (PSO-SVM)

Carlos Eduardo Robles-Rodriguez, Carine Bideaux, Gilles Roux, Carole Molina-Jouve and Cesar Arturo Aceves-Lara

Abstract On-line monitoring fermentation variables (*e.g.* biomass) can improve the performance of bio-processes, as well as the quality of the targeted products. However, on-line estimation could be a challenging task when an accurate model is not available. Over the existing methods for state estimation, the support vector machine (SVM) is an attractive method for its fast convergence and generalization of the approximated function. In this paper, a soft-sensor based on SVM and coupled to Particle Swarm Optimization (PSO) algorithm is presented and applied to estimate the concentrations of lipid fermentation variables: lipids, biomass, and citric acid. The soft-sensor was trained with one data set, and validated with an independent data set of fed-batch fermentations. The PSO-SVM was compared with the SVM algorithm. In general, the results show that the PSO-SVM is an efficient alternative for monitoring fermentations.

Keywords Support vectors machine · Soft-sensor · PSO · Lipid fermentation · Biomass

C.E. Robles-Rodriguez · C. Bideaux · C. Molina-Jouve · C.A. Aceves-Lara(✉)
Université de Toulouse, UPS, INSA, INP, LISBP, 31077 Toulouse, France
e-mail: {roblesro,aceves}@insa-toulouse.fr

C.E. Robles-Rodriguez · C. Bideaux · C. Molina-Jouve · C.A. Aceves-Lara
INRA, UMR792, Ingénierie des Systèmes Biologiques et des Procédés, Toulouse, France

C.E. Robles-Rodriguez · C. Bideaux · C. Molina-Jouve · C.A. Aceves-Lara
CNRS, UMR5504, 135 Avenue de Rangueil, 31077 Toulouse Cedex, France

G. Roux
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_19

1 Introduction

The monitoring of certain variables is one of the main issues of optimizing fermentation because the yields of the targeted products can be estimated. Usually, it is difficult to measure online substrate, biomass, and product concentrations in the bio-process. The so-called “soft-sensors” are an alternative for on-line estimation. Soft sensors are software based sophisticated monitoring systems, which can relate the infrequently measured process variables with the easily measured [1]. In this way, these soft-sensors assist in making the real-time prediction of the unmeasured variables [2].

Several software sensors have been proposed such as, modeling of mechanisms [3], adaptive observers [4], and Support vector Machine (SVM) [1]. The first two required a detailed model of the process. Model development is sometimes a complicated task because it is not feasible to find a trade-off between simplicity and precision. Furthermore, SVM is a simpler and powerful alternative based on statistical learning with small samples. This method shares many of its features with the artificial neural networks, but it proposes some additional characteristics [5]. It has good generalization ability of the regression function, robustness of the solution, and sparseness of the regression. Moreover, SVM also provides an explicit knowledge of the data points, which are important in defining the regression function. This methodology has been scarcely used in bio-processes/technology applications [1, 6–8] including fermentation of penicillin, and the estimation of biomass, among others. Nevertheless, the realization of the advantages depends on the correct choice of parameters.

Particle Swarm optimization (PSO) is a global and efficient method of parameter estimation. This is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [9] and motivated by the behavior of organisms such as fishing schooling and bird flock. The algorithm is a validated evolutionary computation way of searching the extremum of function, which is simple in application and quick in convergence [10].

In this context this work presents an extension of the SVM by PSO to estimate the parameters of the SVM in order to reach faster the optimal solution. This new PSO-SVM based soft-sensor is applied to estimate lipids, biomass, and citric acid concentrations from fed-batch fermentation cultures. The methodology of SVM and PSO-SVM is presented in section 2. Section 3 holds the results of the soft-sensor with the training and validation data. Finally, section 4 reports the conclusions and some perspectives of this work.

2 Methods

2.1 Support Vector Machine

The support vector machine theory relies on the idea to trap the input data into a feature space via nonlinear mapping. SVM maps the inner product of the feature space on the original space via kernels. Based on a given set of training data where $x \in R^N$ is the vector of the model inputs, and $y \in R$ is the vector of the scalar outputs [5]. The objective of the regression analysis is to determine a function that predicts accurately the desired outputs y in the form

$$y = w^T.\phi(x) + b \tag{1}$$

where $\phi(x): R^N \to R^M$ is the high dimensional feature space which is non-linearly mapped from the input space x. The vector w and the coefficient b are estimated by

$$\min_{w,\,\xi,\,\xi^*} J = \frac{1}{2} w^T.w + C.\sum_{i=i}^{N} \left(\xi_\imath + \xi_i^*\right) \tag{2}$$

Subject to

$$\begin{cases} y_i - w^T.\phi(x_i) - b \le \varepsilon + \xi_i \\ w^T.\phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* \ge 0 \end{cases} \tag{3}$$

In equation (2) the first term is the regularized term, and the second term is the empirical error (risk) measured by the insensitive ε-loss function enabling to use less data points to represent the decision function given by equation (1). The variables ξ_i and ξ_i^* are the slack variables that measure the deviation of the support vectors from the boundaries of the ε-zone. The constant C is the regularization constant. It determines the trade-off between the empirical risk and the regularized term. The term ε is called the tube size and it is equivalent to the approximation accuracy placed on the training data points. Both parameters will determine the efficiency of the estimation [11].

In order to simplify the minimization problem, Lagrange multipliers are introduced as follows:

$$\begin{gathered} L = J - \sum_{i=1}^{N} \alpha_i \left\{\xi_i + \varepsilon - y_i + w^T.\phi(x_i) + b\right\} - \\ \sum_{i=1}^{N} \alpha_i^* \left\{\xi_i^* + \varepsilon + y_i - w^T.\phi(x_i) - b\right\} - \sum_{i=1}^{N} \left(\eta_i.\xi_i - \eta_i^*.\xi_i^*\right) \end{gathered} \tag{4}$$

where the parameters α, α^*, η, and η^* are the Lagrange multipliers. According to the Karush-Kuhn-Tucker (KKT) of quadratic programming, the dual equation that can be obtained [6] is:

$$\begin{gathered} \min_{\alpha,\,\alpha^*} W = \frac{1}{2} \sum_{i,j=i}^{N} \left(\alpha_\imath - \alpha_i^*\right).\left(\alpha_j - \alpha_j^*\right).K\left(x_i, x_j\right) + \\ \varepsilon.\sum_{i=i}^{N} \left(\alpha_\imath + \alpha_i^*\right) - \sum_{i=i}^{N} \left(\alpha_\imath - \alpha_i^*\right).y_i \end{gathered} \tag{5}$$

Subject to
$$\begin{cases} \sum_{i=i}^{N}\left(\alpha_i - \alpha_i^*\right) \\ 0 \le \alpha_i, \alpha_i^* \le C; \quad i = 1,2,\ldots,N \end{cases} \tag{6}$$

Therefore, the final regression function given in equation (1) is rewritten as

$$y = \sum_{i=i}^{N}\left(\alpha_i - \alpha_i^*\right).K\left(x_i, x_j\right) + b \tag{7}$$

where $K\left(x_i, x_j\right) = \phi\left(x_i\right).\phi\left(x_j\right)$ is the kernel function.

2.2 Kernel Functions

Different kernels can be selected to construct different types of SVM. The kernel can be any symmetric function satisfying the Mercer's condition. Typical examples include [12]:

Polynomial kernels $K\left(x_i, x_j\right) = \left[\left(x.x_i\right) + 1\right]^d$;

Radial Basis Function (Gaussian) kernels $K\left(x_i, x_j\right) = \exp\left(-\left(\left\|x - x_i\right\|^2 / 2\sigma^2\right)\right)$;

The most used Kernel function is the Radial Basis Function (RBF) because it can classify multi-dimensional data, and it has fewer parameters to be set in comparison with other kernels. The parameter σ represents the width of the RBF.

2.3 PSO-SVM

The accuracy of the results of SVM is linked to the setting of user parameters such as *C*, *σ,* and *ε.* The parameter *C* determines the trade-off between model complexity and model degree to which deviations larger than *ε* are tolerated [6]. Kernel parameter *σ* determines the kernel width and relates to the input range of the training data set. If *C* is too large the estimation accuracy rate is very high in the training phase, but very low in testing phase. If *C* is too small, the estimation accuracy is unsatisfied, and the model is useless. An excessively large value for parameter *σ* leads to over fitting, while a disproportionately small value to under fitting. It is thus seen that SVM parameters are a combinatorial optimization problem, in which every combination generates a new solution function for equation (7). This optimization problem can be solved by cross verification trial which can result on a local minimum and costs high calculation times. Another solution is the use of global algorithms, such as particle swarm optimization (PSO)[9].

PSO is a simple algorithm implying a small quantity of parameters to be adjusted with fast convergence. In this paper, the objective function of PSO is given by the Akaike Information Criterion (AIC) to reduce the problem to a selection of best models [13]. The *AIC* formula is given by

$$\mathrm{AIC} = n\left(\ln\left(2\pi\right) + 1\right) + n.\ln\left(\frac{SSE}{n}\right) + 2.p \tag{8}$$

where n is the sample size, p is the number of factors in the model, SSE is the squared error: $SSE = 1/n.\sum_{i=1}^{n}\left(y_i - y_i^*\right)^2$, where y_i correspond to the off-line data, and y_i^* are the estimates of the model.

3 Results and Discussion

The method PSO-SVM was applied to construct a soft-sensor of biomass, lipids, and citric acid concentrations from the lipid fermentation of *Yarrowia lipolytica* growing on glucose. Two independent data sets performed in fed-batch cultures [14] were used in this work. Three variables were available from on-line measurements (Fig. 1): (1) added base to control pH, (2) volume, (3) partial oxygen pressure (pO2).

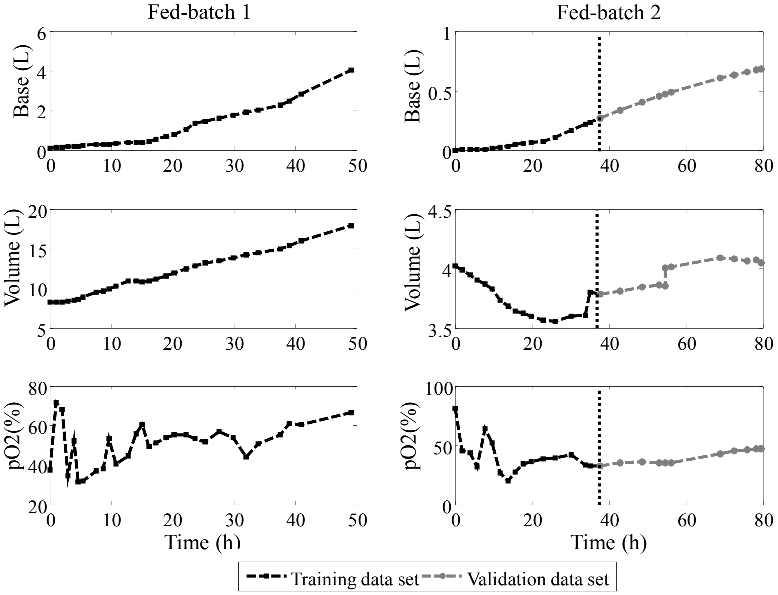


Fig. 1 Available on-line measurements for the fed-batch cultures.

The combination of several variables was tested to design the soft-sensors (Table 1). The discrimination of measurement was considered by the AIC values and the visualization of the results. Table 1 presents the values AIC values for training (AICt) and validation (AICv) for lipid concentration, where the combination of Base - pO2 reported low values for both sets. This methodology was also applied for the biomass concentration and the citric acid mass soft-sensors (Results not shown). The selected measurements for the PSO-SVM based soft-sensors are shown in Fig. 1.

Table 1 Combinations to determine the best set for validation of lipid concentration.

Variable	AICt	AICv	Variable	AICt	AICv
Base	-673.89	67.24	pO2 – Base/Volume	-691.27	39.68
pO2	-144.35	20.14	pO2*Volume	-690.85	19.49
Volume	-690.63	2.46	pO2/Volume	-690.82	23.20
Base - pO2	-691.14	5.43	Base - pO2*Volume	-690.85	12.41
Base - Volume	-690.79	12.75	Base - pO2/Volume	-690.84	24.01
pO2 – Volume	-690.79	12.75	Base*Volume	-690.77	24.12
Base - pO2 - Volume	-690.81	15.44	pO2 – Base*Volume	-691.79	26.28
Base/Volume	38.51	103.61			

The kernel function was the RBF Gaussian function. The PSO follows Clerc's constriction factor [10] using both learning factors (i.e. c_1 and c_2) at 1.5, with an initial inertia weight of 0.72. The population size was settled to 30 to consider a moderate search within the grid. The maximum number of iterations was fixed to 20 where after several runs the AIC variation was less than 0.001 in the AIC value.

The two independent data sets [14] were used to train and to validate the soft-sensor based on PSO-SVM. The first data set (Fed-batch 1, 29 data points), and the 60% of the other (Fed-batch 2, 16 data points between t = 0 - 38h) were used

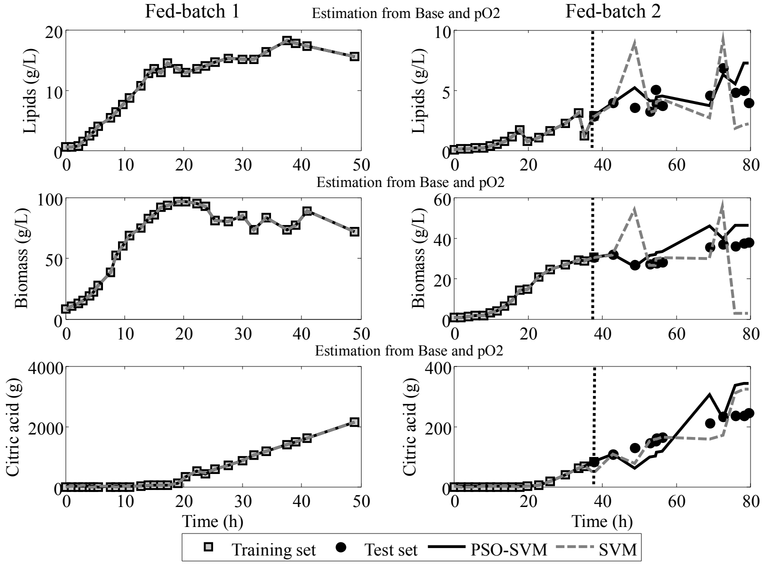


Fig. 2 Training and test results of soft-sensor based on PSO-SVM.

to train and to generate the support vectors of the PSO-SVM. In all the cases of the soft-sensor for lipid and biomass concentrations, and citric acid mass the used measurements were the added base to control pH and the pO2.In order to test the ability of the PSO-SVM, the 40% left of the second data set (12 data points from t = 38 - 79h) was employed to validate the regression of the soft-sensor. In addition, standard SVM was used to compare and to illustrate the improvement of PSO-SVM. The parameters for standard SVM were taken from [1].

Fig. 2 displays the results of the training and test sets based on SVM and PSO-SVM. The dashed line separates the results of the training from the test sets. The values of *C, σ,* and *ε* and the numerical comparison of results are presented on Table 2, where the *AIC* and *SSE* are used as performance indicators. The three soft-sensors reflect an acceptable fitting with the training data set with both PSO-SVM and SVM. However, *AIC* values for SVM are higher than the ones from PSO-SVM. This reflects that the selection of the parameters *C, σ,* and *ε* by PSO improves the fitting. The values of *ε* were small reflecting that the width of the tube surrounding the data is also small. This value allows having good results in the training sets, but it increases the risk of poor generalization performance of the estimator.

Table 2 Comparison of performance evaluation of SVM and PSO-SVM.

Variable	Parameters			AICt		AICv		SSEv	
		SVM	PSO-SVM	SVM	PSO-SVM	SVM	PSO-SVM	SVM	PSO-SVM
Lipids	C	15000	3220	-484.8	-691.14	17.57	5.43	2.48	1.61
	ε	0.001	0.001						
	σ	1.5	0.724						
Biomass	C	15000	4260	-450.6	-690.61	140.6	79.30	200.5	22.47
	ε	0.001	0.001						
	σ	1.5	0.38						
Citric acid	C	15000	9160	107.62	-687.4	188.11	200.45	1095.1	1701.4
	ε	0.001	0.001						
	σ	1.5	0.58						

Regarding the results on the test data, lipid and biomass concentrations reflect acceptable results for PSO-SVM, where the estimation follows the behavior of the state variables. Otherwise, SVM shows a highly disturbed behavior on those concentrations. This is depicted by the higher AICv and SSE for SVM. The soft-sensors for citric acid mass presented a similar regression varying around the values all over the test data. Nevertheless, the AICv and SSE for PSO-SVM were higher than SVM exhibiting a better performance of SVM. This is due to the parameters selection by PSO which was defined to find the best training set.

The CPU times to do the training (t) with PSO-SVM were 317.7, 458.6, and 586.04 seconds for lipids, biomass, and citric acid respectively. The time to calculate the state variables (v) was 0.064s for the three of them, which is an encouraging value to couple the soft-sensor on-line.

4 Conclusions

This work introduced a soft-sensor based on PSO-SVM for monitoring fermentation processes. The addition of PSO to SVM reduced the uncertainty generated by manually choosing the SVM parameters or the cross fitting computation time. The PSO-SVM method was successfully implemented to estimate lipids and biomass concentrations in fed-batch cultures. The results proved that PSO-SVM is an attractive and simple alternative for monitoring. Moreover, this method implemented a well trade-off between the quality of the approximation of the given data and the complexity of the approximating function. Future work will focus on increasing the fitness quality and extending the usefulness of the soft-sensor in other experimental working conditions.

Acknowledgments This work benefits of French financial support by Tereos, Agence Nationale de la Recherche, and Commissariat aux Investissements d'Avenir via the Project (ref. ANR-11-BTBT-0003) ProBio3: Biocatalytic production of lipid bioproducts from renewable resources and industrial by-products: Biojet Fuel Application.

References

1. Wang, X., Chen, J., Liu, C., Pan, F.: Hybrid modeling of penicillin fermentation process based on least square support vector machine. Chem. Eng. Res. Des. **88**, 415–420 (2010)
2. Desai, K., Badhe, Y., Tambe, S.S., Kulkarni, B.D.: Soft-sensor development for fed-batch bioreactors using support vector regression. Biochem. Eng. J. **27**, 225–239 (2006)
3. Ramkrishna, D.: On modeling of bioreactors for control. J. Process Control **13**, 581–589 (2003)
4. Bastin, G., Dochain, D.: On-line Estimation and Adaptive Control of Bioreactors. Elsevier (1990)
5. Vapnik, V., Golowich, S.E., Smola, A.: Support vector method for function approximation, regression estimation, and signal processing. Annu. Conf. Neural Inf. Process. Syst., 281–287 (1996)
6. Liu, G., Zhou, D., Xu, H., Mei, C.: Model optimization of SVM for a fermentation soft sensor. Expert Syst. Appl. **37**, 2708–2713 (2010)
7. Ou Yang, H.-B., Li, S., Zhang, P., Kong, X.: Model penicillin fermentation by least squares support vector machine with tuning based on amended harmony search. Int. J. Biomath. **08**, 1550037 (2015)

8. Nadadoor, V.R., De la Hoz Siegler, H., Shah, S.L., McCaffrey, W.C., Ben-Zvi, A.: Online sensor for monitoring a microalgal bioreactor system using support vector regression. Chemom. Intell. Lab. Syst. **110**, 38–48 (2012)
9. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: MHS 1995. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., pp. 39–43 (1995)
10. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), pp. 81–86. IEEE (2001)
11. Jianlin, W., Tao, Y.U., Cuiyun, J.I.N.: On-line estimation of biomass in fermentation process using support vector machine. Chinese J. Chem. Eng. **14**, 383–388 (2006)
12. Yan, W., Shao, H., Wang, X.: Soft sensing modeling based on support vector machine and Bayesian model selection. Comput. Chem. Eng. **28**, 1489–1498 (2004)
13. Burnham, K.P.: Multimodel Inference: Understanding AIC and BIC in Model Selection. Sociol. Methods Res. **33**, 261–304 (2004)
14. Cescut, J.: Accumulation d’acylglycérols par des espèces levuriennes à usage carburant aéronautique: physiologie et performances de procédés. PhD Thesis. INSA, Toulouse, France (2009)

An Underwater Target Recognition Method Based on Tracking, Trajectory, and Optimum Seeking Data Joint

Liang Yu, Yong-mei Cheng, Ke-zhe Chen, Jian-xin Liu and Zhun-ga Liu

Abstract Most of the underwater target recognition method are built on the spectral analysis. The recognition accuracy isn't high in the short attack time which leads the torpedo attack on the false target. A method which take the tracking, trajectory and optimum seeking data join have been put forward, which can use the targets' various information for tracking, trajectory control, optimum seeking, so multiple information can be used for other processes. The method can take full advantage of all key data during the attacking progress which can improve the recognition and attack accuracy. The final attack simulation results verified the high accuracy of the method.

Keywords Acoustics · Automatic target recognition · Military electronic countermeasures · Target tracking · Feature · Trajectory · Optimum seeking

1 Introduction

Under the underwater acoustic countermeasure environment, when the torpedo attacking the real target, it may release some false targets, as the decoy. As the development of the electronic technology, high quality decoys can simulate the real

L. Yu(✉) · Y.-m. Cheng · J.-x. Liu · Z.-g. Liu
Department of Automatic Control, Northwestern Polytechnical University,
Xi'an 710072, Shaanxi, China
e-mail: ylozy@126.com

L. Yu
Engineering Practice and Training Center,
Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

K.-z. Chen
The Twentieth Research Institute of China Electronic Technology Group Corporation,
Xi'an 710068, Shaanxi, China

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_20

target's motion feature and acoustic feature, so it is very hard to identify them in short time that the torpedo has a large probability to attack the wrong target so that the attack failed.

Now, most of the underwater target identify methods are focused on the targets' acoustic feature, for example, the targets' time domain characterization, frequency domain feature and the time-frequency domain features to identify them [1-2], as the spectral centroid [3], LOFAR spectrum [4], the spatial scales [5], the echo of the spatial orientation distribution [6], dimensional spectrum [7], and so on. The support vector machines combined with genetic algorithm [8], genetic algorithm based fuzzy logic [9], model matching method [10], and adaptive immune feature selection algorithm [11] had been proposed for analysis these spectral data. However, it should be noted that, firstly, it is hard to extract the original underwater acoustic target samples, there exist extraordinary redundancy rate during the target characteristic extraction process. Then, the decoy could simulate completed almost all of the time domain and frequency-domain characteristics of the real target, just from the time-frequency information only, the torpedo is very hard to identify the real target and decoys timely and correctly. So, it is difficult to distinguish them accurately just depend signal processing, and more information should be utilized. Base on the hierarchical information fusion method, the motion data which got by data processed has also been utilized to identify the targets [12]. The article expands the acoustic target recognition technology from just the spectral features to motion and acoustic features. The simulation results demonstrate that the method can effectively identify the targets.

But, all the method above are based on that the torpedo is static or when the torpedo is moving, other changing information has been ignored. The tracking, attacking, the identifying process is at the same time. The entire data in the attacking process could be used to the trajectory design, tracking, to each other. For example, if we got the identify results, we can know which one is more like the authentic target, so we can design the trajectory to identify the target better and sooner. Also, targets' feature information could not only use for probability calculation that the target is the actual target, but also it can be used to the tracking process, to compute the position or velocity which can help to make it faster and more accurate. So, finding a method which can include targets' tracking requirement, trajectory control and target feature information acquisition together to identify the target, which is the necessary requirement of the technology and it will play a major role to improve the identify effect.

Based on the idea that the information got by tracking can be used for others, al-so, other information can be used to identify, which can make the identify more ac-curacy and faster, a new method that which taking tracking, trajectory, and optimum seeking data joint for identifying the underwater targets is provided. The paper is structured as follows. In Section 2, this method's structure are described. In Section 3, the thought for tracking using feature information is provided. In Section 4, the joint data identify method which uses more information is built, and in section 5, the optimum seeking result has been used to ballistic trajectory design, the effect of the whole process has been verified through a series of simulations in Section 6. Section 7 concludes this paper.

2 Structure of the Method

When we got the targets' feature information, we use it for comparing with the actual target's feature, to compute the targets' plausibility probability and then get all targets' optimum seeking results. Supposing that we have got the results, next we should design the trajectory to track the targets. Firstly, we should ensure that the target of maximum plausibility probability would not be lost, then should ensure that as many as possible targets should be tracked simultaneously and be not lost. Only in which way can we obtain the features of targets as many as possible. So, the time can be extended for multi-target identifying. Next shooting time, according to the trajectory guidance results, the sports orientation of the torpedo would change, using the feature information, we could finish the multi-target track in the ensuing shoot, and repeat the above process. It is clear that in the complex underwater acoustic environment, three factors, tracking, optimum seeking and trajectory design, form a relation to influence and to condition each. The overall framework of the method is shown in Fig. 1.

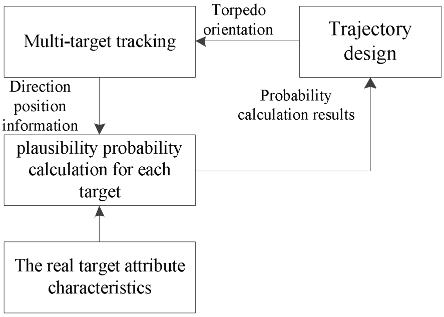

Multi-target tracking
Torpedo orientation
Trajectory design
Direction position information
Probability calculation results
plausibility probability calculation for each target
The real target attribute characteristics

Fig. 1 Joint data identify method framework in complex underwater acoustic environment

3 Feature Aided Underwater Multi-target Tracking

The acoustic feature filtering process is parallel to the motion data filtering process. The main problem of the method is the way how to do the data association, how to get the association gate size.

The measurement confirmation formula for our method can be defined as,

$$d^2(Z)+d_L^2(Z_L)\le\gamma \tag{1}$$

where, γ is the relation gate size. As the introduction of features， we should know that one dimension should be added to the measurement dimension. Algorithm flow of the feature aided nearest neighbor tracking method (FA-NNF) is shown in Fig. 2. The subscript L represents the acoustic feature.

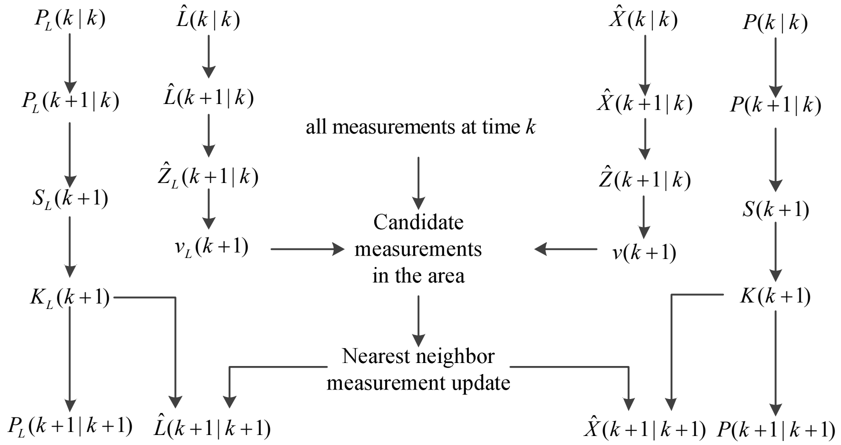


Fig. 2 Algorithm flow of the FA-NNF

4 Optimum Seeking with More Information

Using the feature and motion data simultaneously, hierarchical information fusion algorithm structure for underwater acoustic target optimum seeking is illustrated in Fig. 3.

All information that the direction, speed, et al., motion information; the radiated noise, target strength, et al., real time acoustic features could be utilized to do the target optimum seeking. During the targets tracking process, the real time motion information can be got by the filtering, acoustic features can be got according to the time and frequency domain analysis, and furthermore, bearing and other information of the target can be estimated. The above data are general referred as measured value, and the real value at same time is called as theoretical value; of course, the theoretical model and the error model of the feature can be built to compute the probability of that the target is the real target or the decoy. Firstly, the feature theoretical models of the real target and decoy have been built. Then, compute the differences between theoretical value and the measured value of various attributes, the differences are used to build as the basic confidence function. Then the optimum seeking method based on the DS evidence theory could be realized as the following step. At the first level of the fusion structure, all acoustic features will be fused and all motion information be fused respectively. At different distance and orientation, the credibility of the motion information and acoustic feature is changed. So, the fusion results of the first level will be weighted merged at the second level. After that, It can be judge that, whether the difference between largest and second largest probability result of all targets which get by second level smaller than the threshold ε, if the difference bigger than ε, use the second level results as the final results, otherwise, fusion in the time dimension, that is, fusion at the second level and the previous time results will be done to get the final probability results.

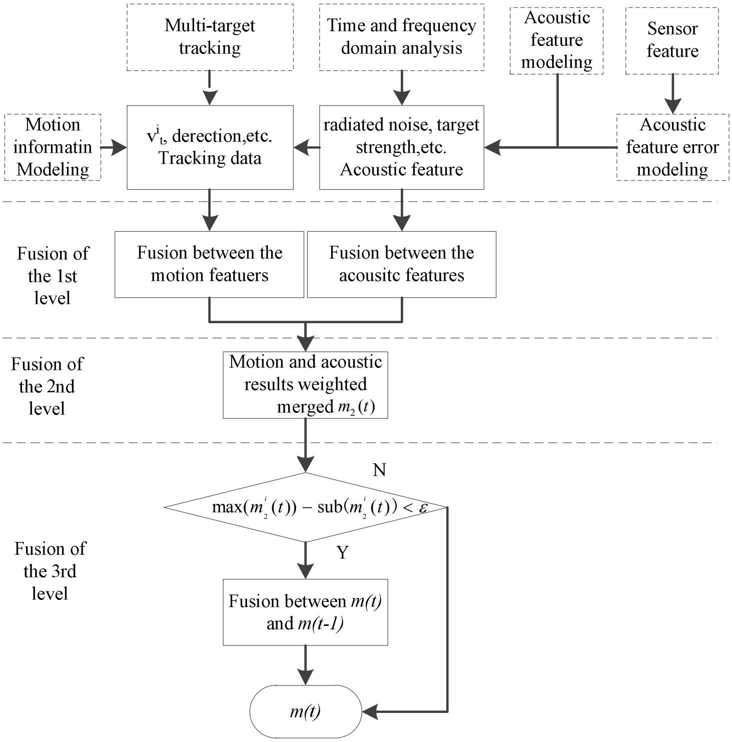

Multi-target tracking
Time and frequency domain analysis
Acoustic feature modeling
Sensor feature
Motion informatin Modeling
v^i_t, derection,etc. Tracking data
radiated noise, target strength,etc. Acoustic feature
Acoustic feature error modeling
Fusion of the 1st level
Fusion between the motion featuers
Fusion between the acousitc features
Fusion of the 2nd level
Motion and acoustic results weighted merged $m_2(t)$
N
$\max(m^i_2(t)) - \mathrm{sub}(m^i_2(t)) < \varepsilon$
Y
Fusion of the 3rd level
Fusion between $m(t)$ and $m(t-1)$
$m(t)$

Fig. 3 Hierarchical information fusion algorithm structure for optimum seeking

5 Trajectory Designs by the Comprehensive Target Situation

As the sector of the torpedo is limited, when we do the trajectory design, the most important is that we should keep the highest probability target which got by the optimum seeking method will not be lost in next time, and more targets should be detected and kept tracking. Based on the thought of the artificial potential field method (APFM), using the form of the velocity vector field, the artificial potential field method applied to torpedo multi-objective guidance has been attempted. The principle schematic based on APFM of underwater multi-target guidance method, is shown in Fig. 4.

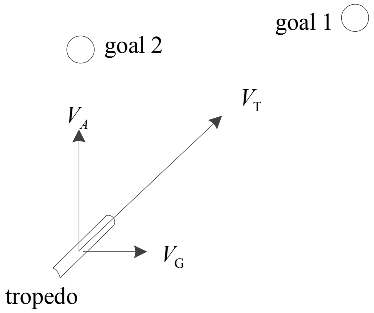


Fig. 4 The guidance diagram based on artificial potential field method

On the basis of the plausibility probability P_i of each target obtained in the real-time process, choose the target of the highest probability as the priority target for the torpedo (defined as traction target T), for example, the goal 1 showed in Fig. 4. The targets within the scope of the torpedo detection will not only produce the repulsive force, but also attract the torpedo (defined as attract targets $A_i, i = 1,2,...,n-1$, *n* is the target number, for example, the goal 2). The traction target produces traction field, which is called drawing attention V_T . The attract target generates a gravitational field, which is called attracting attention V_A . Drawing attention and attracting attention to produce synthetic attention. To avoid the case that the $\sum V_A$ may be bigger than V_T , the torpedo will be seriously deviated from the direction of traction target, the V_G , which produced by attract targets and always towards the direction of traction target is introduced. V_G is perpendicular to V_A , and its direction points to the traction target.

The sign *A* represents the attract target, *T* represents the traction target, so the potential field configuration is expressed as:

$$U = U_A + U_T \tag{2}$$

Where, U stands for the torpedo suffered total potential field.

As there are multiple false targets, the total potential field can be expressed as,

$$U = U_T + \sum_{i=1}^{n} U_A \tag{3}$$

According to the spatial dynamics equation and the Lagrange equation, the role of the artificial potential field for the controlled objects can be derived as,

$$F_T = -grad\left(U_T\right), F_A = -grad\left(U_A\right) \tag{4}$$

F_T is the drawing force; F_A is the attractive force.

F determines the movement of the controlled objects. Generally, there is just one drawing force and many multiple attractive forces, so the resultant force is:

$$F = F_T + \sum_{i=1}^{n} F_A \tag{5}$$

Where *n* represents the number of attractive targets.

According to Newton's mechanics, when the distance between the torpedo and the target is far, the torpedo can be seen as a particle, so,

$$ma = ma_T + m\sum_{i=1}^{n} a_A = m\left(a_T + \sum_{i=1}^{n}(a_A + a_G)\right) \tag{6}$$

a is acceleration for the torpedo, $a = \dot{V}$, $\dot{V}$ is torpedo's acceleration, according to formula (8), we can obtain,

$$\dot{V} = \dot{V}_T + \sum_{i}^{n-1}(\dot{V}_A^i + \dot{V}_G^i) \tag{7}$$

$\dot{V}$ represent the torpedo's acceleration by all targets, is the synthetic attention referred in the paper. In fact, the effect of each target for the torpedo is virtual, and its acceleration is also virtual, that this effect can be developed according to the design purpose. What's the trajectory design considers is the probability of each target, direction and position between the torpedo and targets, and trajectory smoothness requirements.

6 Simulation Results

The simulation system that the torpedo and all the targets could move separately has been built basing on the real situations. If the torpedo trajectory path changes, position between the torpedo and target would be different, as the target properties acquired would vary with the position [12], and then the probability of each object would change, thus affect the trajectory direction in the next period. The torpedo can automatically obtain the target features, and the probability of the current target belonging to the actual target, and then adjust the trajectory. At the beginning of the simulation, the maximum probability would be assigned to a target randomly. Subsequently the probability of each target would be got by real-time position according to [12]. The initial distance of the torpedo and actual target is 3000 m, assumed torpedo keeps a constant speed, and the longest simulation time is 500 shoots. The guidance success rate is defined as the probability torpedo could be guided to identify the real target position, during which the time cost is defined as the guidance time. As the identifying process is parallel with and attacking process, we use different trajectory guidance method [13], take the final attack success data to represent the final result. Under condition of one decoy and one real target, and the condition of three decoys and one real target, the same tracking and optimum seeking method, Monte Carlo simulation is conducted 500 times, the results are shown in Table 1.

As showed in table 1, in the case of a few or many targets, this method can significantly improve the success rate of the guide, but the lead time goes up under the strike time's permission. This is caused by that the method taking time to account for the target features detection and the target affirmed, so the longer ballistic makes the time increases. Synthesis simulation tests show that this method can realize the effect that the method would not lose the biggest target, take into account more target detection, to recognize the targets more accurate and thus improve the hit accuracy effectively.

Table 1 500 times Monte Carlo simulation results in system environment

	Two targets		Four targets	
	Guidance success rate	Guidance time (time shot)	Guidance success rate	Guidance time (time shot)
Back tracking method [13]	54.33%	182	31.92%	439
Fixed advance angle method [13]	59.24%	162	36.21%	422
Method of this paper	92.88%	198	89.62%	283

7 Conclusions

In this paper, we analyzed the environment that when the real target and decoys are existing, how to improve the recognition accuracy. We proposed the method which takes the tracking, trajectory and optimum seeking data join. All data from one process can be used in other processes. When we do the tracking process, the method which use the length feature for tracking has been given out; when we do the optimum seeking, the method which use the acoustic feature and motion information has been proposed; and the trajectory design method based on APFM have taken the optimum seeking and tracking requirement. Simulation results demonstrate that the method can improve the accuracy and of target recognition with a little more time, and then improve the accuracy of the strike. So, reducing the amount of computation for further engineering and the semi physical engineering test should be done at the next.

Acknowledgment This research was financially supported by the national natural science foundation of China (61135001), the Xi'an (China) Science and Technology Project in 2014 (CXY1436 (9)).

References

1. Li, X., et al.: Classification of underwater target echoes based on auditory perception characteristics. J. Marine Sci. Appl **13**, 218–224 (2014)
2. Kean, C., et al.: Subjective evaluation experiments of timbre attribute based acoustic target identification. Chinese Sci. Bull. **55**, 651–659 (2010)
3. Na, W., Kean, C.: Application of subband spectral centroid features to recognizing underwater targets. Acta Armamentarii **30**, 144–149 (2009)
4. Song, Z., et al.: The method of underwater target recognition based on LOFAR spectrum. Journal of Naval Aeronautical Engineering Institute **26**, 283–286 (2011)
5. Deng, K., Xiang, X., Gu, Z.: Multi-highlight model of scaling acoustic decoy. Technical Acoustics **30**, 201–205 (2011)

6. Liu, C., Fu, Z., Wang, M.: Underwater target identification based on the methods of bearing and cross-spectrum. Acta Armamentarii **27**, 933–935 (2006)
7. Ning, X., Zhao, X.M., Cheng, J.T.: A way to Identify Targets Based on Dimension Spectrum and SVM. Control & Automation **24**, 196–197 (2008)
8. Liang, K.-H., Wang, K.-M.: Using simulation and evolutionary algorithms to evaluate the design of mix strategies of decoy and jammers in anti-torpedo tactics. In: Proceedings of the 2006 Winter Simulation Conference, Monterey, CA, pp. 1299–1306. IEEE (2006)
9. Tu, C.-K., et al.: Applying genetic algorithms on fuzzy logic system for underwater acoustic signal recognition. In: Proceedings of the 2000 International Symposium on Underwater Technology, pp. 405–410 (2000)
10. Tucker, S., Brown, G.J.: Classification of transient sonar sounds using perceptually motivated features. IEEE Journal of Ocean Engineering **30**, 588–600 (2005)
11. Young, V.W., Hines, P.C.: Perception-based automatic classification of impulsive-source active sonar echoes. Journal of the Acoustical Society of America **122**, 1502–1517 (2007)
12. Yu, L., et al.: Underwater acoustic multi-target recognition algorithm based on hierarchical information fusion structure. In: 17th International Conference on Information Fusion, Salamanca, Spain (2014)
13. Chen, C., et al.: Anti-torpedo technology. National Defense Industry Press, Beijing (2006)

Recognizing Unseen Gym Activities from Streaming Data - Accelerometer Vs. Electromyogram

Heli Koskimäki and Pekka Siirtola

Abstract Wearable sensors based activity recognition is a research area where mostly inertial measurement unit based information is used to recognize human activities. Commonly the approaches are based on accelerometer data while in this study the potential of electromyogram signals in activity recognition is studied. The actual research problem tackled is one of the major drawbacks in activity recognition, namely to add completely new activities in real life to the recognition models. In this study, it was shown that in gym settings electromyogram signals clearly outperforms the accelerometer data in recognition of completely new sets of gym movements from streaming data even though the sensors would not be positioned directly to the muscles trained.

Keywords Activity recognition · Wearable sensors · Electromyogram · Data mining

1 Introduction

During the last decade, many practical constraints related to carry-on sensors like accelerometers, magnetometers and GPS-receivers have been solved. This has enabled monitoring and classification of human activity based on information from wearable sensors to become a growing research area of pattern recognition and machine learning. The overall activity recognition process includes a data set collected from the activities wanted to be recognized, preprocessing, segmentation, feature extraction and selection, and classification [3]. Activity recognition is used in recognizing,

H. Koskimäki(✉) · P. Siirtola
Biomimetics and Intelligent Systems Group, University of Oulu, PO Box 4500,
90014 Oulu, Finland
e-mail: {heli.koskimaki,pekka.siirtola}@ee.oulu.fi

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_21

for example, daily activities [2, 18], in sport sector [4, 16] and in monitoring of assembly tasks [10, 17].

One of the problems of activity recognition is that to recognize n activities, training data must be collected from at least n-1 activities [15]. The remaining activity could be recognized based on the assumption that if the performed activity was not recognized as one of the n-1 from which training data was collected, it must be the one from which training data was not available. Nevertheless, in practice the streaming data consists also plenty of data not interesting from application specific point of view, and that cannot be collected inclusively. This so called as null-data or "other activities" makes the decision if there actually is a novel activity or should it be considered to belong null-class a challenging task.

Thus in this article the problem studied for unseen activities is that how to recognize them as activities instead of belonging to the null-class. Moreover, in this study, a new sensor is introduced to be used to solve the problem in gym setting. The gym activity recognition makes a quite unique problem into the activity recognition area while the gym exercises mostly consists of repetitive movements. How to recognize different gym activities based on acceleration sensors have been studied, for example, in [4, 12, 13]. In [4] there were no null-data collected thus making the research simpler but in [12, 13] both a segmentation approach was used as a solution to decide beforehand if gym activity is performed against the null-data. Nevertheless, in both cases the segmentation is optimized based on the existing activities (the leave-one out approach is used as person independent approach) and there are no information of the generalization of the segmentation to novel gym sets. Moreover, the few studies considering the unseen activities are also completely different to ours. For example, in [5], they are concentrating to recognize the actual gym exercises based on semantic attributes (e.g. dumbbell curl consist of arm down and arm curl actions) and there are no null-data in the study.

On the other hand, electromyogram (EMG) is used to measures muscles to see the power needed to perform certain gym exercises [8]. Nevertheless, to be able to do that EMG device has to be positioned directly on the muscle to be measured. Thus although it could sound trivial to use EMG to recognize the actual gym exercises from the other activities the approach where sensors are not positioned to the actual trained muscle or changed between the exercises makes the study novel. While the EMG-sensors are attached in the forearm of the user in this study the movement of individual fingers also effect to the tension of the forearm muscles making the approach more challenging.

2 Sensors and Data Collection

The data were collected using a Myo Armband [1]. Myo includes 8 EMG sensors and a nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer (Figure 1). It is developed for gesture recognition purposes and thus meant to be worn in a forearm of the user. In our study, the Myo was located at

Table 1 Gym exercises, more details can be found from [11].

Muscle group	Exercises
Triceps	Close-Grip Barbell Bench Press, Bar Skullcrusher, Triceps Pushdown, Bench Dip / Dip, Overhead Triceps Extension, Tricep Dumbbell Kickback
Biceps	Spider Curl Dumbbell Alternate Bicep Curl, Incline Hammer Curl, Concentration Curl, Cable Curl, Hammer Curl
Shoulders	Upright Barbell Row, Side Lateral Raise, Front Dumbbell Raise, Seated Dumbbell Shoulder Press, Car Drivers, Lying Rear Delt Raise
Chest	Bench Press, Incline Dumbbell Flyes, Incline Dumbbell Press, Dumbbell Flyes, Pushups, Leverage Chest Press
Back / lats	Seated Cable Rows, One-Arm Dumbbell Row, Wide-Grip Pulldown Behind The Neck, Bent Over Barbell Row, Reverse Grip Bent-Over Row, Wide-Grip Front Pulldown

the right forearm positioned so that the IMU was on the top of the forearm while the EMG sensors located evenly distributed around the arm. In this study the frequency of 50 Hz were used in data collection.

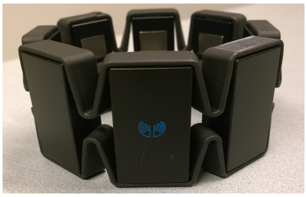


Fig. 1 Myo Armband

The actual data were collected from 10 persons and from 30 different gym exercises, each of them consisting a set of ten repetitions. The exercises were mostly done using free weights, and for every upper body muscle group, data from six different exercises were collected (Table 1). While the data set was gathered as a continuous signal, the data set constituted also data between every exercise set in which the subject moved around at the gym, changed weights, stretched or just stayed still (null-data). Altogether, there were more than 11 hours of data of which 77 percent was considered as null-data.

Table 2 Features calculated from acceleration data, EMG signals (channels summed with adjacent channels (EMG), or channels summed altogether (EMG sum))

Data set	Feature type	Features
Acc	Statistical features	std, mean, min, max, median, percentiles (5, 10, 25, 75, 90, 95), zero and mean crossing
	Frequency domain	FFT sums (1:2, 1:5, 6:10, 10:15), squared sum using all channels
	Haar wavelets	sums of wavelet decompositions using different bookkeeping vectors
	correlation	autocorrelation and cross-correlation
EMG	Statistical features	std, mean, min, max, median, percentiles (5, 10, 25, 75, 90, 95)
	Sums	sums of data value over 25, 50, 100, 150 and 200
EMG sum	Statistical features	std, mean, min, max, median, percentiles (5, 10, 25, 75, 90, 95), zero and mean crossings
	correlation	autocorrelation

3 Methods

The EMG signals were pre-processed with two different ways: 1) all the eight EMG signals were summed up as a single signal, or 2) different channels were summed with the values of adjacent EMG signals (the EMG signal 1 consisted of sum of signals 8, 1 and 2 and signal 2 of signals 1, 2 and 3, etc.). For acceleration signals, no pre-processing was done.

After the pre-processing the continuously measures signals were divided into segments using the sliding window method, where window length of two seconds with a slide of 0.5 seconds between two sequential windows was used. For every of the windows, features were calculated including statistical values for all the signals and for acceleration also frequency domain and correlation features were calculated. The amount of features for acceleration signals were 219, for 8 channels of EMG 128 and for summed EMG channel 19.

In this article, the best features were chosen using sequential forward selection (SFS) and minimum Redundancy Maximum Relevance Feature Selection (mRMR). With SFS the best features were selected one at a time using the classification accuracy of the model in question as a selection criteria [6]. However, the selection was not stopped at local minimum but it was allowed to choose until "the best features" included all the features. On the other hand, with mRMR the feature selection was done model independently by selecting features having the highest correlation to the classification variable but locating far from each other [14]. With mRMR the amount of features was decided before hand as signal-wise based on a preliminary test with all the data.

The classifiers used in this study were the parametric linear discriminant analysis (LDA), quadratic discriminant analysis (QDA). The LDA and QDA model the class-conditional densities parametrically as multivariate normals [7]. In practice, QDA separates classes using nonlinear decision boundaries while LDA uses linear decision boundaries. Both of the methods are fast to train, easy to implement and the memory requirements are small thus making them well-liked in practical applications and devices. Moreover, in practical activity recognition applications the simplest methods can outperform the more sophisticated methods [9].

4 Results

The model generalization to new exercises were studied by selecting suitable subsets of activities into training and testing under leave-one-person-out cross-validation schema. But instead of the traditional version where a single activity is deleted at a time the deletion is done muscle-group specifically in four scenarios.

In practice this means that for every person at a time, in scenario 1, every set of exercises (6 exercises) at a time and the null-data were used as testing data while the other 4 sets of exercises (24 exercises) and the null-data from the remaining 9 persons were used for training (see Figure 2). In scenario 2, the same procedure was done by using two sets (12 exercises) for testing and three sets (18 exercises) for training, in scenario 3 using three sets (18 exercises) for testing and two (12 exercises) for training and in scenario 4 using four sets (24 exercises) for testing and one (6 exercises) for training. Thus the classification becomes more and more difficult between scenarios. In every scenario, all the combinations are gone through and the results are shown as an average of every person and of those combinations (6, 10, 10 and 6 combinations, respectively). Moreover, the average is shown as an average of class-wise averages preventing the massive amount of null-data to skew the results.

The results in Table 3 clearly show that the accuracies achieved with mRMR feature selection method are remarkably different from the SFS results. The only accuracy staying over 80 percent through the four scenarios is the accuracy achieved when using features calculated from the summed EMG-signal. Naturally, the reason for that is that there were not so many features to be selected (19 original features). Nevertheless, when using the summed EMG-signal and QDA, over 83 percent accuracies were achieved even when only movements targeted to single muscle groups were used as training data (scenario 4) which is over 20 percentage units higher than the accuracy achieved using acceleration signal (62.2%).

On the other hand, when considering the results achieved with SFS feature selection a more higher accuracies overall can be seen. The first obvious remark also with this case is that EMG signals contained more generalizable information than the acceleration signals. From the scenario 1 to scenario 4 only 0.6 percentage units drop was shown while within the acceleration signals a drop of 6 percentage units is seen between the scenarios, in addition to the 3 percentage units lower accuracy already in the first scenario (LDA). Moreover, by combining the acceleration information with EMG-information, it can be seen that no remarkable improvement in overall accuracies is achieved at least in the scenarios 3 and 4.

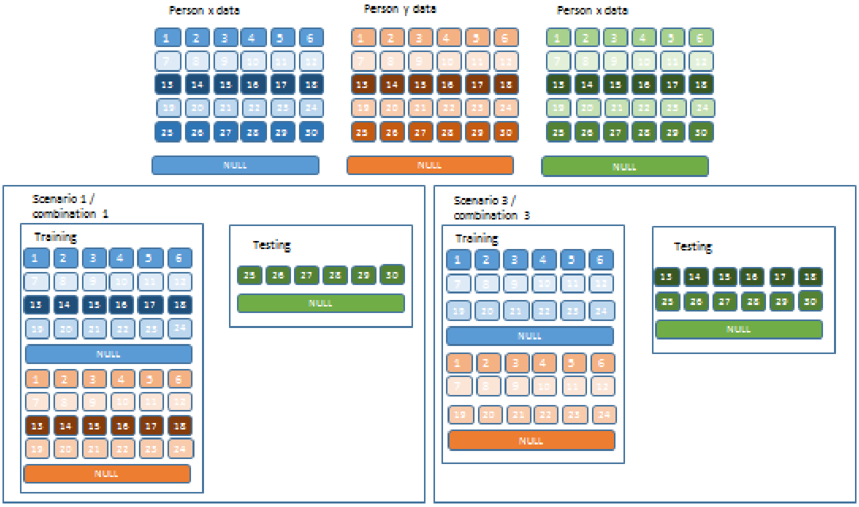


Fig. 2 Two scenario examples for case of data from three persons. The training data includes data from the persons as well as gym activities not used for testing. In the study all the combinations are went through.

5 Discussion

From the results it was clearly seen that the EMG signals contained more generalizable information than the acceleration signals. While the acceleration signals still coped the problem when there can be assumed to be at some level similar information in the training set, the more novel the new activity is the more difficult it is classified using the acceleration. This is quite surprising while the gym exercises contained sequential movements (repetitions) which are in acceleration signal based studies considered to separate the activity from the null-data. Nevertheless, as stated before, in previous studies the optimization of segmentation is based on the known activities which can affect to the results.

From the feature selection point of view an interesting remark was that the mRMR feature selection itself had a notable negative effect on the generalization results. This can be explained that with the SFS the features were selected based on results achieved for the testing data, in practice, telling the feature selection method that we do not want to optimize the training data classification but the testing data classification. For mRMR no information of the actual problem was introduced. Nevertheless, it has been already show that the recognition rates are biased in SFS while the same data is used for selecting the features and validating the features [9]. Thus by using SFS the unseen activities are not unseen but already used in model optimization. Although the difference between the accuracy of EMG and acceleration signals with SFS is so

Table 3 Average recognition rates using mRMR and SFS feature selection methods with both LDA and QDA classifiers using acceleration data, EMG signals (channels summed with adjacent channels (EMG), or channels summed altogether (EMG sum)), or a combination of the signals.

Feature selection	Classifier	Scenario	Signal				
			ACC	EMG	EMG sum	ACC+EMG	ACC+EMG sum
mRMR	LDA	1	77.7	85.2	82.7	83.2	**84.2**
		2	74.8	81.1	82.3	81.8	82.6
		3	69.9	76.2	81.9	76.3	77.8
		4	62.2	70.4	81.3	70.3	71.6
	QDA	1	68.6	77.0	84.0	81.4	83.4
		2	66.5	77.2	**84.0**	80.8	81.3
		3	64.0	78.5	**83.8**	75.2	76.6
		4	59.7	77.5	**83.2**	70.1	65.2
SFS	LDA	1	85.2	88.2	83.0	89.9	88.1
		2	83.1	88.2	83.1	89.5	87.7
		3	82.3	88.0	83.1	**88.6**	87.0
		4	79.3	87.7	83.0	**88.2**	85.7
	QDA	1	84.7	87.8	85.3	**90.1**	89.8
		2	83.9	87.8	85.2	**89.6**	89.1
		3	83.9	87.8	85.1	88.5	87.5
		4	77.5	87.5	84.7	87.8	86.5

apparent that it cannot be caused by this bias, the more reliable overall results are those achieved with mRMR which clearly favored EMG-data.

On the other hand, in this study, all the activities were targeted to upper body muscles which still leaves the question "how the results generalize in the cases of lower body muscles workouts" open. For example, there are lower body muscles targeted gym equipments where hands are positioned into stationary handles causing the acceleration to fall behind. Nevertheless, interesting would be seen, if the adherence of the handles would be enough to EMG-signals to contain the information of exercise time.

6 Conclusions

In this article, the generalization of acceleration signals information was compared with EMG signals in novel events at gym activity recognition. It was shown that even non-optimally positioned EMG-sensor will outperform the accelerometer information; the most dissimilar new activities can be extracted from null-data with 10 to 20 percentage unit higher accuracy by using EMG signal. Naturally, more accurate results could be achieved by using optimally located EMG sensor but this was considered non-practical in real world usage while the end-user cannot be obligated to change the sensor location between every gym set.

While in this study, the approach covered mostly the segmentation part of activity recognition, in future work, the actual gym activity recognition aspects using EMG signals will be studied.

References

1. Myo. https://www.myo.com/ (accessed: October 30, 2015)
2. Banos, O., Damas, M., Pomares, H., Prieto, A., Rojas, I.: Daily living activity recognition based on statistical feature quality group selection. Expert Systems with Applications **39**(9), 8013–8021 (2012)
3. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv. **46**(3), 33:1–33:33 (2014). http://doi.acm.org/10.1145/2499621
4. Chang, K., Chen, M., Canny, J.: Tracking free-weight exercises. In: UbiComp 2007: Ubiquitous Computing, pp. 19–37 (2007)
5. Cheng, H., Sun, F., Griss, M., Davis, P., Li, J., You, D.: Nuactiv: recognizing unseen new activities using semantic attribute-based learning. In: Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2013, pp. 361–374. ACM, New York (2013). http://doi.acm.org/10.1145/2462456.2464438
6. Devijver, P.A., Kittler, J.: Pattern recognition: A statistical approach, vol. 761. Prentice-Hall, London (1982)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2012)
8. Holviala, J., Kraemer, W., Sillanpää, E., Karppinen, H., Avela, J., Kauhanen, A., Häkkinen, A., Häkkinen, K.: Effects of strength, endurance and combined training on muscle strength, walking speed and dynamic balance in aging men. European Journal of Applied Physiology **112**(4), 1335–1347 (2012). http://dx.doi.org/10.1007/s00421-011-2089-7
9. Koskimäki, H.: Avoiding bias in classification accuracy - a case study for activity recognition. In: IEEE Symposium on Computational Intelligence and Data Mining (2015) (accepted)
10. Koskimäki, H., Huikari, V., Siirtola, P., Laurinen, P., Röning, J.: Activity recognition using a wrist-worn inertial measurement unit: a case study for industrial assembly lines. In: The 17th Mediterranean Conference on Control and Automation, pp. 401–405 (2009)
11. Koskimäki, H., Siirtola, P.: Recognizing gym exercises using acceleration data from wearable sensors. In: 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 321–328. IEEE (2014)
12. Morris, D., Saponas, T., Guillory, A., Kelner, I.: Recofit: using a wearable sensor to find, recognize, and count repetitive exercises. In: Proceedings of ACM CHI (2014)
13. Muehlbauer, M., Bahle, G., Lukowicz, P.: What can an arm holster worn smart phone do for activity recognition?. In: 15th Annual International Symposium on Wearable Computers (ISWC), pp. 79 – 82 (2011)
14. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8), 1226–1238 (2005)
15. Siirtola, P.: Recognizing human activities based on wearable inertial measurements: methods and applications. Doctoral dissertation, Department of Computer Science and Engineering, University of Oulu (Acta Univ Oul C 524) (2015)
16. Siirtola, P., Koskimäki, H., Huikari, V., Laurinen, P., Röning, J.: Improving the classification accuracy of streaming data using sax similarity features. Pattern Recognition Letters **32**(13), 1659–1668 (2011)
17. Stiefmeier, T., Roggen, D., Tröster, G., Ogris, G., Lukowicz, P.: Wearable activity tracking in car manufacturing. IEEE Pervasive Computing **7**(2), 42–50 (2008)
18. Zhang, M., Sawchuk, A.A.: Human daily activity recognition with sparse representation using wearable sensors. IEEE Journal of Biomedical and Health Informatics **17**(3), 553–560 (2013)

A Comparison of the YC_BC_R Color Space with Gray Scale for Face Recognition for Surveillance Applications

Jamal Ahmad Dargham, Ali Chekima, Ervin Gubin Moung and Segiru Omatu

Abstract Face recognition is an important biometric method because of its potential applications in many fields, such as access control and surveillance. In this paper, the performance of the individual channels from the YCBCR color space on face recognition for surveillance applications is investigated and compared with the performance of the gray scale. In addition, the performance of fusing two or more color channels is also compared with that of the gray scale. Three cases with different number of training images per persons were used as a test bed. It was found out that, the gray scale always outperforms the individual channel. However, the fusion of CBxCR with any other channel outperforms the gray scale when three images of the same class from the same database are used for training. The CBxCR channel gave the best performance for the individual color channels followed by CB, CB-CR, CB/CR and CR respectively. It was also found that, in general, increasing the number of fused channels in-creases the performance of the system.

Keywords Principal Component Analysis · YCbCr color space · Face recognition · Surveillance applications

1 Introduction

Face detection and face recognition are growing biometric research fields because of their potential applications as important tools for many applications such as

J.A. Dargham(✉) · A. Chekima · E.G. Moung
Fakulti Kejuruteraan, Universiti Malaysia Sabah, Jalan UMS, 88400
Kota Kinabalu, Sabah, Malaysia
e-mail: {jamalad,chekima}@ums.edu.my, menirva.com@gmail.com

S. Omatu
Faculty of Engineering, Department of Electronics, Information and Communication Engineering, Osaka Institute of Technology,
5-16-1, Omiya, Asahi-ku, Osaka 535-8585, Japan
e-mail: omatu@rsh.oit.ac.jp

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_22

security surveillance and human-computer interaction. There are several reports that have investigated the effect of color spaces in biometric fields. Karimi and Devroye [1] reported that using color images with PCA-based face recognition system improves the recognition accuracy. Gonzalez et. al. [2] investigated the effect of color spaces for face detection application using RGB, CMY, YUV, YIQ, YP_BP_R, YC_BC_R, YC_GC_R, YD_BD_R, HSV— or HSI— and CIE-XYZ. They tested 15 colored frontal face images of different people obtained from the AR face database. They found out that using the color information can improve the accuracy of the biometric system and reported that HSV color spaces is the best one for skin detection with a success rate of 95.06%. Yoo et. al. [3] investigated the effect of color spaces (RGB, HSV, YC_BC_R, and YC_GC_R) for face recognition. They reported that the YC_BC_R color space gives the best performance (92.3% correct recognition) when tested on the color FERET database using PCA-based face recognition method. Chelali et. al. [4] proposed a PCA-based face recognition system that uses the segmented skin region of the face as the features for their system. Three color spaces were investigated for skin region segmentation; RGB, HSV, and YC_BC_R. They used three databases for their experiment, two in a controlled environment with the same illumination and background (Computer Vision Center database and LCPTS Facial database), and the third database (CALTECH database) in an uncontrolled environment that present a variation in illumination, pose and background. The best reported recognition rate (96%) is obtained when the YC_BC_R color space is used. As can be seen from the review, there are evidence that using color can enhance face recognition systems and that most researchers reported that the YC_BC_R color space gives better recognition rate than other color spaces. Since, we have used gray scale images obtained from color images in our previous work [6], in this paper a comparison of the performance of the YC_BC_R color space with gray scale for face recognition for surveillance application is carried out. In addition, an investigation of which channel or combination of channels from the YC_BC_R color space gives the best performance is carried out.

2 Surveillance Dataset and Database Preparation

The ChokePoint Dataset [5] is one of the freely available surveillance databases under real-world surveillance conditions. It is a collection of surveillance videos of 25 persons in portal 1 and 29 persons in portal 2. Portal 2 dataset was used for database preparation since it has more subjects compared to portal 1. Portal 2 dataset consists of two sub datasets; Portal 2 Entering scene dataset (P2E) and Portal 2 Leaving scene dataset (P2L). P2L dataset contains better quality frontal faces. Thus, P2L dataset were selected for database preparation for all experiments. Although the video images were captured in four sessions, only session 1 (S1), session 2 (S2), and session 3 (S3) were used. Session 4 video images were not used because the captured image of the face is not frontal as seen in Fig. 1(a). The ChokePoint Dataset provided the eyes coordinates for their color video images. All the frontal face images are cropped according to their given eyes coordinate and the eyes are aligned so that their eyes are in the same horizontal line. Fig. 1(b) shows the face cropping template. Assuming E_L is the length between the right and left eye and M is the middle point between the right and left eye, then the length from the right eye to R, and from M to T, as well as from the left eye to L, is equal to E_L and the length from M to B is $2E_L$.

In our previous work [6], the gray scale database provided by the ChokePoint Dataset was used while in this paper, the color video images of the ChokePoint Dataset are used. The video frames were divided into three classes, namely; FAR, MEDIUM, and CLOSE. Each person has 18 frames per class. The first 18 frames from the 54 frames for each person are assigned to the FAR class. The next 18 frames after the FAR class frames are assigned to the MEDIUM class and the last 18 frames are assigned to the CLOSE class. A total of 4536 cropped video images with frontal face of a single person were used. Three databases representing three different session containing 1512 video images each were created. They represent 28 different persons. 14 persons are used for training and recall testing, and the other 14 different persons are used for reject testing only. The image size of the gray scale frontal face that ChokePoint Dataset provided is 96 by 96 pixels. Thus, in this work, the cropped video images will be set to 96 by 96 pixels. The gray scale and YC_BC_R color space obtained from the color video images will be used in this work and the performance of both the gray scale and the YC_BC_R will be compared.

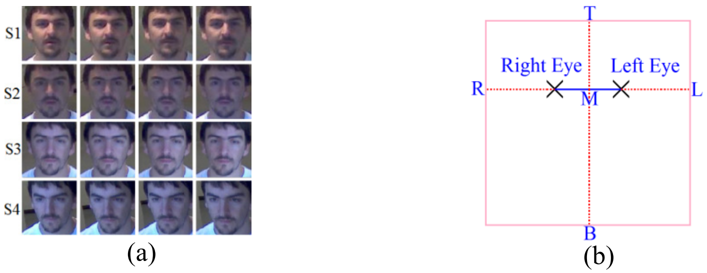


Fig. 1 (a) Samples of frontal face images in the P2L dataset, Source ChokePoint Dataset. (b) The face cropping template.

The color channels that are investigated are the C_B, C_R, C_B multiplied by C_R (C_BxC_R), the ratio of C_B over C_R (C_B/C_R), and the differences between C_B and C_R (C_B-C_R).

3 The Training and Testing Database

In our previous work [6], we have investigated the effect of training data selection on face recognition for surveillance application by using the provided gray scale video images of the ChokePoint Dataset as our data. Eight cases of selecting training images were proposed. It was found that, Case 1 gives the lowest performance while the Case 8 gives the best performance followed by Case 6. Thus, in this work Case 1, Case 6, and Case 8 selection methods will be used to create the training database. Table 1 shows the training database criteria while Table 2 shows the training database details for each case.

Table 1 The training database criteria

Case	Selection criteria
Case 1	Three images per person randomly selected from the same class and from the same database session will be used
Case 6	Nine images per person from all the database session with 1 mean image per class will be used.
Case 8	18 images per person; consist of nine images and nine mean images from all the database session with 1 randomly selected image and 1 mean image per class will be used. (Case 8 is Case 4 and Case 6 images combined together in the same folder)

Table 2 Training database details for case 1, 6, and 8.

Case	Training database size	Number of persons in training database	Number of images per person	Number of class per person	Number of images per class
1	42 images	14	3	1 class, 1 session	3
6	126 images	14	9	3 classes, all sessions	1
8	252 images	14	18	3 classes, all sessions	2

Two testing databases were created. The first database, Client test database, has 756 images of the 14 persons for each session. This database will be used to test the recall capability of the face recognition system. The second database, Imposter test database, also has 756 images of 14 different persons for each session. This database will be used to test the rejection capability of the system.

4 The Face Recognition System

The face recognition system used in this work is the Principal Component Analysis (PCA) [7][8] as shown in Fig. 2. It consists of an offline training phase and an online testing phase. During the training phase, the entire set of training images are projected, using PCA, to a lower dimensional feature space. This projection will produce sets of feature vectors having much smaller dimensions compared to the original image dimensions. The feature vectors, a lower dimensional representation of the grays cale or YC_BC_R are then stored in the training database. During the online recognition process, a test image is projected to a lower dimensional feature space using PCA. The feature vectors representing the test image are then compared with the feature vectors from the training database in the matching process.

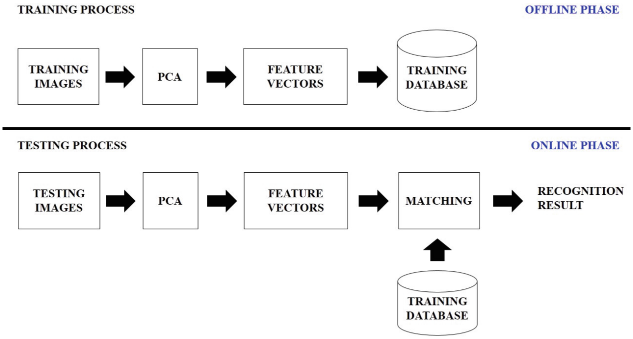


Fig. 2 Block diagram of the PCA based face recognition system.

5 The Training Phase

For the training and testing phase, the Euclidean distance, a popular distance measurement in biometric, is used to calculate the distance between two vectors. In order to do the matching task during the testing phase, a formula generating a threshold value using the training database, denoted as, t, is proposed. Let f be the feature vector from the training database, then, the threshold value, t, is the largest calculated Euclidean distance between any two feature vectors from the training database, denote as $max\{||f_M - f_N||\}$, divided by a tunable parameter denoted as *Tcpara* as shown in equation (1).

$$t = \frac{max\{\| f_M - f_N \|\}}{Tcpara} \tag{1}$$

where M, N = 1 to Z. M, N is the feature vector index in the training database and Z is the size of the training database. The threshold value, t, will be used during the testing phase where it will be compared with a Euclidean distance value generated by the test person image during the matching task.

6 The Testing Phase

Let i = 1, 2, ... 42 and M = 1, 2, ... Z and Z is the size of the training database. Let $T_R = \{f_{R1}, f_{R2}, ..., f_{R42}\}$ be the test person that have 42 test frames, f_{Ri} is a feature vector that represents the test person frame, and f_M is a feature vector from the training database, the minimum Euclidean distance between every test feature vector, f_{Ri} , with all the training database feature vector f_M , is calculated and denoted as $E = \{e_1, e_2, ... e_{42}\}$ where E is a set that contain all the 42 minimum Euclidean distances and e_i is the minimum Euclidean distance calculated between f_{Ri} and f_M. The person P from the training database that has the minimum Euclidean distance e_i with the test feature vector f_{Ri}, is denoted as l_i and the collection of l_i is

$L = \{l_1, l_2, \ldots l_{42}\}$. Let P_{mode} = mode$\{L\}$, be the highest occurrence from the set L, then the euclidean distance average, denoted as E_{av}, of all the e_i from E with l_i equal to P_{mode}, were calculated. If the average Euclidean distance between test person T_R and P person in the training database, $E_{av}(T_R,P)$ is smaller than a given threshold t, then test person T_R and P person in the training database are assumed to be of the same person. To measure the performance of both systems, several performance metrics are used. For a **Recall** test, if a test person ***Ti*** is correctly matched to the same person ***Pi*** in the training database, it is a **Correct Recall**. If a test person ***Ti*** is incorrectly matched with person ***Pj***, where ***i*** and ***j*** are not the same person, it is a **False Acceptance**. If a test person ***Ti*** is of a person ***Pi*** in the training database but rejected by the system, it is a **False Rejection**. For a **Reject** test, if a test person ***Ti***, from the Imposter test database is rejected by the face recognition system, it is a Correct Reject. If a test person ***Ti*** from the Imposter test database is accepted by the program, it is a **False Acceptance**. Let NCR = number of correct recall, NCJ = number of correct reject, and NT = number of test, then **Correct Recall Rate** = NCR/NT and **Correct Reject Rate** = NCJ/NT. For this work, the threshold tuning parameter, *Tcpara*, was set so that the system has equal correct rates for both recall and reject. This classification rate is defined as the **Equal Correct Rate (ECR)**.

7 Data Fusion Strategy

The fusion strategy fused the outputs from two or more face recognition systems. The logic operator chosen for the data fusion strategy is the OR operator. For the recall test, if at least one system give a correct matching and the other system(s) give wrong matching or not found, then a correct match is found. If not, the fusion system will give a no match is found. For the reject test, if at least one system give a correct reject and the other system(s) give false acceptance, then the fusion system give a correct reject. If not, the fusion system will give a false acceptance. The use of AND operator will make the fusion strategy become more strict, for example, for a recall test, all the individual system need to give a correct match found in order for the fusion system to give a correct match found. Thus, the OR operator were selected since it require at least one system to give a TRUE statement for the fusion system to give TRUE statement too.

8 Results and Discussion

Each experiment for each of the cases is carried out 10 times and the average of the recognition rate from the 10 experiment runs is used for comparison. Fig. 3 shows the performance of the gray scale with individual channels of the YC_BC_R. It can be seen, from Fig. 3, that regardless of the cases, the gray scale always outperform the other five YC_BC_R channels, giving 76.27%, 100%, and 100% rate in case 1, 6, and 8 respectively. However, the C_BxC_R channel outperforms the other channels. From the results of case 1, it can be seen that the C_B, C_R, C_B/C_R, and C_B-C_R channels are not suitable when only three images from one class are used for training as all of them give an equal error rate below 50%. As can be seen from Fig. 3, the performance of all channels improves as the number of images per person increases from 3 images in case 1) to 9 images in case 2 and then to 18 images in case 3.

The fusions of several channels from the YC_BC_R color space are also carried out to investigate whether or not the fusion strategy can compete with the gray scale performance. The proposed fusion strategies are shown in Table 3. The average rate of the recall and reject rates given by the fusion strategy is used for comparison and is shown in Fig. 4.

As can be seem from Fig. 3 and Fig. 4, the fusion of two or more channels always gives better performance than the individual channels regardless of the number of images used for training or the data set from which they are selected. As can be seen from Fig. 4, the fusion of the two best channels slightly outperforms the gray scale for case 1. In addition, as the number of fused channels increases so does the performance of the system albeit not at the same rate.

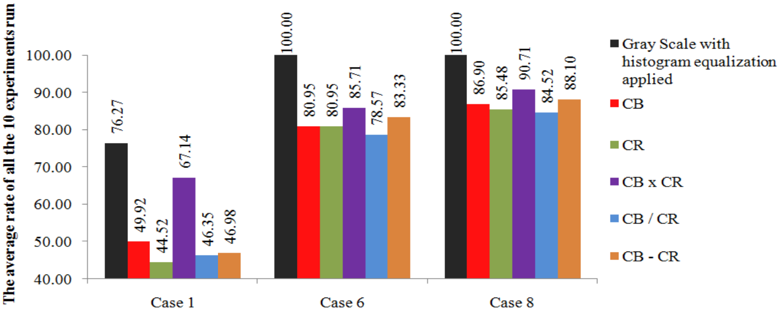


Fig. 3 The average equal correct rate for the individual color channels compared to that of the gray scale.

Table 3 The proposed fusion strategy

	Case 1	Case 6	Case 8
Fusion of the two highest rate	C_B, C_BxC_R	C_BxC_R, C_B-C_R	C_BxC_R, C_B-C_R
Fusion of the three lowest rate	C_R, C_B/C_R, C_B-C_R	C_B, C_R, C_B/C_R	C_B, C_R, C_B/C_R
Fusion of the three highest rate	C_B, C_BxC_R, C_B-C_R	C_B, C_BxC_R, C_B-C_R	C_B, C_BxC_R, C_B-C_R
Fusion of the one highest rate and two lowest rate	C_R, C_BxC_R, C_B/C_R	C_R, C_BxC_R, C_B/C_R	C_R, C_BxC_R, C_B/C_R
Fusion of all the five channels	All channels	All channels	All channels

It can also be seen that the fusion of channels that individually gives good performance will give higher performance that fusing those with lower performance. This suggests that the performance of the individual channel is an important criterion when selecting channels for fusion. In Case 6 and 8 however, the gray scale outperforms all the proposed fusion strategies. This shows that using one image and one mean image from each class from all the database sessions gives the best performance in term of accuracy and process time since converting RGB to gray scale is faster than converting RGB to YC_BC_R.

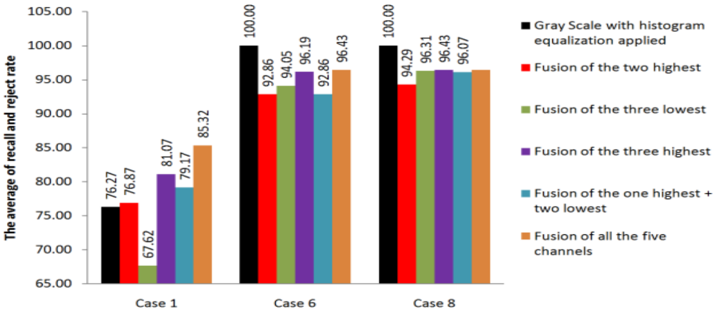


Fig. 4 The recognition performance of the data fusion

The entire results of the individual channels and their fusions, shown in Fig. 3 and Fig. 4, are new results which were not reported in our previous work [6]. However, the results of the gray scale shown in Fig. 3 and in [6] are for the same cases. The reason that they had different recognition rates is because of the different training data set used. Nevertheless, the same performance pattern is shown in both works where Case 1 gives the lowest performance while Case 8 gives the best performance.

9 Conclusion

In this work, an investigation of the performance of different channels of the YC_BC_R color space on face recognition for surveillance applications has been carried out. It was found that, the gray scale gives better performance than any of the individual color channel regardless of the cases used. However, the fusion of C_BxC_R with any other channel outperforms the gray scale when three images of the same class from the same database were used for training (Case 1). For the YC_BC_R color space, the best performance is achieved by using the C_BxC_R channel. The C_B, C_R, C_B/C_R, and C_B-C_R are not suitable to be used when the number of training images per person is small. It was also found that if the differences between individual channels performance vary significantly, the individual channel performance become an important criteria when selecting channels for fusion. In our further work, we intend to investigate the reason why some channels obtain much better results that the other channels or combinations.

References

1. Karimi, B., Devroye, L.: A study on significance of color in face recognition using several eigenface algorithms. In: Canadian Conference Electrical and Computer Engineering (CCECE), pp. 1309–1312 (2007)
2. Chaves-González, J.M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Detecting skin in face recognition systems A colour spaces study. Digital Signal Processing **20**(3), 806–823 (2010)
3. Yoo, S., Park, R., Sim, D.: Investigation of Color Spaces for Face Recognition. In: Proceedings of Machine Vision Application, pp. 106–109 (2007)
4. Chelali, F.Z., Cherabit, N., Djeradi, A.: Face recognition system using skin detection in RGB and YCbCr color space. In: 2nd World Symposium on Web Applications and Networking (WSWAN), pp. 1–7 (2015)
5. http://arma.sourceforge.net/chokepoint/, National ICT Australia Limited (2014)
6. Dargham, J., Chekima, A., Moung, E., Omatu, S.: The Effect of Training Data Selection on Face Recognition in Surveillance Application. Advances in Distributed Computing and Artificial Intelligence Journal **3**, 58–66 (2015)
7. Dargham, J., Chekima, A., Moung, E.: Fusion of PCA and LDA based face recognition system. In: International Conference on Software and Computer Applications, IPCSIT, vol. 41 (2012)
8. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)

A Study of Consensus Formation using Kinetic Theory

Stefania Monica and Federico Bergenti

Abstract In this paper consensus formation in multi-agent systems is studied using the framework of kinetic theory of gases. This can be done by interpreting agents as the molecules of a gas and interactions among agents as the collisions among the molecules. Each agent can interact with any other agent in the considered system and interactions are binary, i.e., they involve only two agents at a time. Under such hypothesis, starting from a microscopic model which describes the effects of interactions in a pair of agents, we derive, analytically, global properties of the system, such as the conservation of the average opinion. Simulation results show that the proposed kinetic approach allows reaching consensus and the velocity of convergence to consensus depends, as expected, on the parameters of the model.

Keywords Consensus formation · Kinetic theory · Multi-agent systems · Opinion dynamics

1 Introduction

Consensus formation is a typical problem related to distributed computing and cooperative multi-agent systems; it is also critical in social networks, where agents have been recently applied [1, 2]. Consensus, in particular, is an important problem in various fields such as: computer science, robotics, mathematics, physics, and biology [3]. When dealing with consensus problems, we are interested in finding interaction rules which lead to collective agreement, without involving objective functions. Various approaches have been used in the literature to describe the effects of interactions among agents and to derive consensus protocols, such as those based on cellular automata (e.g., [4]), on thermodynamics (e.g., [5]), and on Markov chains or Bayesian

S. Monica(✉) · F. Bergenti
Dipartimento di Matematica e Informatica, Università degli Studi di Parma,
Parco Area delle Scienze 53/A, 43124 Parma, Italy
e-mail: {stefania.monica,federico.bergenti}@unipr.it

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_23

networks (e.g., [6]). Gossip-based algorithms have been proposed in the field of distributed systems [6]. Moreover, graph theory and, in particular, graph laplacians and their spectral properties have been applied to deal with consensus issues [7].

In this paper, we model opinion as a continuous variable defined in a closed interval I and we describe the opinion changes due to interactions among agents according to a simple formula, which is meant to reproduce *positive influence* (or *compromise*), i.e., the tendency of agents to change their opinions towards those of others they interact with. The considered model is inspired by kinetic theory of gases: agents are interpreted as the molecules of gases and their interactions are studied as collisions among the molecules. We restrict our analysis to *binary interactions*, namely we assume that each interaction involves only two agents, which is a common assumption in kinetic theory. This assumption may seem limiting, but it is necessary to apply the kinetic-based approach and, hence, to obtain analytic results [8, 9]. The kinetic framework, and in particular the Boltzmann equation, are used to derive macroscopic properties of the system.

This paper is organized as follows. Section 2 models, analytically, the considered kinetic approach to opinion dynamics and consensus. Section 3 shows simulation results obtained with different values of the parameters of the model. Section 4 concludes the paper and outlines some ongoing work.

2 Leading to Consensus Through Kinetic Equations

According to kinetic theory, macroscopic properties of gases, such as pressure and temperature, can be derived from a microscopic analysis of the interactions among molecules. Similarly, macroscopic properties of multi-agent systems, such as the reach of consensus, can be studied starting from the effects of single interactions among agents. This can be done by associating the molecules of gases with agents and by reinterpreting collisions as interactions among agents. Since kinetic theory typically deals with binary collisions, which involve pairs of molecules, we assume that the interactions among agents in the considered system are binary. In order to study the opinion dynamics of the system and how and when consensus is reached, we assume the existence, at each time $t \geq 0$, of a distribution function of the opinion, denoted as $f(v, t)$, which evolves according to the Boltzmann equation, namely an integro-differential equation typically used to study the evolution of the distribution function in kinetic theory [10]. Opinion is represented as a continuous variable v defined in the compact interval $I = [-1, 1]$. According to this assumption, $v = \pm 1$ represent extremal opinions while values close to 0 represent moderate opinions.

We start by defining the (microscopic) model according to which the opinions of a pair of agents are updated after an interaction. More precisely, the following rule for opinion updates holds

$$\begin{cases} v' = v + \gamma P(v, w)(w - v) \\ w' = w + \gamma P(w, v)(v - w) \end{cases} \tag{1}$$

where: v and w are the opinions of the two agents before they interact; v' and w' are the opinions of the two agents after the interaction; γ is a deterministic parameter; $P(v, w)$ is a function of the opinions of the two agents related to positive influence. According to this model, the opinions of two interacting agents are updated by adding to the pre-interaction opinions an addend which is proportional to the difference between the pre-interaction opinions according to the parameter γ and the function P.

From (1) it can be easily derived that

$$v' + w' = v + w + \gamma \left[(P(v, w) - P(w, v)) (w - v) \right] \tag{2}$$

If P is a symmetric function of v and w, namely $P(v, w) = P(w, v)$, then (2) implies that opinion is conserved in each interaction, namely the sum of the post-interaction opinions equals the sum of the pre-interaction opinions.

Concerning, instead, the difference between the opinions of the two agents involved in an interaction, by subtracting the second equation from the first in (1), it follows that

$$v' - w' = \left[1 - \gamma \left(P(v, w) + P(w, v) \right) \right] (v - w). \tag{3}$$

The idea of compromise is respected if the difference between the post-interaction opinions is smaller than that of the pre-interaction opinions, namely if

$$|v' - w'| < |v - w|. \tag{4}$$

According to (3), this is true if and only if $|1 - \gamma \left(P(v, w) + P(w, v) \right)| < 1$, namely:

$$0 < \gamma \left(P(v, w) + P(w, v) \right) < 2. \tag{5}$$

Another idea that we aim at reproducing with this model is that the post-interaction opinion of an agent must be closer to its own pre-interaction opinion than to that of the agent it interacts with, namely

$$|v' - v| < |v' - w| \qquad \text{and} \qquad |w' - w| < |w' - v|. \tag{6}$$

From the first equation of (1), one can derive that

$$|v' - v| = |\gamma P(v, w)||w - v| \qquad |v' - w| = |1 - \gamma P(v, w)||w - v|$$

and, therefore, the first condition in (6) is satisfied if and only if

$$\gamma P(v, w) < \frac{1}{2}. \tag{7}$$

Similarly, the second condition in (6) is satisfied if and only if

$$\gamma P(w, v) < \frac{1}{2}. \tag{8}$$

Concerning the function P, different expressions have been proposed in the literature [11, 12]. In order to reproduce *bounded confidence*, namely the fact that each agent interacts only with those whose opinions are not very different from its own, we can set

$$P(v, w) = \begin{cases} 1 & \text{if } |v - w| < \Delta_P \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\Delta_P > 0$ is a constant related to the inclination of agents to interact with others on the basis of their opinions. According to this choice of P, positive influence can modify the opinions of the two interacting agents if and only if their difference is less than Δ_P [11]. For this reason, if $\Delta_P \simeq 0$ opinions can be modified after an interaction only if the pre-interaction opinions of the two agents are very close. At the opposite, if $\Delta_P > 2$ the opinions of two agents are always modified after their interaction, since the function P always equals 1.

In the following, we restrict our analysis to this last case and, therefore, we assume that $P(v, w) = 1$ in (1) for each possible value of v and w. With this choice of the function P, according to (2), opinion is conserved in each single interaction. Moreover, the compromise idea expressed in (5) becomes $0 < \gamma < 1$ and the inequalities in (7) and (8) can be written as $\gamma < 1/2$. In order to take into account both previous conditions, from now on we assume that

$$0 < \gamma < 1/2. \tag{10}$$

This choice of γ and P also guarantees that post-interaction opinions still to the interval I where opinions are defined. As a matter of fact, from (1) it follows

$$\begin{aligned} |v'| &= |(1-\gamma)v + \gamma w| \le (1-\gamma)|v| + \gamma|w| \le \max\{|v|, |w|\} \le 1 \\ |w'| &= |(1-\gamma)w + \gamma v| \le (1-\gamma)|w| + \gamma|v| \le \max\{|v|, |w|\} \le 1. \end{aligned}$$

As far as now, we focused on microscopic aspects of interactions between agent. We now aim at deriving macroscopic properties of the considered system, through the Boltzmann equation. As already observed, the kinetic approach relies on the definition of a distribution function $f(v, t)$ which represents the number of agents with opinion in $(v, v + \mathrm{d}v)$ at time $t \ge 0$. We remark that time is defined as a discrete variable and each instant corresponds to a single binary interaction. Such a function satisfies

$$\int_I f(v, t)\mathrm{d}v = N(t) \tag{11}$$

where $N(t)$ is the number of agents at time t. The temporal evolution of $f(v, t)$ is described according to the Boltzmann equation. In particular, we refer to the following (homogeneus) formulation of the Boltzmann equation

$$\frac{\partial f}{\partial t} = \mathcal{Q}(f, f)(v, t) \tag{12}$$

where $\mathcal{Q}$ is the *collisional operator* which describes the effects of interactions in a pair of agents. In order to simplify notation, in the following derivation of the explicit formulation of $\mathcal{Q}$ we omit the dependance of the distribution function f on time t, since only integration with respect to the variable v are considered.

In order to derive the explicit formulation of the collisional operator, let us define the *transition rate* $W(v, w, v^*, w^*)\mathrm{d}v^*\mathrm{d}w^*$ which represents the probability that after the interaction between two agents with opinions v and w, respectively, the opinion of the first agent is in $(v^*, v^* + \mathrm{d}v^*)$ and the opinion of the second agent is in $(w^*, w^* + \mathrm{d}w^*)$. We remark that v^* and w^* are dummy variables. The loss of agents in $(v, v + \mathrm{d}v)$ can then be written by multiplying the transition rate by $f(v)\,\mathrm{d}v$, which represents the number of agents in $(v, v + \mathrm{d}v)$, and by $f(w)\,\mathrm{d}w$, which represents the number of agents in $(w, w + \mathrm{d}w)$, and then integrating with respect to all variables except v, obtaining

$$\mathrm{d}v \int_I\int_I\int_I W(v, w, v^*, w^*) f(v) f(w)\mathrm{d}w\, \mathrm{d}v^*\mathrm{d}w^* = f(v)\mathrm{d}v \int_I \beta(v, w) f(w)\mathrm{d}w \tag{13}$$

where $\beta(v, w) = \int_I\int_I W(v, w, v^*, w^*)\mathrm{d}v^*\mathrm{d}w^*$ represents the probability of interaction between an agent of opinion v and an agent of opinion w. Similarly, the gain of individuals in $(v, v + \mathrm{d}v)$ can be written as

$$\begin{aligned}&\mathrm{d}v \int_I\int_I\int_I W(v_*, w_*, v, w) f(v_*) f(w_*)\mathrm{d}w_*\, \mathrm{d}v_*\mathrm{d}w \\ =&\mathrm{d}v \int_I\int_I \beta(v_*, w_*)\Pi(v_*, w_*, v) f(v_*) f(w_*)\mathrm{d}w_*\, \mathrm{d}v_*\end{aligned} \tag{14}$$

where $\Pi(v_*, w_*, v)$ is defined as $\int_I W(v_*, w_*, v, w)\mathrm{d}w/\beta(v_*, w_*)$ and it represents the probability density function that the first agent has opinion v after the interaction, when its pre-interaction opinion is v_* and the pre-interaction opinion of its peer is w_*. Since, in our hypothesis, the opinions after any interaction are deterministic, the function $\Pi(v_*, w_*, v)$ can be expressed as

$$\Pi(v_*, w_*, v) = \delta\,(v - v_* - \gamma(w_* - v_*)) \tag{15}$$

where $\delta(\cdot)$ is the Dirac's δ function. In the following, we assume that β is constant, namely that it does not depend on the opinions of the agents [13]. The velocity of variation of the number of individuals in $\mathrm{d}v$ at time t is defined as $\dfrac{\partial f}{\partial t}\mathrm{d}v$ and it is equal to the difference between the gain (14) and the loss (13) of individuals in $(v, v + \mathrm{d}v)$. Therefore, using (14) and (13) in (12), the Boltzmann equation can be written as

$$\begin{aligned}\frac{\partial f}{\partial t} &= \beta \int_I \int_I \delta\left(v - v_* - \gamma(w_* - v_*)\right) f(v_*) f(w_*) \mathrm{d}w_* \, \mathrm{d}v_* \\ &\quad - \beta f(v) \int_I f(w) \mathrm{d}w\end{aligned} \tag{16}$$

where the right-hand side represents the collisional operator.

3 Analytic Study and Simulations

In order to derive macroscopic properties of a system, we focus on the weak form of the Boltzmann equation, which is obtained by multiplying (12) by a proper test function $\phi(v)$ and by integrating with respect to the variable v. The weak form of the Boltzmann equation (12) can then be written as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_I f(v, t)\phi(v) \mathrm{d}v = \int_I \mathcal{Q}(f, f)(v, t)\phi(v) \mathrm{d}v \tag{17}$$

where the right-hand side is called weak form of the collisional operator $\mathcal{Q}$ with respect to the test function $\phi(v)$. Functions that satify (18) are denoted as *weak solutions* of the Boltzmann equation. The explicit formulation of the weak form of the Boltzmann equation can be derived by multiplying (16) by $\phi(v)$ and by integrating with respect to v, obtaining

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} \int_I f(v, t)\phi(v) \mathrm{d}v &= \beta \int_I \int_I f(v_*) f(w_*) \phi(v_* + \gamma(w_* - v_*)) \mathrm{d}w_* \, \mathrm{d}v_* \\ &\quad - \beta \int_I \int_I f(v) f(w) \phi(v) \mathrm{d}v \mathrm{d}w\end{aligned} \tag{18}$$

where in the first addend on the right-hand side we have used the fact that

$$\int_I \delta\left(v - v_* - \gamma(w_* - v_*)\right) \phi(v) \mathrm{d}v = \phi(v_* + \gamma(w_* - v_*)).$$

Using a proper change of variable, (18) can then be written as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_I f(v, t)\phi(v) \mathrm{d}v = \beta \int_I \int_I f(v) f(w) \left(\phi(v') - \phi(v)\right) \mathrm{d}v \, \mathrm{d}w \tag{19}$$

where, as in the first part of the paper, v' denotes the post-interaction opinion of the agent whose pre-interaction opinion was v.

We now consider some simple test functions in order to derive macroscopic properties of the considered model. First, we set $\phi(v) = 1$ in (19). Under this assumption, the difference $\phi(v') - \phi(v)$ inside the integrals in (19) is 0 so that

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_I f(v,t)\mathrm{d}v = 0. \tag{20}$$

From (11), this result corresponds to the conservation of the number of agents $N(t)$ in the system; this property is analogous to mass conservation in a gas.

Then, considering the function $\phi(v) = v$ one can define

$$u(t) = \frac{1}{N} \int_I f(v,t) v \, \mathrm{d}v \tag{21}$$

as the average value of the opinion distribution at time t. Using this definition in (19) and recalling (1) with $P(v,w) = 1$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} u(t) = \frac{\beta\gamma}{N} \int_I \int_I f(v) f(w)(w - v)\mathrm{d}v\mathrm{d}w. \tag{22}$$

The integral on the right-hand side of (22) is 0 for symmetry reasons. One can then conclude that the average opinion is conserved, since, from (22), the following equality is derived

$$\frac{\mathrm{d}}{\mathrm{d}t} u(t) = 0. \tag{23}$$

This property corresponds to the conservation of momentum in gases.

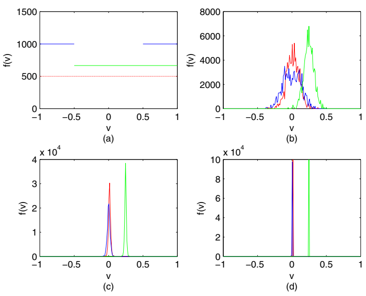


Fig. 1 The distribution functions $f(v,t)$ relative to $\gamma = 0.1$ and to: (a) $t = 0$; (b) $t = 10^4$; (c) $t = 2 \cdot 10^4$; (d) $t = 4 \cdot 10^4$.

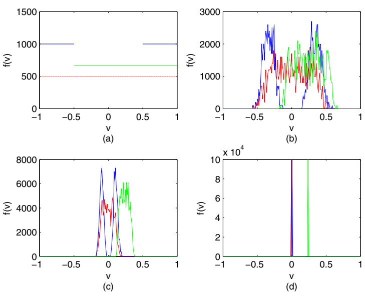


Fig. 2 The distribution functions $f(v,t)$ relative to $\gamma = 0.01$ and to: (a) $t = 0$; (b) $t = 4 \cdot 10^4$; (c) $t = 10^5$; (d) $t = 3 \cdot 10^5$.

We now show simulation results obtained considering a multi-agent system composed of 10^3 agents, whose opinions are initialized according to different distribution functions. From (4) it is expected that consensus is reached and, therefore, the stationary profile $f_\infty(v)$ is a Dirac's δ. Since, according to (23), the average opinion is constant, consensus is reached in correspondence of the average value of initial opinions, regardless of the value of $\gamma \in (0, 1/2)$ and of the initial opinion distributions $f(v, 0)$. We remark that the approach is general and the presented results are only illustrative. Three different initial distribution functions are considered and they are shown in Fig. 1 (a) and Fig. 2 (a), namely: agents with uniformly distributed opinions in the interval I (red line); half of the agents with uniformly distributed opinions in the interval (-1, -1/2) and half of the agents with uniformly distributed opinions in the interval (1/2, 1) (blue line); and agents with uniformly distributed opinions in the interval $(-1/2, 1)$ (green line). In the first two cases the value of the average opinion is $u = 0$ while in the last case $u = 1/4$. The second case differs from the first because the initial opinions of the agents are concentrated near the extremes of I. The choice of γ only influences the convergence speed, as shown in Figs. 1 and 2.

Fig. 1 shows the opinion distribution relative to $\gamma = 0.1$ and obtained after: (b) 10^4 interactions; (c) $2 \cdot 10^4$ interactions; and (d) $4 \cdot 10^4$ interactions. Regardless of $f(v, 0)$, $4 \cdot 10^4$ interactions are sufficient to reach consensus. Fig. 2 shows the results obtained with a smaller value of γ, namely $\gamma = 0.01$, which is equal to a tenth of the one previously considered. In particular, Fig. 2 shows the opinion distributions obtained after: (b) $4 \cdot 10^4$ interactions; (c) 10^5 interactions; and (d) $3 \cdot 10^5$ interactions. Comparing these results with those in Fig. 1 shows that decreasing the value of γ increases the number of interactions needed to reach consensus. As a matter of fact, while Fig. 1 (d) shows that $4 \cdot 10^4$ interactions are sufficient to achieve consensus if $\gamma = 0.1$, Fig. 2 (b) shows that they are not enough to reach consensus if $\gamma = 0.01$. Fig. 2 (d) shows that consensus is reached after $3 \cdot 10^5$ interactions if $\gamma = 0.01$.

4 Conclusions

This paper shows how consensus problems can be studied by applying ideas from kinetic theory of gases. Considering binary interactions and assuming that all agents can interact with each other, we show, through a kinetic approach, that consensus is always reached, despite of the parameters of the model. While the microscopic model used to describe the effects of interactions is very simple, the investigation of consensus problems through kinetic theory is not trivial. The considered model could be extended in order to take into account different characteristics, such as bounded confidence. Moreover, additional macroscopic properties of the system can be investigated by considering different test functions in the weak formulation of the Boltzmann equation. For instance, it can be shown that the standard deviation of the opinion tends to 0 exponentially, thus showing, analytically, that the proposed approach always leads to consensus. Possible evolutions of the model may involve non-deterministic parameters.

References

1. Bergenti, F., Franchi, E., Poggi, A.: Agent-based interpretations of classic network models. Computational and Mathematical Organization Theory **19**, 105–127 (2013)
2. Bergenti, F., Franchi, E., Poggi, A.: Agent-based social networks for enterprise collaboration. In: 20 th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 25–28 (2011)
3. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE **95**(1), 215–233 (2007)
4. Monica, S., Bergenti, F.: A stochastic model of self-stabilizing cellular automata for consensus formation. In: Proceedings of 15 th Workshop "Dagli Oggetti agli Agenti" (WOA 2014), Catania, Italy, September 2014
5. Schweitzer, F., Holyst, J.: Modelling collective opinion formation by means of active brownian particles. European Physical Journal B, 723–732 (2000)
6. Kempre, D., Dobra, A., Gehrke, J.: Gossip-based computation of aggregate information. In: Proceedings of the 44 th Annual IEEE Symposium Foundations of Computer Science (FOCS 2003), Hannover, Germany, pp. 71–78, March 2008
7. Fiedler, M.: Flocking for multi-agent dynamic systems: Algorithms and theory. Czechoslovak Math. **23**(98), 298–305 (1973)
8. Monica, S., Bergenti, F.: Simulations of opinion formation in multi-agent systems using kinetic theory. In: Proceedings of 16 th Workshop "Dagli Oggetti agli Agenti" (WOA 2015), Napoli, Italy, June 2015
9. Monica, S., Bergenti, F.: Kinetic description of opinion evolution in multi-agent systems: analytic model and simulations. In: Proceedings of the 18 th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2015), Bertinoro, Italy, pp. 483–491, October 2015
10. Groppi, M., Monica, S., Spiga, G.: A kinetic ellipsoidal BGK model for a binary gas mixture. EPL: Europhysics Letter **96**, December 2011
11. Pareschi, L., Toscani, G.: Interacting Multiagent Systems: Kinetic Equations and Montecarlo Methods. Oxford University Press, Oxford (2013)
12. Monica, S., Bergenti, F.: A kinetic study of opinion dynamics in multi-agent systems. In: Atti del Convegno (AI*IA 2015), Ferrara, Italy, September 2015
13. Toscani, G.: Kinetic models of opinion formation. Communications in Mathematical Sciences **4**, 481–496 (2006)

An Individual-Based Model for Malware Propagation in Wireless Sensor Networks

A. Martín del Rey, A. Hernández Encinas, J. D. Hernández Guillén, J. Martín Vaquero, A. Queiruga Dios and G. Rodríguez Sánchez

Abstract In this work a novel mathematical model to simulate malware spreading in wireless sensor networks is introduced. This is an improvement of the global model (based on a system of delayed ordinary differential equations) proposed by Zhu and Zhao in 2015 ([15]). Specifically, our model follows the individual-based paradigm which allows us to consider the particular characteristics and specifications of each element of the model.

Keywords Malware · Epidemic spreading · Wireless sensor networks · Individual-based model

1 Introduction

Wireless sensor networks (WSNs for short) are wireless networks of several of inexpensive miniature devices, called sensors, capable of computation, communication and sensing ([13]). This technology allows us the ability to observe the previously unobservable at a fine resolution over large spatio-temporal scales. WSNs have a wide range of applications to industry, science, civil infrastructure, transportation, physical security, environmental managing and tracking, etc. (see, for example, [4, 11]).

Contrary to what happens with traditional wireless networks, special security and performance issues have to be carefully considered for WSNs ([2]). Sensors can be easily compromised by an adversary since WSNs are usually deployed in hostile environments without human supervision ([14]). In this sense, the study and analysis of malware propagation in wireless sensor networks is an important challenge for security community. Unfortunately, there are not many models dealing with this problem.

A.M. del Rey(✉) · A. Hernández Encinas · J.D. Hernández Guillén · J. Martín Vaquero ·
A. Queiruga Dios · G. Rodríguez Sánchez
Department of Applied Mathematics, University of Salamanca, Salamanca, Spain
e-mail: {delrey,ascen,diaman,jesmarva,queirugadios,gerardo}@usal.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_24

The great majority of proposals study the dynamic of the malware propagation by means of systems of ordinary differential equations (see, for example, [3, 9]) or, more precisely, reaction-diffusion equations ([5, 16]). Especially interesting is the SIRS model proposed in [15]; it is based on a system of delayed ordinary differential equations where all sensors of the network are identical and they are be divided into three classes: susceptible (healthy), infectious and recovered (immunized) sensors. In this work, the local stability of the endemic equilibrium is obtained and, using sophisticated mathematical tools, an optimal control strategy is designed.

This is a global model based on continuos mathematical tools (differential equations) and, consequently, the particular characteristics and specifications of each sensor can not be considered. Moreover, it does not provide the individual evolution of each device. To avoid this drawback, it is necessary to consider individual-based models.

Individual-based models are mathematical approaches whose main goal is to enhance the understanding of different types of phenomena by modeling them as evolving systems of autonomous interacting entities ([7]). This paradigm is based on the local interaction of individual agents in simulated and discrete space and time to produce emergent, often nonintuitive outcomes at the individual level. It is achieved through the use of states (endowed to each agent) and local transition rules that govern the changes of these states through time. The main examples of this paradigm are cellular automata ([10]) and agent-based models ([8]).

Few works have been appeared dealing with the use of cellular automata to simulate malware spreading in WSNs. As far as we know, the most important is due to Hu and Song ([6]), where a SIR model based on the use of a two-dimensional cellular automata defined over an homogeneous cellular space is proposed. In this model each cell stands for a square area which can be occupied by (at most) one sensor. Two local transition functions are used to define the dynamic: one of them rules the detection, whereas the other one governs the spreading. They used both Von Neumann and Moore neighborhoods depending on the communication radius of the sensors.

The main purpose of our work is to design a novel malware propagation model in WSNs. More specifically, we intend to improve the global non-linear model recently proposed by Zhu and Zhao ([15]) by designing its individual-based counterpart. In a more precise way, we will model the dynamic of the system by means of a graph cellular automaton. The approach shown in this work is radically different from the one used in [6] since each individual agent (a cell of the cellular automaton) of our model stands for a particular sensor device instead of a regular area where a sensor can be placed (or not). As a consequence the WSN topology defines the cellular space as a graph whose nodes are the sensors, and a more realistic interaction between the devices is obtained.

The rest of the paper is organized as follows: In section 2 the description of the model due to Zhu and Zhao is shown; The novel individual-based model is detailed in section 3, and finally the conclusions and further work are presented in section 4.

2 Description of the Global Model Due to Zhu and Zhou

The model introduced in [15] is a compartmental model where the population of sensors is divided into three classes: susceptible, infectious and recovered sensors. In this sense, $S(t)$, $I(t)$, and $R(t)$ stand for the densities of susceptible, infectious and recovered sensors at time t, respectively. In this model the susceptible devices become infectious when the malware reaches them, the infectious sensors become recovered when the malware is detected and successfully removed, and finally some recovered devices change into susceptible again when they loose the immunity conferred by the antivirus software. Consequently, this is a SIRS model whose dynamic (see Figure 1) is given by the following system of delayed ordinary differential equations:

$$\begin{cases} \frac{dS(t)}{dt} &= rS(t)\left(1-\frac{S(t)}{K}\right)-\beta S(t)I(t)-\eta S(t)+\delta R(t-\tau) \\ \\ \frac{dI(t)}{dt} &= \beta S(t)I(t)-\varepsilon I(t)-\eta I(t) \\ \\ \frac{dR(t)}{dt} &= \varepsilon I(t)-\eta R(t)-\delta R(t-\tau) \end{cases} \tag{1}$$

with the following initial conditions: $S(t)=S_0, I(t)=I_0, R(t)=R_0, t\in[-\tau,0]$.

As the incidence follows the bilinear mass action, the number of new infectious nodes at time t is given by $\beta S(t) I(t)$ where $\beta > 0$ is the contact rate. A proportion of infectious sensors, $\varepsilon I(t)$, recovers from malware infection at time t due to the action of antivirus software. Some recovered sensors acquire permanent immunity and other become susceptible again with probability $\delta > 0$ after the temporary immunity period (of constant duration $\tau \geq 0$): $\delta R(t-\tau)$. Furthermore, the model also considers sensor vital dynamics. In this sense $\eta S(t)$, $\eta I(t)$ and $\eta R(t)$ are the proportion of susceptible, infectious and recovered sensors removed from the network at time t, where $\eta > 0$ is the death rate. On the other hand, the growth of the population of sensors is governed by means of the logistic equation with carrying capacity K and intrinsic increase rate r; consequently the new susceptible devices at time t is $rS(t)\left(1-\frac{S(t)}{K}\right)$.

3 The Individual-Based Proposed Model

3.1 Description of the Model

The model introduced in this work deals with routing protocols in WSNs based on the network structure; specifically, we will consider the hierarchical routing protocol SOP (*Self Organizing Protocol*) ([1]). This protocol supports heterogeneous sensor devices that can be mobile or stationary: sensor nodes, router nodes and sink nodes

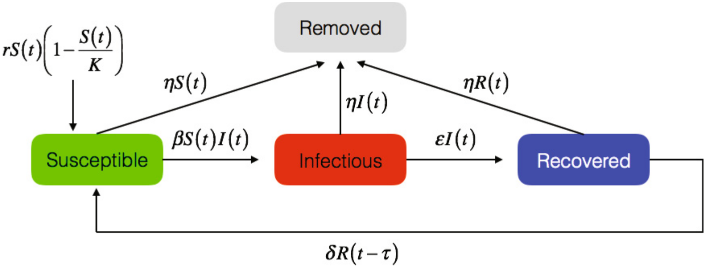


Fig. 1 Flow diagram of the transition of states.

(see Figure 2). The sensor nodes (or source nodes) monitor the environment and forward the data to router sensors; these nodes are usually grouped in clusters. The router sensors are stationary and forward the collected data from source nodes to sink nodes; only one router node will be associated to each cluster. Finally, the sink nodes have Ethernet functionality to connect to the Internet and send the data for further processing.

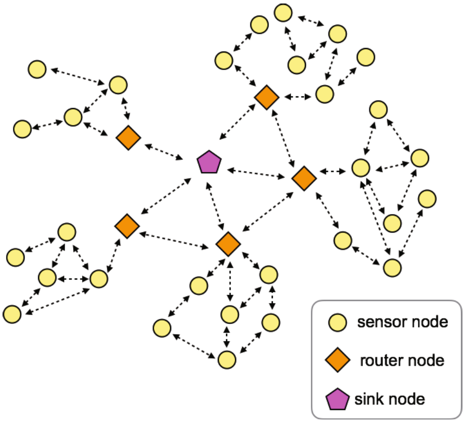


Fig. 2 An illustrative example of a WSN following SOP routing protocol.

The activity of the sensor nodes is controlled by means of the reduced-function device (RFD). This RFD regularly implements a sensing task, reports the sensor reading to a router node, and then goes to sleep for a certain period before waking up for the next round of sensing ([12]). As a consequence the life cycle of sensor nodes consists of three states that are repeated cyclically: sleep, wake-up and active. As is mentioned above, we will introduce the individual-based SIRS model counterpart of that one proposed in [15]. It is based on a graph cellular automata and its main characteristics are the following:

1. Each sensor device (sensor node, router node or sink node) of the wireless network stands for a cell of the cellular space. Let n be the number of nodes in the network; the i-th node will be labelled as $[i]$ where $1 \leq i \leq n$.
2. The topology of the network (that is, the graph defining the cellular automata) is giving by the routing protocol. This graph determines the neighborhood of every node $[i]$, $N[i]$, which are defined as the set of nodes that can be connected with $[i]$ at every step of time.
3. Each node is endowed with a state at every step of time. The possible states of the nodes are the following: sleep-susceptible, sleep-infectious, sleep-recovered, active-susceptible, active-infectious, active-recovered, and damaged.
4. The states of each node is updated in discrete steps of time accordingly to the following local transition rules:

   - Transition from susceptible to infectious: an active-susceptible node $[i]$ at time t becomes infectious at time $t+1$ with probability $\beta[i]$ if there exists a neighbor active-infectious node at time t, $[j] \in N[i]$. Note that a sleep-susceptible node at time t remains susceptible at time $t+1$.
   - Transition from infectious to recovered: an active-infectious node at time t becomes recovered at time $t+1$ with probability $\varepsilon[i,t]$. A sleep-infectious node at time t remains infectious at time $t+1$.
   - Transition from recovered to susceptible: immediately after the immunity period τ is finished, an active-recovered sensor becomes susceptible at the next step of time with probability $\delta[i]$. Otherwise, an active-recovered sensor remains in this state. As in the previous cases, sleep-recovered sensors at time t remains recovered at time $t+1$.
   - Transition from susceptible, infectious or recovered to damaged: an active sensor (regardless of its epidemic state) at time t changes into damaged at the next step of time with probability $\eta[i]$.

3.2 *Setting of Parameters*

It is extremely important to properly choose the parameters of the model. In what follows the main assumptions about the parameters involved in the proposed model are shown. The probability of infection is the same for all nodes of the same class, that is:

$$\beta[i] = \beta_{sensor}, \text{ if } [i] \text{ is a sensor node}, \tag{2}$$

$$\beta[i] = \beta_{router}, \text{ if } [i] \text{ is a router node}, \tag{3}$$

$$\beta[i] = \beta_{sink}, \text{ if } [i] \text{ is a sink node}, \tag{4}$$

such that $0 \leq \beta_{sink} \leq \beta_{router} \leq \beta_{sensor} \leq 1$. Note that this probability is greater in the source nodes than in the sink nodes because it is supposed that sink nodes are provided with more robust security (both physical and logical) measures.

The probability of recovery and the damage probability depend both on the node and on time, in such a way that:

$$0 \leq \varepsilon[\text{sensor node}, t] \leq \varepsilon[\text{router node}, t] \leq \varepsilon[\text{sink node}, t] \leq 1, \tag{5}$$

$$0 \leq \eta[\text{sink node}, t] \leq \eta[\text{router node}, t] \leq \eta[\text{sensor node}, t] \leq 1. \tag{6}$$

As in the previous case, the damage probability of sink nodes is lesser than the damage probabilities of router and source nodes. On the other hand, the recovery probabilities must be greater in sink nodes and, to a lesser extent, in router nodes since it is supposed that they are better maintained. Moreover, the level of maintenance may vary depending on the circumstances; in this sense, in this work we will suppose that for every i and every t (see Figure 3):

$$\varepsilon[i,t] = \epsilon[i]t\left(1+\epsilon[i]t\right)^{-1}, \quad \eta[i,t] = e[i]^{-1}\left(\text{Erf}\left(t-t_i\right)+1\right), \tag{7}$$

where $\epsilon[i] > 0, t_i > 0, e[i] > 0$. Note that in both cases, the probabilities tend to 1; in the first case (when the recovery transition is considered) the probability grows rapidly, whereas in the second case (damaged transition) the corresponding probability increases in a more attenuated form.

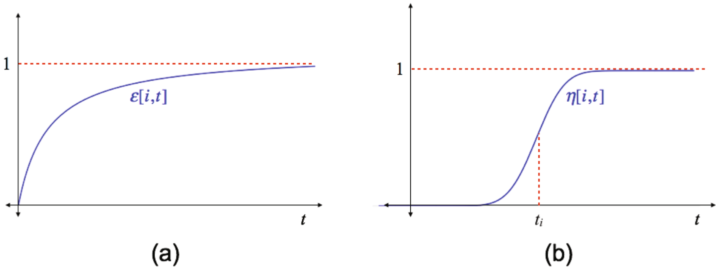


Fig. 3 (a) Probability of recovery of node $[i]$ at time t. (b) Probability of damage of node $[i]$ at time t.

3.3 Simulations

We will illustrate the model introduced in the last section by showing two simulations using the software Mathematica 10. The first one simulates the behavior of a global model (as is presented in [15]), and the second one considers different values of the coefficients associated to the nodes. 100 temporal units are computed in both cases whereas the number of devices are different: in the homogeneous case we will consider 100 sensors and in the other case only 35 sensors will be used.

It is supposed that in the “global" simulation all nodes are connected to all other nodes (the topology is defined by a complete graph), and the value of each parameter is the same for all nodes. Moreover, all nodes are active at every step of time. As a consequence, if we consider the parameters used in Example 1 of [15], that is: $\beta[i] = \varepsilon[i] = 0.3$, $\eta[i] = 0.05$, $\delta[i] = 0.4$, and $\tau = 4$ for every node $[i]$, the simulations obtained are shown in Figure 4-(a).

On the other hand, in Figure 4-(b) the simulation corresponding to an heterogeneous environment (whose topology is that one considered in Figure 2) is shown. It is supposed that the sleep period is chosen at random between 5 and 10 temporal units, and the active period has a duration of 3 time units (one to wake-up, one for monitoring and one to send the collected data). The probability of infection is given by:

$$\beta[i] = \begin{cases} 0.1, & \text{if } i = 1 \\ 0.15, & \text{if } 2 \leq i \leq 6 \\ 0.2, & \text{if } 7 \leq i \leq 35 \end{cases} \tag{8}$$

The probability of recovery is defined by $\epsilon[1] = 3$, $\epsilon[i] = 2$ with $2 \leq i \leq 6$, and $\epsilon[i] = 1$ for $7 \leq i \leq 35$, whereas the damage probability is given by $e[1] = 9.5$, $e[i] = 8.5$ for $2 \leq i \leq 6$, $e[i] = 7.5$ with $7 \leq i \leq 35$, and $75 \leq t_i \leq 90$, with $1 \leq i \leq 35$. Finally $\delta[i] = 0.15$ with immunity period $\tau_i = 10$ for every node.

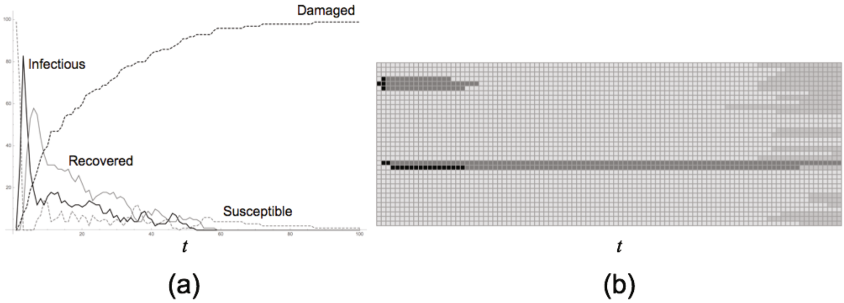


Fig. 4 (a) Evolution of the number of nodes in each compartment in the “global" simulation case. (b) Individual evolution of the nodes in the heterogeneous case. (Light gray: susceptibles, black: infectious, light black: recovered, and light gray: damages).

4 Conclusions and Further Work

In this work, an individual counterpart of a global model to study the propagation of malware in WSNs was proposed. This new approach overcomes the main drawback of global models: it is possible to consider the individual characteristics of the devices and, consequently, it is also possible to obtain the individual evolution of each sensor.

The experimental results show that the simulations obtained in the homogeneous case are in agreement with those ones obtained with the global model based on differential equations. Moreover, the global model tends to overestimate the force and spread of the infection (which is due to the topology based on a complete graph), whereas the simulations obtained with the proposed model are more accurate since the individual characteristics are considered. Finally, the control strategies must be based on the identification of the router and sink nodes and the isolation of the cluster infected by the malware.

Further work aimed at considering mobile sensor devices so that the topology of the graph changes with time. Moreover, different states could be considered as exposed or latent, infected (but not infectious), and, finally, a future and detailed study of the contact rate between the nodes and other coefficients is also interesting.

Acknowledgments This work has been supported by Ministerio de Economía y Competitividad (Spain) and the European Union through FEDER funds under grant TIN2014-55325-C2-2-R.

References

1. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Netw. **3**, 325–349 (2005)
2. Conti, M.: Secure Wireless Sensor Networks. Threats and Solutions. Advances in Information Security, vol. 65. Springer, NY (2015)
3. De, P., Liu, Y., Das, S.K.: An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks. IEEE. Trans. Mob. Comput. **8**(3), 413–425 (2009)
4. Fadel, E., Gungor, V.C., Nassef, L., Akkari, N., Malik, M.G.A., Almasri, S., Akyildiz, I.F.: A survey on wireless sensor networks for smart grid. Comput. Commun. **71**, 22–33 (2015)
5. He, Z., Wang, X.: A spatial-temporal model for the malware propagation in MWSNs based on the reaction-diffusion equations. In: Bao, Z., Gao, Y., Gu, Y., Guo, L., Li, Y., Lu, J., Ren, Z., Wang, Ch., Zhang, X. (eds.) Proceedings of WAIM 2012. LNCS, vol. 7419, pp. 45–56. Springer, Berlin (2012)
6. Hu, J., Song, Y.: The model of malware propagation in wireless sensor networks with regional detection mechanism. Commun. Comput. Inf. Sci. **501**, 651–662 (2015)
7. Jorgensen, S.E., Fath, B.D.: Individual-Based Model. Dev. Env. Model. **23**, 291–308 (2011)
8. Railsback, S.F., Grimm, V.: Agent-Based and Individual-Based Modeling. Princeton University Press, NJ (2012)
9. Shen, S., Li, H., Han, R., et al.: Differential game-based strategies for preventing malware propagation in wireless sensor networks. IEEE Trans. Inf. Forensic Secur. **9**(11), 1962–1973 (2014)
10. Wolfram, S.: A New Kind of Science. Wolfram Media, IL (2002)
11. Wu, M., Tan, L., Xiong, N.: Data prediction, compression and recovery in clustered wireless sensor networks for environmental monitoring applications. Inf. Sci. **239**, 800–818 (2016)
12. Yang, S.H.: Wireless sensor networks. Principles, design and applications. Springer-Verlag, London (2014)
13. Yick, J., Mukherjee, B., Ghosai, D.: Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2009)
14. Zhang, Y., Kitsos, P.: Security in RFID and Sensor Networks. Auerbach Publications, Boca Raton (2009)
15. Zhu, L., Zhao, H.: Dynamical analysis and optimal control for a malware propagation model in an information network. Neurocomputing **149**, 1370–1386 (2015)
16. Zhu, L., Zhao, H., Wang, X.: Stability and bifurcation analysis in a delayed reaction-diffusion malware propagation model. Comput. Math. Appl. **69**, 852–875 (2015)

Finding Electric Energy Consumption Patterns in Big Time Series Data

R. Perez-Chacon, R. L. Talavera-Llames, F. Martinez-Alvarez and A. Troncoso

Abstract In recent years the available volume of information has grown considerably due to the development of new technologies such as the sensor networks or smart meters, and therefore, new algorithms able to deal with big data are necessary. In this work the distributed version of the k-means algorithm in the Apache Spark framework is proposed in order to find patterns from a big time series. Results corresponding to the electricity consumptions for years 2011, 2012 and 2013 for two buildings from a public university are presented and discussed. Finally, the performance of the proposed methodology in relation to the computational time is compared with that of Weka as benchmarking.

Keywords Big data · Time series · Patterns · Clustering

1 Introduction

Rapid and huge data storage is in frequent use nowadays. This new scenario causes extreme difficulties to both efficiently process and store such big amount of data [6]. In this context, much effort is being devoted to enhance existing data mining techniques in order to process, manage and discover knowledge from this big data [13].

The limitations of the MapReduce paradigm [3] for iterative algorithms development have led to new paradigms, such as Apache Spark [8], which is an open source software project. Among its most important capacities, multi-pass computations, high-level operators, diverse languages usage, in addition to its own language called Scala, are most notable. Moreover, a machine learning library, MLlib [7], is also integrated within the framework.

R. Perez-Chacon · R.L. Talavera-Llames · F. Martinez-Alvarez · A. Troncoso(✉)
Division of Computer Science, Universidad Pablo de Olavide, 41013 Seville, Spain
e-mail: {rpercha,rltallla,fmaralv,ali}@upo.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_25

The objective of this work is the discovery of patterns in big time series of electricity consumption. Given its size, the collected data cannot be processed with classical machine learning approaches. Therefore, implementations for distributed computing must be used and, in particular, a distributed methodology based on the, still relatively unknown, parallelized version of k-means++ is proposed. This methodology has been developed by using MLlib in the framework Apache Spark, under the Scala programming language. Real–world big data sets collected from a sensor network located in several buildings of Pablo de Olavide University have been analyzed. The successful analysis of these patterns is expected to be used for efficient management of the university electricity resources, as well as for characterizing the electricity consumption over time.

Increased attention has been paid to big data clustering in recent years. A survey on this topic can be found in [5]. Specifically, several approaches have been recently proposed to apply clustering to big time series data. Namely, in [4] the authors propose a new clustering algorithm based on a previous clustering of a sample of the input data. The dynamic time warping was tested to measure the similarity between big time series in [14]. In [16] a data processing based on mapreduce was used to obtain clusters. A distributed method for the initialization of the k-means is proposed in [2]. However, there is still much research to be conducted in this field, especially considering that very few works have been published.

The study of electricity profiles by means of clustering techniques for small and medium datasets has been studied in the literature. In [15] a methodology based on the visualization to obtain the clusters is provided. In [12] the authors examined Spanish electricity prices, discovering some associated patterns to different days and to different seasons. The study was performed by applying crisp clustering, in contrast to the study carried out in [11], where fuzzy clustering was also shown to be useful in this context.

Clustering consumption data was also the goal in [10] but, this time, the authors went one step further and used this information as input for consumption forecasting.

Later in 2012, classification and clustering of electricity demand patterns in industrial parks was addressed [9]. In this work, a data processing system to analyze energy consumption patterns in Spanish parks, based on the cascade application of a Self-Organizing Maps and the k-means algorithm, was introduced.

As for the particular case of clustering big time series consumption data, there is no study carried out so far, to the best of the authors' knowledge. And this is precisely the reason why this study is presented.

The remainder of the paper is structured as follows. In Section 2 the proposed method to discover patterns in time series is described. Section 3 presents the experimental results corresponding to the clustering of the energy consumption coming from a sensor network of building facilities. Finally, Section 4 summarizes the main conclusions drawn from this study.

2 Methodology

This section describes the methodology proposed in order to find consumption patterns in electricity-related big time series data. In particular, the k-means algorithm, included in MLlib, is used in a Spark context to obtain clusters that define consumption patterns.

Figure 1 shows the key steps of the proposed methodology to obtain patterns as a result of the clustering. The first phase consists of data cleansing and the transformations carried out over a RDD variable of Spark, in order to use it in a distributed way. The dataset is the electricity consumption time series from two buildings from a public university for every fifteen minutes of the years 2011, 2012 and 2013. Each row of the dataset contains the following information: building name, date (split into five fields) and the electricity consumption data. Nevertheless, some of these rows contains accumulated consumption power data due to existing missing values, which were successfully preprocessed to learn correctly the models in the next phase. The preprocess stage is properly detailed in the Section 3.

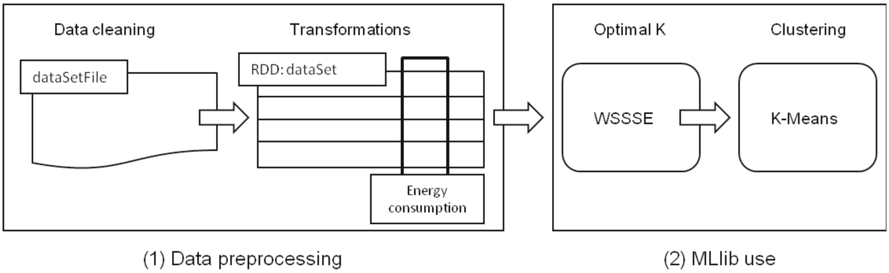


Fig. 1 Illustration of the proposed methodology.

Once the RDD dataset is created, it is necessary to group it in rows of 96 values (4 values per hour per 24 hours of a day) and reduce the dimension of the original RDD dataset before creating a model. This reduction consists in removing the building name field, and transforming the date into a numerical index. Thus, each row finally contains the 96 values corresponding to electricity consumption for a given day.

The second phase consists in the use of MLlib. Firstly, it is necessary to obtain an optimal number of clusters k, which will be used as an input parameter in the k-means algorithm. For that, the Within Set Sum of Squared Errors (WSSSE) index, defined by the sum of the squared Euclidean distance between the elements of a cluster and its centroid for all clusters and instances of the big data, is computed when applying the k-means for a certain number of clusters. In fact, the optimal k is usually the one which is a local minimum in the WSSSE graph.

MLlib includes a parallelized version of k-means++ [1], called k-means||, that it is used in this work to obtain the resulting models. The k-means|| algorithm runs the k-means algorithm a number of times in a concurrent way, returning the best clustering result.

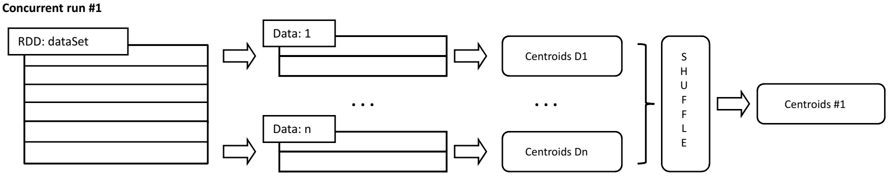


Fig. 2 One concurrent execution of the parallelized k-means algorithm.

In Figure 2 one execution of the the parallelized k-means algorithm version is described. Firstly, the RDD dataset is parallelized in n nodes for each concurrent run of the k-means. Therefore, n provisional centroids are obtained. Secondly, Spark shuffles the n centroids to provide a resulting centroid for each concurrent run. Finally, once all concurrent runs have been executed, the k-means|| algorithm computes the WSSSE for each centroid (note that there are as many centroids as number of concurrent runs). Thus, the algorithm returns the centroid that minimizes the WSSSE as the best centroid.

3 Results

The datasets used are related to the electrical energy consumption in two buildings located at a public university for years 2011, 2012 and 2013. The consumption is measured every fifteen minutes during this period. This makes a total of 35040 instances for years 2011 and 2013, and 35136 for the year 2012.

Note that there were several missing values ($< 3\%$). However, subsequent time stamps store the accumulated consumption for such instances. Therefore, the cleansing process consisted in searching for such values and assuming that consumption had been constant during these periods of time. That is, the stored value after missing values is divided by the number of consecutive registered missing values and assigned to each one.

Figure 3 shows the error obtained when applying the k-means for a number of clusters varying from 2 to 15 for the consumption of electricity of the years 2011, 2012 and 2013. The error used was the sum of squares of the distance between the points of each cluster. The error decreases smoothly for values greater than 6, do to this a number of clusters equal to 6 can be selected to provide satisfactory results.

Figure 4 presents the classification into 6 clusters obtained for K-means for the year 2013 (similar figures were obtained for the years 2011 and 2012). With just a quick look, the weekends, the working days and the typical periods of vacations in the university such as the Easter week (values from 83 to 90), the summer holidays (values from 213 to 243) or Christmas (values from 356 to 365) can be clearly differentiated.

The percentage of days classified into 6 clusters for each building is shown in Table 1. The last row presents the average of the electricity consumption for all the

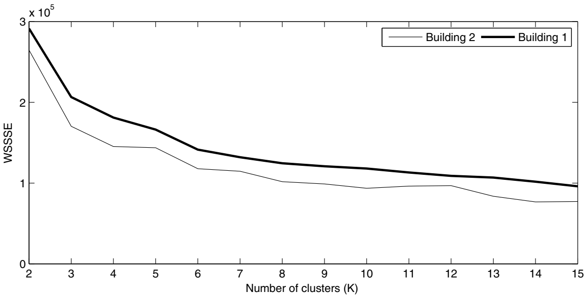


Fig. 3 Errors versus number of clusters for each building.

days associated with the cluster. Although it seems that the working days are equally distributed, a detailed analysis from Table 1 and Figure 4 reveals interesting patterns related to temperature and days with or with no scheduled teaching. For Building 1, teaching days with low and high temperatures (winter and summer) belong to the clusters of greater consumption, that is, cluster 4 and 5 respectively, and finally, teaching days with no extreme temperatures (autumn) are classified into cluster 2 of moderate consumption. Similar patterns can be found for Building 2.

Characteristic curves of each cluster are depicted in Figure 5. It can be observed that clusters 1 and 2 for Building 1, and clusters 1 and 2 for Building 2 are clusters composed of days of low consumption, namely weekends and holidays. Likewise, the remaining clusters correspond to working days, which are days of greater consumption.

Table 1 Percentage of days for each cluster in years 2011, 2012 and 2013.

Days	Building 1					
	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Monday	5.73%	15.29%	12.74%	17.83%	10.83%	37.58%
Tuesday	5.73%	19.75%	12.10%	19.11%	8.92%	34.39%
Wednesday	5.77%	20.51%	12.18%	16.67%	8.33%	36.54%
Thursday	6.41%	16.67%	15.38%	17.95%	8.33%	35.26%
Friday	10.90%	11.54%	14.74%	14.74%	10.26%	37.82%
Saturday	42.68%	2.55%	48.41%	0.00%	0.64%	5.73%
Sunday	16.56%	1.27%	81.53%	0.64%	0.00%	0.00%
Average (in kW)	1.90	3.37	4.57	5.47	6.54	6.94

Days	Building 2					
	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Monday	14.65%	38.22%	14.01%	16.56%	10.83%	5.73%
Tuesday	15.92%	32.48%	15.92%	19.11%	8.28%	8.28%
Wednesday	16.03%	35.90%	14.74%	15.38%	7.69%	10.26%
Thursday	19.87%	35.90%	15.38%	12.82%	7.69%	8.33%
Friday	19.23%	41.67%	13.46%	12.18%	8.97%	4.49%
Saturday	80.89%	17.83%	0.00%	0.64%	0.00%	0.64%
Sunday	96.82%	1.91%	0.00%	0.64%	0.00%	0.64%
Average (in kW)	1.41	2.39	3.96	4.71	5.43	6.39

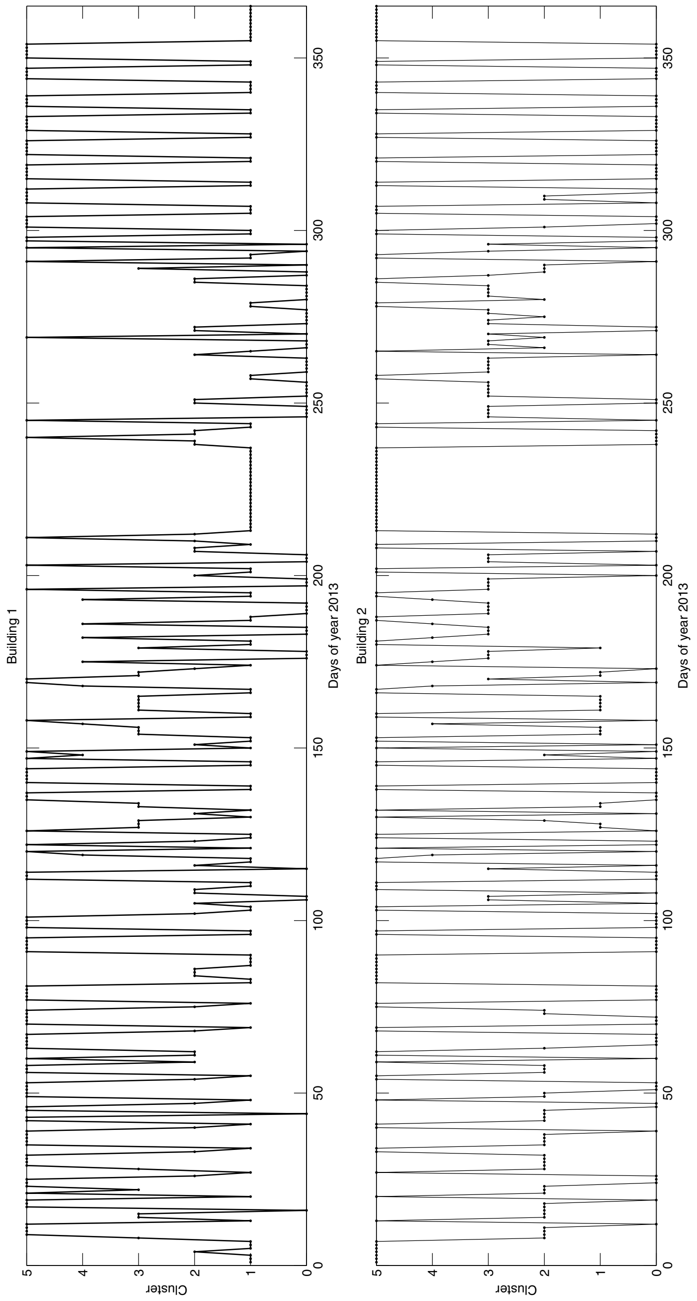


Fig. 4 Classification of the electricity consumption for each building.

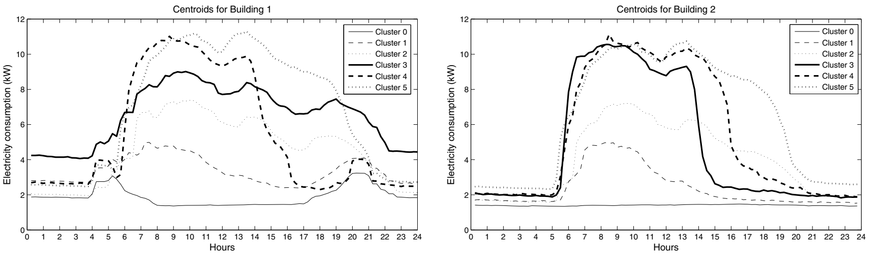


Fig. 5 Centroids for each building in years 2011, 2012 and 2013.

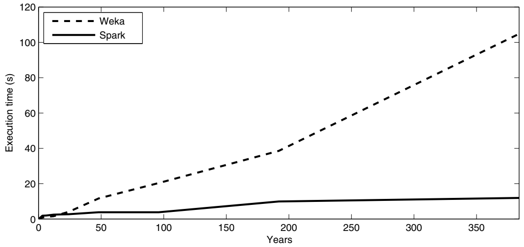


Fig. 6 CPU time for different sizes of datasets when using Weka and Spark.

Figure 6 shows the relation between the CPU time and the size of the dataset for k-means from Weka and Spark. This size has been set to 370 years, by synthetically generating such years. As a consequence of the results, it can be noticed that Weka has an exponential growth when the number of years comprising the time series increases, being remarkable the differences with Spark when the dataset is considered as big data.

4 Conclusions

In this work, a real big time series data composed of electricity consumptions has been analyzed by means of the k-means algorithm distributed version for Apache Spark. This parallelized version of the algorithm allows the discovery of daily consumption behaviors with a low computational cost. The results show different kinds of days according to the daily consumption as well as the identification of significant patterns related to working days with or with no scheduled teaching. Moreover, the CPU time of the proposed methodology has been compared to Weka, as a reference tool in data mining, proving to be a good solution for the big data clustering. Future works will be directed in the prediction of big time series once known the previous clustering, and the discovery of patterns for the classification of all the buildings of an organization in the context of smart cities.

Acknowledgments The authors would like to thank the Spanish Ministry of Economy and Competitiveness, Junta de Andalucía for the support under projects TIN2014-55894-C2-R and P12-TIC-1728, respectively.

References

1. Bahmani, B., Moseley, A., Vattani, R., Kumar, R., Vassilvitskii, S.: Scalable k-means++. Proceedings of the VLDB Endowment **5**(7), 622–633 (2012)
2. Capó, M., Pérez, A., Lozano, J.A.: A recursive k-means initialization algorithm for massive data. In: Proceedings of the Spanish Association for Artificial Intelligence (2015)
3. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)
4. Ding, R., Wang, Q., Dan, Y., Fu, Q., Zhang, H., Zhang, D.: Yading: fast clustering of large-scale time series data. Proceedings of the VLDB Endowment **8**(5), 473–484 (2015)
5. Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Zomaya, A.Y., Khalil, I., Sebti, F., Bouras, A.: A survey of clustering algorithms for big data: Taxonomy & empirical analysis. IEEE Transactions on Emerging Topics in Computing **5**, 267–279 (2014)
6. Fernández, A., del Río, S., López, V., Bawakid, A., del Jesús, M.J., Benítez, J.M., Herrera, F.: Big data with cloud computing: an insight on the computing environment, mapreduce, and programming frameworks. WIREs Data Mining and Knowledge Discovery **4**(5), 380–409 (2014)
7. Machine Learning Library (MLlib) for Spark (2015). http://spark.apache.org/docs/latest/mllib-guide.html
8. Hamstra, M., Karau, H., Zaharia, M., Knwinski, A., Wendell, P.: Learning Spark: Lightning-Fast Big Analytics. O' Really Media (2015)
9. Hernández, L., Baladrón, C., Aguiar, J.M., Carro, B., Sánchez-Esguevillas, A.: Classification and clustering of electricity demand patterns in industrial parks. Energies **5**, 5215–5228 (2012)
10. Keyno, H.R.S., Ghaderi, F., Azade, A., Razmi, J.: Forecasting electricity consumption by clustering data in order to decline the periodic variable's affects and simplification the pattern. Energy Conversion and Management **50**(3), 829–836 (2009)
11. Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C., Riquelme, J.M.: Discovering patterns in electricity price using clustering techniques. In: Proceedings of the International Conference on Renewable Energy and Power Quality, pp. 245–252 (2007)
12. Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C., Riquelme, J.M.: Partitioning-clustering techniques applied to the electricity price time series. In: LNCS, vol. 4881, pp. 990–991 (2007)
13. Minelli, M., Chambers, M., Dhiraj, A.: Big Data, Big Analytics: emerging business intelligence and analytics trends for today's businesses. John Wiley and Sons (2013)
14. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. ACM Transactions on Knowledge Discovery from Data **7**(3), 267–279 (2014)
15. Van Wijk, J.J., Van Selow, E.R.: Cluster and calendar based visualization of time series data. In: Proceedings of the International IEEE Symposium on Information Visualization (1999)
16. Zhao, W., Ma, H., He, Q.: Parallel k-means clustering based on mapreduce. In: LNCS, vol. 5391, pp. 674–679 (2009)

The Influence of Human Blaming or Bragging Behaviour Towards Software Agent Sincerity Implementation

Nur Huda Jaafar, Mohd Sharifuddin Ahmad and Azhana Ahmad

Abstract Machine ethics have become an important field of research in software agent technology. Granting autonomy to agents and instilling strong moral values in the agents have become a priority for designing agents to ensure that they will ethically perform tasks. Sincerity is one ethical behaviour that has been proven in human organizations to motivate humans in ethically performing their jobs. However, the sincerity is ruined if they display blaming or bragging behavior while performing their jobs. This paper shows how human blaming and bragging behaviour can influence software agent sincerity implementation. We propose operational rules of software agent sincerity implementation that responds to human blaming or bragging behaviour.

Keywords Software agent · Sincerity · Blaming · Bragging · Machine ethics

1 Introduction

Ethical behaviour among organizational members is very important in ensuring that the organization successfully delivers high quality services or products. Ethical behaviour can be defined as a “set of principles of right conduct” or “motivation based on ideas of right and wrong [1]. Previous research in social science

N.H. Jaafar(✉)
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA,
KM 12, Jalan Muar, 85009 Segamat, Johor Malaysia, Malaysia
e-mail: nurhu378@johor.uitm.edu.my

M.S. Ahmad · A. Ahmad
College of Information Technology, Universiti Tenaga Nasional, Putrajaya Campus, Jalan
IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia
e-mail: {sharif,azhana}@uniten.edu.my

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_26

show that unethical behaviours among organizational members give negative impacts on the organization [2]. One ethical behaviour that should be enculturated is sincerity. In fact, several researchers agree that sincerity is the foundation to sustain goodness of all human beings [3][5]. Sincerity behaviour has been proven as an important strong moral values that can help organizations to boost quality management and give many benefits to organizational members [4] [6] [8].

Nowadays, the growth of autonomous technology in various fields yields many benefits to humans. It reduces humans' burden by taking over some humans' tasks, especially those that entail high risks. However, researchers and developers of autonomous technology need to consider machine ethics in their products in order to ensure that the decisions made by these products, such as robots, are ethically right [9].

Software agents espouse the autonomous technology that plays the role of assistants to humans. Adapting human sincerity in software agents would bring benefits to software agent societies as exemplified by human societies. Instilling human sincerity behaviour in software agents would give rise to effective teamwork among agents in a society. Moreover, such sincerity behaviour could reduce conflicts between the agents, especially when a task requires the involvement of a teamwork. In this regard, a formula to instill and implement sincere behaviour in software agents needs to be precisely designed. In our previous research, we have identified the factors that influence humans sincerity while completing their tasks [10]. These factors are used as a guide for us to formulate sincerity behaviour in software agents.

A work phenomenon that affects sincerity is blaming others for one's failure or bragging about one's success. Such behaviours are negative elements of human sincerity. These negative behaviours can influence others to be insincere. In this paper, we show how human blaming and bragging behaviours can influence software agent sincerity implementation. We propose operational rules that show such influence on the software agent sincerity implementation.

This paper presents the work-in-progress of our research in implementing sincerity in software agents. The rest of this paper is organized as follows: Section 2 discusses the related works in the research of blaming, bragging and sincerity. It also reviews the research on machine ethics and software agents. Section 3 presents the operational rules of software agent sincerity implementation that influenced by human blaming and bragging behaviours. Finally, Section 4 concludes the paper.

2 Related Works

2.1 Sincerity, Blaming and Bragging

Sincerity is one human ethical behaviour that can bring about positive impacts on organizations or communities. According to Zhang [4], sincerity is the quality of being honest and genuine, and free from duplicity. A sincere person does things voluntarily with pure intentions without hoping for rewards [11]. Sincerity brings positive impacts to individuals and organizations such as being trusted by followers or clients [4], [7]. However, negative behaviour such as blaming and bragging not only promote negative influence on our sincerity, but also on the sincerity of others.

Blaming is the act of holding the cause of a negative outcome at fault or produced by an undesirable event towards someone associated with that event [12], [13]. It can be expressed in many ways such as explicit oral remark, body language or text [14]. Blaming is common in a team context [12]. Normally, it gives a negative impact towards an organization or a community. While some research consider the blaming behaviour in a positive view as a lesson for the organization to improve the quality of services or products [15], [16], in our context, this behaviour demonstrates a negative impact because it shows that the person is insincere in performing the task.

Bragging is another behaviour that shows that an individual or a group of persons is insincere. In demonstrating this behavior, a bragging person uses arrogant or boastful language to deliver the message [17]. The appearance of a bragging behaviour may cause difficulty for any person to accept the brag's opinions or suggestions. Moreover, a communication that consists of bragging or boasting about one's own success or possession may lead to other interpersonal negative consequences, such as feeling humiliated or envious [18]. These kinds of sentiments would have a bad impact on the organization.

2.2 Machine Ethics

Machine ethics is a branch of applied ethics created by researchers in order to produce a machine that can ethically play its roles. It is our responsibility to create machines that can follow the ideals of ethical principles while performing their tasks [10].

An ethical machine behaviour is able to set aside items that are not required as well as to highlight items to be utilised [19]. The trend of increasing autonomy in machines is a signal to researchers and developers of autonomous machines to include machine ethics as a part and parcel of their design priority. Moreover, the characteristics of autonomous machines that have the authority in decision-making at a certain level shows that the instilling of ethical behaviour in machines is very important [9]. Therefore, instilling ethical behaviour in machines is significant in order to command high level of trust from humans in letting the machines to work on behalf of humans in completing any tasks.

2.3 Software Agents

The autonomous technology in software agents depicts an autonomous machine. Its ability to react immediately to a dynamic environment with flexible autonomous actions is an evidence of its useful roles on humans' behalf [20]. The characteristics of a software agent, such as its ability to negotiate and cooperate with other agents render it useful for deployment in many social encounters. It can work in a team in order to achieve a common goal. While this behavior is commendable, an agent can be selfish when its individual goal becomes a priority [21]. Thus, it is not surprising if it only chooses to cooperate with a particular agent if this action increases its utility.

Instilling humans' strong moral values in software agents is a novel strategy to prevent agents from making unethical actions in their environments. Many software agent research have proven that adapting humans' behaviours in software agent is highly possible. Sincerity is one such strong moral value that can improve teamwork and prevent the agents from being exploited as unethical agents.

3 The Influence of Human Blaming or Bragging Behaviour on the Operational Rules of Software Agent Sincerity Implementation

As mentioned in the previous section, blaming and bragging may influence a negative outcome towards sincerity. Although the research in agent behaviour such as blaming and bragging had been discussed by other researchers, but it focused more on the reaction of this kind of behaviours among the agent communities. For example, Briggs and Scheutz [22]study on how to integrate the computational model of blame to prevent inappropriate threats to positive face and the focus is among agent communities. The finding is contrasting to our study, which we focus on how the human blaming and bragging can influence software agent sincerity.

Therefore, in this section, we propose operational rules for implementing software agent sincerity when humans show their blaming or bragging behaviour in task executions. Our focus is to demonstrate the reaction of an agent towards a human's blaming or bragging behaviour that can influence the agent's sincerity implementation. The agent initially detects a human's blaming or bragging behaviour and responds accordingly to other agents for further actions.

In order to detect such behaviour, the agent analyses message exchanges issued by humans. With this signal, if there are blaming or bragging messages, it decides whether to issue or abstain the issuance of the message to other agents or humans. The action taken by the agent reflects its sincerity level. This section briefly discusses the process of evaluating message exchanges issued by humans followed by the operational rules for implementing sincerity as influenced by the blaming or bragging behavior.

3.1 Evaluating Humans Message Exchanges

The messages from humans can be expressed as texts, speeches or body gestures [12], [23]. However, in this study, we focus on human messages in texts. The process starts when humans issue the messages and agents detect the messages from human. In our reseach, we consider that each human is paired with an agent as an assistant to the human. When the agent detects the message from its human counterpart, it evaluates the message. We apply the sentiment analysis approach to evaluate the message. Sentiment analysis is an approach for classifying sentence category [24], [25]. The basic classifications are positive or negative categories. Some research extend this category into other categories such as neutral and both (the sentence may fall into negative and positive opinion) [25].

Sentiment analysis can be implemented via machine learning or lexicon-based approach [24]. We opt for the lexicon-based approach at this stage. Dictionary-based is one of the lexicon-based approaches that can be used together with the manual approach for reducing time-consuming problem. Lexicon tools such as WordNet and Thesaurus are utilized in sentiment analysis. To obtain more accurate result of sentiment analysis, we combine the sentiment analysis with a natural language processing technique.

The result of the sentiment analysis is used for sentiment detection, which for this study, we focus only on blaming and bragging. If the result of the analysis indicates a blaming or bragging sentiment, then the agent exploits this information for further action. Figure 1 summarizes the process of evaluating human message exchanges.

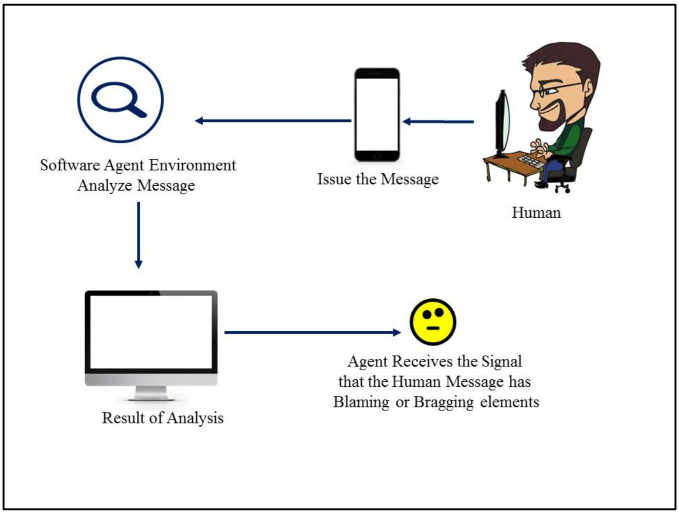

Software Agent Environment
Analyze Message
Issue the Message
Human
Result of Analysis
Agent Receives the Signal
that the Human Message has
Blaming or Bragging elements

Fig. 1 The Process of Evaluating Human Message Exchanges

3.2 *Agent Reaction towards Human Blaming or Bragging Behaviour*

This subsection formulates the operational rules for an agent's reaction when it gets the message's sentiment signal concerning the human's blaming or bragging behaviour. In this study, we consider two types of agent :

1. Worker agent; A_w: An agent which is in-charge of its own task or help its teammate to complete a task.
2. Head of Department (HOD) agent; A_{HOD}: This agent plays the role as a leader in a department or unit. It also monitors the work progress of all agents under its responsibility.

These agents analyze the blaming or bragging message issued by their human counterparts and decide to take actions. The decision for any actions determines whether the agent is sincere in completing its task. A merit point is awarded to the sincere agent, but a 0 point is otherwise given to the agent.

If the signal indicates that the human is bragging or blaming and the reaction of the agent is to extend the human's intention (i.e blaming someone or bragging about his/her achievement), then the agent gets zero (0) merit point because this action implies the agent's insincerity. Otherwise, if it decides not to extend the human's intention, it gets one (1) merit point as a reward for its sincerity. Figure 2 shows the operational rules of software agent sincerity implementation when reacting to human's blaming or bragging behaviour.

```
BEGIN
READ Signal
Signal = Blaming OR Bragging Element
IF $A_w$ OR $A_{HOD}$ issues the Blaming OR Bragging message then
  Merit point = 0
ELSE
  Merit point = 1
END
```

Fig. 2 The Operational Rules of Software Agent Sincerity Implementation reacting to a Human Blaming or Bragging Behaviour.

3.3 An Example Case Study

We present here a case which depicts a real world blaming scenario involving humans and how a sincerity-based agent architecture would respond in such scenario. The control of temperature, moisture and lighting in a library is very important to prevent books from destructive fungi. It should be solved immediately because books are important assets of the library.

A few books show signs of damage because there was a period (between January 2015 to December 2015) when a librarian-in-charge overlooked this problem and forgot to inform the Facility Management Department (FMD) to do inspection. Johan is a person-in-charge from FMD. He is requested by his Head of Department (HOD) to handle this problem. He did his work as required but his HOD receives negative feedback from the library management regarding the quality of Johan's work. His HOD instructs him to fix the problem.

Johan is unhappy with this situation. He issues a blaming message in the organization's messaging system. In this system, A_{WI} is an agent that assists Johan in completing his tasks. When Johan performs this action, the system detects and analyzes the message. The result is sent to A_{WI}, which decides to extend Johan's intention (of blaming the library management). When A_{WI} chooses this action, it shows that it is not sincere. So, it gets 0 merit point for its insincerity. Otherwise, if it blocks this message, it gets 1 merit point for demonstrating a sincere behavior.

In this scenario, we observe a benefit of deploying a sincerity-based agent system which could avoid a conflict from occurring between humans. Sincere agents can help to avoid conflicts by understanding sentiments within messages passed between humans.

4 Conclusion and Future Works

The increase in the number of autonomous machines as assistants to humans shows that instilling ethical behaviour in machines should be the top priority in autonomous systems research. Sincerity is one of the ethical behaviours that has been proven to help an organization maintain high quality services or products. By adapting sincerity behaviour in software agents, it offers a new mechanism for agents to efficiently perform tasks .

However, sincerity is ruined if there is a negative element, such as blaming or bragging, in task performance. In this paper, we conceive a mechanism for evaluating humans blaming or bragging messages and formulate operational rules for implementing agent sincerity when confronted with such messages. Sentiment analysis is used to analyse messages from humans to detect the blaming or bragging elements. The reaction of agent towards a blaming or bragging message determines the sincerity of the agent. A merit or a demerit point is given to show the dichotomy of agent sincerity.

In our future work, we shall test these operational rules by running simulations of the scenarios in order to validate the rules. We shall also integrate these rules with other operational rules of other sincerity factors to create a complete software agent sincerity implementation.

References

1. Hameed, A.S., Aljumaily, A.A., Ahmed, I.M., Daoud, I.J.: Ethical Based Mathematical Model to Evaluate Software Engineering. World Appl. Sci. J. (Mathematical Appl. Eng.) **21**, 44–49 (2013)
2. Bhatti, O.K., Alkahtani, A., Hassan, A., Sulaiman, M.: The Relationship between Islamic Piety (Taqwa) and Workplace Deviance with Organizational Justice as a Moderator. Int. J. Bus. Manag. **10**(4), 136–154 (2015)
3. Gandhi, A.: The sincerity of purpose: sustainability and world peace. In: Madhavan, G., Oakley, B., Green, D., Koon, D., Low, P. (eds.) Practicing Sustainability, pp. 45–49. Springer New York, New York (2013)
4. Di Zhang, J.P.D.: Quality alone is not enough to be trustworthy: The mediating role of sincerity perception. Int. J. Pharm. Healthc. Mark. **8**(2), 226–242 (2014)

5. Al-Mishri, M.: Ensiklopedia Akhlak Muhammad (SAW), 2nd end. Pena Pundi Aksara, Jakarta (2011)
6. Jaafar, N.H., Ahmad, M.S., Ahmad, A.: Operational rules for implementing sincere software agents in corrective and preventive actions environment. In: Comput. Intell. Inf. Syst., vol. 331, pp. 307–314 (2015)
7. Caza, A., Zhang, G., Wang, L., Bai, Y.: How do you really feel? Effect of leaders' perceived emotional sincerity on followers' trust. Leadersh. Q. **26**(4), 518–531 (2015)
8. Mat, Z., Basir, S.A., Zanariah, J.: A Study on Practice of Islamic Professional Ethics in Shaping an Ethical Work Culture within Malaysian Civil Service Sector. Asian Soc. Sci. **11**(17), 28–34 (2015)
9. Anderson, M., Anderson, S.L.: Robotics Robot Be Good. Scientific American, 72–77, October 2010
10. Jaafar, N.H., Basir, N.M., Ahmad, M.S., Ahmad, A.: Human sincerity factors for adaptation in software agents. In: 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014), November 2014
11. Jamaluddin, A.-S.M., Ad-Dimasqi, A.-Q.: Mutiara Ihya' Ulumuddin: Hak Milik Muslim - Imam Al-Ghazali, 5th edn. Illusion Network, Shah Alam (2013)
12. Groom, V., Chen, J., Johnson, T., Kara, F.A., Nass, C.: Critic, compatriot, or chump?: responses to robot blame attribution. In: Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction, pp. 211–218 (2010)
13. Gurdal, M.Y., Miller, J.B., Rustichini, A.: Why Blame? J. Polit. Econ. **121**(6), 1205–1247 (2013)
14. Friedman, M.: How to Blame People Responsibly. J. Value Inq. **47**(3), 271–284 (2013)
15. Vince, R., Saleem, T.: The Impact of Caution and Blame on Organizational Learning. Manag. Learn. **35**(2), 133–154 (2004)
16. Gorini, A., Miglioretti, M., Pravettoni, G.: A new perspective on blame culture: An experimental study. J. Eval. Clin. Pract. **18**(3), 671–675 (2012)
17. Decapua, A., Boxer, D.: Bragging, boasting and bravado: Male banter in a brokerage house : WL WL. Women Lang. **22**(1), 5–22 (1999)
18. Zammuner, V.L.: Felt Emotions, and Verbally Communicated Emotions: The Case of Pride. Eur. J. Soc. Psychol. **26**, 233–245 (1996)
19. Anderson, M., Anderson, S.L.: Toward Ensuring Ethical Behavior from Autonomous Systems: A Case-Supported Principle-Based Paradigm, pp. 19–28 (2014)
20. Wooldridge, M.: An Introduction to Multiagent Systems, 2nd edn. John Wiley & Sons Ltd., UK (2009)
21. Kaur, H., Kahlon, K.S., Virk, R.S.: Migration of Agents in Artificial Agent Societies: Framework and State-of-the-Art. Comput. Intell. Data Min. **3**, 27–51 (2015)
22. Briggs, G., Scheutz, M.: Modeling blame to avoid positive face threats in natural language generation. In: Proceedings of the INLG and SIGDIAL 2014 Joint Session, pp. 1–5, June 2014
23. De Melo, C.M., Carnevale, P., Gratch, J.: The effect of expression of anger and happiness in computer agents on negotiations with humans. In: Proceeding AAMAS 2011 The 10th International Conference on Autonomous Agents and Multiagent Systems, pp. 937–944 (2011)
24. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. Ain Shams Eng. J. **5**(4), 1093–1113 (2014)
25. Wilson, T.A., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity: an exploration of features for phrase-level sentiment analysis. Comput. Linguist. **35**(3), 399–433 (2009)

A Simulation Framework for Evaluating Distributed Reputation Management Systems

Vincenzo Agate, Alessandra De Paola, Giuseppe Lo Re and Marco Morana

Abstract In distributed environments, where interactions involve unknown entities, intelligent techniques for estimating agents' reputation are required. Reputation Management Systems (RMSs) aim to detect malicious behaviors that may affect the integrity of the virtual community. However, these systems are highly dependent of the application domain they address; hence the evaluation of different RMSs in terms of correctness and resistance to security attacks is frequently a tricky task. In this work we present a simulation framework to support researchers in the assessment of a RMS. The simulator is organized in two logic layers where network nodes are mapped to system processes that implement the interactions between the agents. Message Passing Interface (MPI) is used to enable communication among different distributed processes and provide the synchronization within the framework. A case study addressing the simulation of two different attacks to a RMS is also presented.

Keywords Simulation framework · Distributed reputation management · Multiagent system

1 Introduction

Nowadays many distributed applications are based on unknown agents that interact to each other in a cooperative way. However, malicious or selfish agents may act in a unpredictable way leading to a detriment of the performance of the distributed system. Reputation Management Systems (RMSs) are designed to detect and discourage antisocial behaviors that negatively affect the whole community. The role of a RMS is crucial in totally distributed systems, where a centralized entity capable

V. Agate · A. De Paola · G. Lo Re · M. Morana(✉)
Università Degli Studi di Palermo, 90128 Palermo, Italy
e-mail: {alessandra.depaola,giuseppe.lore,marco.morana}@unipa.it
http://www.dicgim.unipa.it/networks/ndslab/

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_27

Table 1 Comparison between simulation frameworks.

	Proposed Approach	**ART**	**TREET**	**[1]**
Independence from application scenario	X			X
Simulation of security attacks	X		X	X
Specification of event pattern	X			
Simulation of agent interactions	X	X	X	

of coordinating the interactions among agents is missing. In such a scenario, distributed RMSs allow each member of the community to contribute in estimating the reputation of the agents so as to reward those that act honestly and cooperatively.

Since reputation management systems are frequently designed to fit a specific application scenario, researchers are used to design ad-hoc simulators to evaluate the performances of a single RMS. In this work we move a step forward by presenting a novel simulation framework that aims to support researchers in the evaluation of the effectiveness and resistance to security attacks of a generic RMS. The simulator we propose here is independent of a specific application scenario, so it can be adopted in heterogenous contexts ranging from peer-to-peer applications for file sharing [16], e-Commerce frameworks [4] to service oriented architectures [2, 14].

In order to separate the high-level representation of the network from the underlying management of the simulation, the framework is organized in two logic layers, i.e., the *reputation layer* and the *simulation layer*. The overall RMS is modeled as a synchronous distributed algorithm, according to the principles described in [15]. Such assumption does not affect generality and correctness in the evaluation process and allows to straightforwardly verify the RMS's ability to correctly evaluate agents reputation and its resistance to security attacks.

The remainder of the paper is organized as follows: related work is reported in Section 2. The architecture of the simulation framework is described in Section 3, whilst Section 4 presents a case study on the use of the simulator to model the behavior of a RMS and two attacks to its security. Conclusions are discussed in Section 5.

2 Related Work

In distributed systems, where a central authority is missing, and the agents cooperatively estimate the reputation of the other members of the community, RMSs are susceptible to different type of security attacks [10]. Malicious agents may aim at different goals: *self-promoting*, to exploit system vulnerabilities in order to increment their own reputation; *slandering*, to decrease the reputation of a "victim"; *white-washing*, to "clean" a bad reputation avoiding the negative effects of the disincentive system; and *denial of service*, to block the functioning of the system. To be more

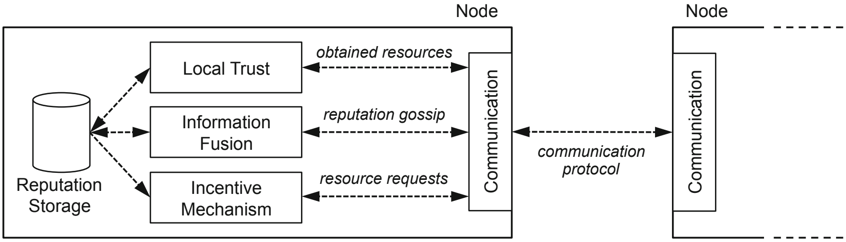


Fig. 1 The components of a distributed RMS. Each node privately performs local trust evaluation and information fusion algorithms; gossip protocol and incentive mechanism determine the interactions with other agents.

effective, attacks may follow an orchestrated plan that requires several malicious agents to work together.

Some testbeds and simulators have been proposed for assessing the performances of a RMS, but none of them allows for large-scale simulations of the behavior of a RMS under security attacks. ART (Agent Reputation and Trust) [5] is a popular simulation testbed in the field of multi-agent systems and allows to apply several evaluation metrics, and to define competitions in which different strategies can be combined and compared. TREET [12] allows to measure the resistance of a RMSs to some attacks (e.g., reputation lag, proliferation, and value imbalance), but does not consider common attacks to distributed systems, such as whitewashing and self-promoting. TREET overcomes many limitations of ART, allowing agents to randomly join or leave a running simulation. The testbed presented in [1] models existing RMSs as a set of transformations on a graph that represents transactions and trust among agents. Even if this system allows to evaluate security resistance to slandering and self-promotion attacks, it does not simulate agent interactions that are crucial to perform large-scale simulations where agents may modify their behavior. The main differences between the simulation framework proposed here and the discussed approaches are outlined in Table 1.

3 Simulation Framework Architecture

In order to design a generic framework, we identified the main components that are common to most of the distributed RMSs proposed in the literature (Fig. 1): (i) a *local trust* evaluation mechanism, used for assessing the behavior of the nodes involved in direct interactions, (ii) a *gossip protocol*, which propagates the local trust to other nodes of the network, (iii) an *information fusion* mechanism to merge information gathered through the gossip protocol with the local trust, and obtain the reputation values, and (iv) a *disincentive mechanism* which exploits reputation values in order to discourage antisocial behaviors.

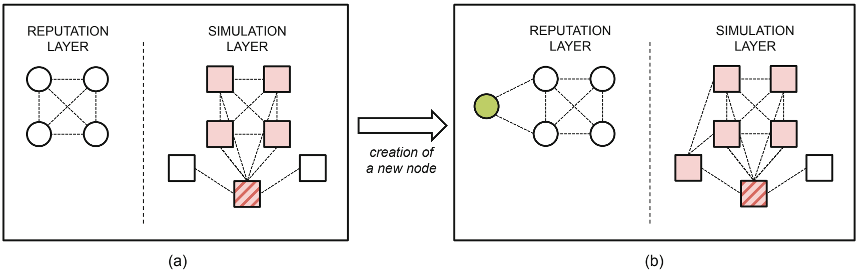


Fig. 2 Creation of a new node as seen at the reputation and simulation layer. (a) The agents (circles) are modeled by active processes (squares) managed by a controller (striped square). (b) When a new node (green circle) is connected to the network, one of the inactive processes (white squares) is woken up and communicates with other active processes.

In our framework we adopt a synchronous model to represent a RMS, which implies that the nodes of the reputation network act simultaneously. Such simplification allows to disregard many details (e.g., unpredictable communication delay or unfair work overload) that, although irrelevant to the assessment of the RMS, may affect the simulation. The simulation proceeds in rounds, each consisting of seven steps: (1) generate resource requests according to the current state, i.e. current view of other nodes reputation, (2) send resource request messages to the destination nodes, (3) elaborate incoming requests and determine whether provide a positive resource response according to the incentive mechanism and the current state, (4) send resource response messages to the destination neighbors, (5) update the current state according to the local trust evaluation, (6) send updated reputation values to the neighbors according to the gossip protocol, and (7) update the information fusion mechanism.

From a logical point of view, the simulation framework is organized in two different layers so as to to separate low-level functionalities, necessary for driving the simulation, from high-level routines, concerned with the tasks of the RMS. A similar architecture was adopted in a previous work where we addressed the design of a simulator for Wireless Sensor Networks in a distributed scenario [13].

The topmost layer, namely the *reputation layer*, is made of nodes connected to each other according to a specific network topology. At this level of abstraction, the reputation network is shown as totally distributed, non-centralized, and a set of high-level configuration utilities are offered to the user. At the *simulation layer*, each agent is mapped to a different process. When the simulator is started, a set of *inactive* processes (see Fig. 2-a) is created. Then, if a new node is added to the network, one of the *inactive* processes is awoken by the simulation manager, i.e. a process called *controller*, and linked to the other processes involved in the simulation (Fig. 2-b).

Since each process can run on a distinct remote host, we chose to adopt the Message Passing Interface (MPI) to enable communication among different distributed processes. MPI provides a protocol for parallel message-passing in distributed scenarios where processes exchange data through cooperative operations.

The synchronization is managed by means of blocking and non-blocking, point-to-point or collective, communication primitives that guarantee safe access to shared data. All processes use the thread-safe **MPI_Iprobe()** routine to manage incoming messages. Messages are sent and received by means of the nonblocking functions **MPI_Isend()** and **MPI_Irecv()** respectively, whilst **MPI_Barrier()** is used to synchronize all processes within the coordinator.

4 Case Study

In order to prove the effectiveness of our framework in evaluating a general reputation management system, we considered as case study a simple RMS inspired by [11] and [3].

The local trust component we adopted is a based on that used in EigenTrust [11], one of the best known RMSs for P2P networks. Each agent i stores the number of satisfactory, $sat(i, j)$, and unsatisfactory, $unsat(i, j)$, transactions between agents i and j. Then, the local trust s_{ij} is defined as the difference between such values, i.e., $s_{ij} = sat(i, j) - unsat(i, j)$. The simulation framework supports the user by providing the number of requests sent by each agent, together with the number of negative and positive feedback obtained by other agents. These values are updated at each time step; thus, the researchers can choose how much information they want to use, e.g., all values obtained since the beginning of the simulation or the average value computed in a sliding window.

The gossip protocol can be defined by means of a set of routines that allow for obtaining information about the reputation network and for supporting communication among agents. In particular, it is possible to get the list of the current neighbors, send unicast messages to specific neighbors, and send broadcast messages to the whole neighborhood. The gossip protocol deployed in the case study assumes that agents send their reputation values to all their neighbors. At the end of this phase, each agent knows the opinion of all its one-hop neighbors about the reputation of its two-hop neighborhood. Reputation information gathered so fare is then merged during the information fusion phase, inspired by the work proposed in [3]. Here each agent merges only information coming from reliable agents, i.e., those whose reputation is beyond a given threshold τ. Merged information is weighted with the reputation of the "gossiper" agents, and the resulting reputation value r_{ij} is a linear combination of this weighted mean and the normalized local trust c_{ij}, as specified by the following equation:

$$r_{ij}(t) = (1 - \beta) * c_{ij}(t) + \beta * \frac{\sum\limits_{k \in K} r_{ik}(t-1) * r_{kj}(t-1)}{\sum\limits_{k \in K} r_{ik}(t-1)}, \tag{1}$$

where β is a coefficient in [0, 1] and K is the set of reliable agents:

$$K = \{k : r_{ik}(t-1) \geq \tau\}. \tag{2}$$

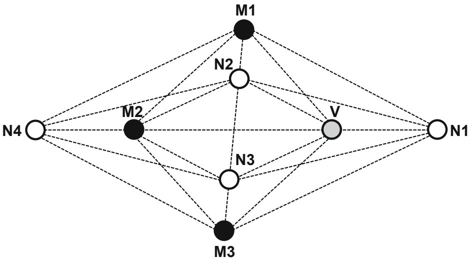


Fig. 3 The network topology used for simulating a slandering attack performed by the malicious nodes M_i against the victim node V.

Different attacks can be performed by defining the behavior of the agents through a set of configuration files, and by specifying the number of nodes, the cooperativeness degree of each agent, and the topology of the reputation network. It is important to guarantee that the network is not partitioned in islands, and that each node has an adequate number of neighbors. The current version of the simulator supports two of the most relevant attacks to distributed RMSs, i.e., slandering and whitewashing attacks.

Slandering attacks aim to change the reputation of other agents by disseminating false negative feedbacks in order to obtain some advantage. The simulation framework can be used to assess slandering attack by selecting a victim node V, and a set of M malicious nodes, adjacent to V. These nodes are programmed to share with their neighbors a fake reputation value of the node V, as shown in Fig. 3.

During a whitewashing attack, a malicious node exploits system resources until its reputation goes under an acceptable threshold; then, it leaves and rejoins the community with a new identity associated with a default reputation value. The scenario considered for evaluating whitewashing attacks is shown in Fig. 4. Here, we suppose that a node A, with a low reputation value, wants to duplicate itself to keep exploiting the system resources. In such a situation, the corresponding *active process* P_A sends a duplication request to the *controller* P_C. Then, P_C awakes one of the *inactive* processes P_E and change its state creating the process $P_{A'}$. This cloned process is initialized with the same adjacency list and behavior of P_A, e.g., if the node A was connected to B, then the process $P_{A'}$ will be connected to the process P_B.

At the end of a simulation, the framework provide the detailed trends of reputation and obtained resources over time for all agents. The comparison of the estimated reputation for a given agent with its cooperativeness degree allows to evaluate the average accuracy rate of a RMS, while the average utility provides information about the quality of service experienced by the end user. The slandering attack may be assessed by analyzing the reputation of the victim agent as observed from the point of view of a neutral agent. If the RMS is vulnerable to such class of attacks, the reputation of the victim node should decrease over time, in spite of its honest behavior, because of the effect of the false bad opinions gossiped by malicious agents. The whitewashing attack may be assessed by comparing the aggregate resources obtained

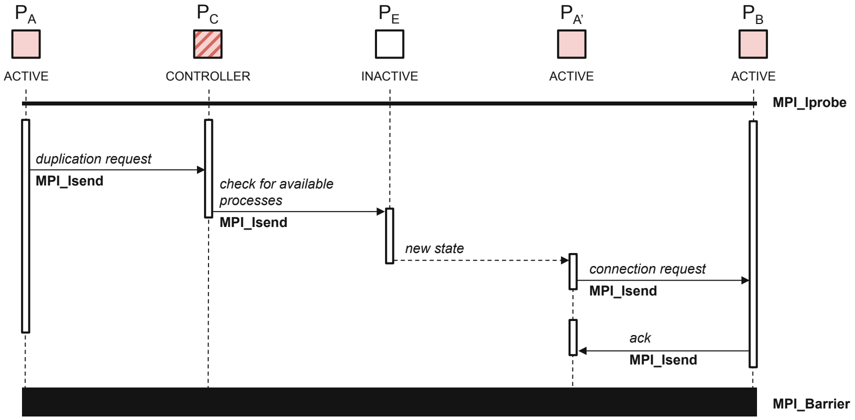


Fig. 4 High-layer interactions and MPI calls used to duplicate a node A, associated with a process P_A.

through all the false identities of the malicious agent, with respect to those obtained only by its original identity; such value should present an oscillatory trend over time, due to the connection of new identities.

5 Conclusions and Future Work

In this work we presented a two-layer simulation framework that can be used by researchers for an early evaluation of the performances of a Reputation Management System. The two logic layers allow to separates the reputation management techniques from the software routines that actually drive the simulation. Each agent of the *reputation layer* is mapped to a process running at the *simulation layer*, where MPI interfaces are used to enable communication among different distributed processes. The simulator offers several utilities that can be used to define both the topology of the simulation network, and all the parameters needed to implement the behavior of the simulated agents.

We presented a case study where an ad-hoc RMS was modeled and tested against two different attacks. The results we obtained demonstrate the suitability of the proposed framework for providing detailed information required to perform deep analyses on the performance of the considered RMS.

Future experiments will be performed in order to provide an in-depth evaluation of the framework scalability for large-scale simulations, in terms of memory occupancy, computational and communication complexity, also exploiting advanced techniques for an efficient allocation of processes [8, 9]. Moreover, we are considering the adoption of the simulator to model the interactions within large-scale social networks, e.g., Twitter [6, 7].

Acknowledgement Work partially supported by the Italian Ministry of Education, University and Research on the "StartUP" call funding the "BIGGER DATA" project, ref. PAC02L1_0086 CUP: B78F13000700008.

References

1. Chandrasekaran, P., Esfandiari, B.: Toward a testbed for evaluating computational trust models: experiments and analysis. Journal of Trust Management **2**(1), 1–27 (2015)
2. Crapanzano, C., Milazzo, F., De Paola, A., Lo Re, G.: Reputation management for distributed service-oriented architectures. In: Proc. of the 2010 Fourth IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshop (SASOW), pp. 160–165 (2010)
3. De Paola, A., Tamburo, A.: Reputation management in distributed systems. In: Proc. of the 3rd Int. Symp. on Communications, Control and Signal Processing (ISCCSP), pp. 666–670 (2008)
4. Fouliras, P.: A novel reputation-based model for e-commerce. Operational research **13**(1), 113–138 (2013)
5. Fullam, K.K., Klos, T.B., Muller, G., Sabater, J., Schlosser, A., Topol, Z., Barber, K.S., Rosenschein, J.S., Vercouter, L., Voss, M.: A specification of the agent reputation and trust (ART) testbed: experimentation and competition for trust in agent societies. In: Proc. of the Fourth Int. Joint Conf. on Autonomous Agents and Multiagent Systems, pp. 512–518 (2005)
6. Gaglio, S., Lo Re, G., Morana, M.: Real-time detection of Twitter social events from the user's perspective. In: Proc. of the 2015 IEEE Int. Conf. on Communications (ICC), pp. 1207–1212 (2015)
7. Gaglio, S., Lo Re, G., Morana, M.: A framework for real-time Twitter data analysis. Computer Communications **73**(Part B), 236–242 (2016)
8. Genco, A., Lo Re, G.: A recognize-and-accuse policy to speed up distributed processes. In: Proceedings of the Thirteenth Annual ACM Symp. on Principles of Distributed Computing, p. 386. ACM (1994)
9. Genco, A., Lo Re, G.: The egoistic approach to parallel process migration into heterogeneous workstation network. Journal of Systems Architecture **42**(4), 267–278 (1996)
10. Hoffman, K., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. ACM Computing Surveys (CSUR) **42**(1), 1 (2009)
11. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: Proc. of the 12th Int. Conf. on World Wide Web, pp. 640–651 (2003)
12. Kerr, R., Cohen, R.: Smart cheaters do prosper: defeating trust and reputation systems. In: Proc. of The 8th Int. Conf. on Autonomous Agents and Multiagent Systems, vol. 2, pp. 993–1000 (2009)
13. Lalomia, A., Lo Re, G., Ortolani, M.: A hybrid framework for soft real-time wsn simulation. In: Proc. of the 13th IEEE/ACM Int. Symp. on Distributed Simulation and Real Time Applications (DS-RT 2009), pp. 201–207 (2009)
14. Lee, J., Zhang, J., Huang, Z., Lin, K.J.: Context-based reputation management for service composition and reconfiguration. In: Proc. of the 2012 IEEE 14th Int. Conf. on Commerce and Enterprise Computing (CEC), pp. 57–61 (2012)
15. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann (1996)
16. Marti, S., Garcia-Molina, H.: Taxonomy of trust: Categorizing P2P reputation systems. Computer Networks **50**(4), 472–484 (2006)

On the Relationship Between Dimensionality Reduction and Spectral Clustering from a Kernel Viewpoint

D.H. Peluffo-Ordóñez, M.A. Becerra, A.E. Castro-Ospina, X. Blanco-Valencia, J.C. Alvarado-Pérez, R. Therón and A. Anaya-Isaza

Abstract This paper presents the development of a unified view of spectral clustering and unsupervised dimensionality reduction approaches within a generalized kernel framework. To do so, the authors propose a multipurpose latent variable model in terms of a high-dimensional representation of the input data matrix, which is incorporated into a least-squares support vector machine to yield a generalized optimization problem. After solving it via a primal-dual procedure, the final model results in a versatile projected matrix able to represent data in a low-dimensional space, as well as to provide information about clusters. Specifically, our formulation yields solutions for kernel spectral clustering and weighted-kernel principal component analysis.

Keywords Dimensionality reduction · Generalized kernel formulation · Kernel PCA · Spectral clustering · Support vector machine

D.H. Peluffo-Ordóñez
Universidad Técnica del Norte, Ibarra, Ecuador

D.H. Peluffo-Ordóñez
Universidad de Nariño, Pasto, Colombia

M.A. Becerra
Institución Universitaria Salazar y Herrera, Medellin, Colombia

M.A. Becerra · A.E. Castro-Ospina
Research Center of the Instituto Tecnológico Metropolitano, Medellin, Colombia

X. Blanco-Valencia(✉) · J.C. Alvarado-Pérez · R. Therón
Universidad de Salamanca, Salamanca, Spain
e-mail: xiopepa@usal.es

J.C. Alvarado-Pérez
Coorporación Universitaria Autónoma de Nariño, Pasto, Colombia

A. Anaya-Isaza
Universidad Surcolombiana, Neiva, Colombia

A. Anaya-Isaza
Universidad Tecnológica de Pereira, Pereira, Colombia

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_28

1 Introduction

In pattern recognition, the term kernel is used to define a function that establishes the similarity among given input elements. Therefore, a kernel function enables learning methods to use similarities for representing the samples or data points, instead of using explicitly the input data matrix [1]. Kernel-based methods have been widely exploited for both supervised and unsupervised learning approaches [2] showing their usability and versatility in several applications, such as image segmentation [3, 4], time-varying data analysis [5, 6], and hypothesis testing [7], among others. This article explores the benefit of using a kernel model within the design of spectral formulations of clustering and unsupervised dimensionality reduction methods.

On one side, kernel methods are of interest since they allow to incorporate prior knowledge into the clustering procedure [8]. In case of unsupervised clustering methods (that is to say, when clusters are naturally formed by following a given partition criterion), a set of initial parameters should be properly selected to avoid any local optimum solution distant from the desired global optimum. Indeed, in spectral clustering (SC), such initial parameters are traditionally the number of clusters and the input kernel matrix itself. On the other side, the aim of dimensionality reduction (DR) is to extract a lower dimensional, relevant information from high-dimensional data, being then a key stage for the design of pattern recognition systems. Indeed, when using adequate DR stages, the system performance can be enhanced as well as the data visualization can become more intelligible [9, 10]. Recent methods of DR are focused on the data topology preservation [11]. Mostly such a topology is driven by graph-based approaches where data are represented by a similarity matrix, and it is then susceptible to be expressed in terms of a kernel matrix [12], which means that a wide range of methods can be set within a kernel principal component analysis (KPCA) framework [13]. At the moment to choose a method for either SC or DR, aspects such as nature of data, complexity, aim to be reached and problem to be solved should be taken into consideration. In this regard, it must be quoted that there exists a variety of spectral methods making then the selection of a method a nontrivial task. In fact, some problems may require the combination of methods so that the properties of different methods are simultaneously exploited [14]. Some works have studied the benefit of taking advantage simultaneously of DR and SC techniques. For instance, in [15], a DR approach (linear feature extraction) is used to enhance the clustering performance by performing the grouping process over the projected data rather than over the original data. Other works are focused on generating variable relevance [16, 17] or data representation [18] criteria from conventional spectral clustering formulations.

In this work, the authors outline a unified formulation able to explain kernel approaches for both spectral clustering (SC) and unsupervised DR. Such a formulation starts with a latent variable model of a high-dimensional representation of the input data matrix, involving implicitly a mapping function. The model is incorporated within a quadratic functional, which along with an orthonormal constraint constitutes our optimization problem being a non-supervised version of a least-square-support-vector-machine (LS-SVM) formulation. Its solution is accomplished by relaxing the problem, and following a primal-dual scheme, which readily leads to a kernel representation given the quadratic nature of the functional. The proposed formulation represents a framework to easily understand the relationship between kernel-based approaches for SC and unsupervised DR. Also, the resultant model yields explicitly the solution of two well-known methods, namely the so-called kernel spectral clustering (KSC) proposed in [19], and weighted kernel PCA (WKPCA) [13].

The rest of this paper is organized as follows: Section 2 presents a brief overview on kernels. Section 3 describes our unified formulation, and explains the SC and DR perspectives. Finally, some final remarks are drawn in section 4.

2 Overview on Kernels

For following statements, let us consider the following notation: Let $\boldsymbol{Y} \in \mathbb{R}^{D \times N}$ be the input data matrix formed by N samples (data points), denoted by $\boldsymbol{y}_i \in \mathbb{R}^D$ with $i \in \{1, \ldots, N\}$. As well, from another point of view, it is conformed by D variables such that $\boldsymbol{y}^{(\ell)} \in \mathbb{R}^N$ is the ℓ-th variable, with $\ell \in \{1, \ldots, D\}$. Mathematically, kernels involve a mapping process from a d-dimensional input space representing a data set to a (d_h-) high-dimensional space, where $d_h >> D$. In terms of pattern recognition, the advantage of mapping the original data space onto a higher one lies in the fact that the latter space may provide a better data representation regarding cluster separability. Furthermore, it must be taken into account that the mapping is done before carrying out any clustering process. Then, the success of the data clustering task can be partly attributed to the kernel-matrix-building function when grouping algorithms are directly associated with the chosen kernel. Currently, kernels with special structure aimed to attend particular interests have been proposed. For instance, in [20], a structural clustering kernel is introduced by incorporating similarities induced by a structural clustering algorithm to improve graph kernels recommended by literature. Mercer kernels have been used for solving multi-cluster problems [21]. In [1], different kernels (generative, convolution, and covariance kernels, among others) are explained as well as important developments on how to construct kernels from a generating function are described.

2.1 Kernel Function

In terms of human learning theory, one of the fundamental problems is the discrimination among elements or objects. Consider the following instance: We have a set of objects formed by two different classes; then, when a new object appears the classificatio and/or visualization task is to determine to which class such an object belongs. This is usually done by taking into account the object's properties as well as similarities and differences with regards to the two previously known classes. According to the above, and regarding kernel theory, we need to create or choose a similarity or affinity measure to compare the data. Since such similarities are non-negative, kernel functions are positive-definite. A kernel function can be defined in the form: $\mathcal{K}(\cdot,\cdot) : \mathbb{K}^D \times \mathbb{K}^D \longrightarrow \mathbb{K}, \quad \boldsymbol{y}_i, \boldsymbol{y}_j \quad \longmapsto \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, where $\mathbb{K} = \mathbb{C}$ or $\mathbb{R}$. Note that in this case we have assumed elements $\boldsymbol{y}_i$ to be real and D-dimensional. Then, for a total of N data points, we can arrange the kernel function values into a $N \times N$ matrix $\boldsymbol{K}$ with entries $k_{ij} = \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, called Gram matrix or kernel matrix as well. Such a matrix is positive-semidefinite, i.e., a $N \times N$ complex matrix satisfying $\sum_{i=1}^{N}\sum_{j=1}^{N} c_i \bar{c}_j k_{ij} \geq 0$, for all $c_i \in \mathbb{C}$, being $\bar{c}_i$ the complex conjugate of c_i. Similarly, a real symmetric $N \times N$ matrix $\boldsymbol{K}$ satisfying the same condition given for all $c_i \in \mathbb{R}$ is also called positive-semidefinite. In terms of spectral matrix analysis, a symmetric matrix is positive-semidefinite if and only if all its eigenvalues are non-negative. In the literature, a number of different terms are used for positive-definite kernels, such as reproducing kernel, Mercer kernel, admissible kernel, support vector kernel, non-negative definite kernel and covariance.

2.2 Kernel Trick

Now, let us consider a function to map from the D-dimensional space to that d_h dimensional one is in the form $\phi(\cdot)$, such that: $\boldsymbol{\phi}(\cdot) : \mathbb{R}^D \longrightarrow \mathbb{R}^{d_h}, \boldsymbol{y}_i \longmapsto \phi(\boldsymbol{y}_i)$. The matrix $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}(\boldsymbol{y}_1)^\top, \ldots, \boldsymbol{\phi}(\boldsymbol{y}_N)^\top\right]$, $\boldsymbol{\Phi} \in \mathbb{R}^{d_h \times N}$, is a high dimensional representation of the input data matrix $\boldsymbol{Y}$. An interesting property of the kernel functions is the so-called *kernel trick*. In topology, a kernel function can be seen as an inner product in the domain of Hilbert space $\mathcal{H}$, as follows: $\mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \phi(\boldsymbol{y}_i), \phi(\boldsymbol{y}_i)\rangle_{\mathcal{H}}$. Kernel trick allows for performing the mapping and the inner product simultaneously by defining an associated kernel function. Then, we can estimate the kernel matrix without knowing the mapping function. This property gains importance in kernel theory, since it permits to replace a positive-definite kernel with another kernel that is finite and approximately positive-definite. For instance, from a given algorithm formulated in terms of a positive-definite kernel $\mathcal{K}$, we can construct an alternative algorithm by replacing it by another positive-definite kernel $\tilde{\mathcal{K}}$ [22], in such a manner that $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \boldsymbol{K}$. Then, in this case, kernel trick has served to estimate $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ as $\boldsymbol{K}$. In the domain of $\mathcal{H}$, $\boldsymbol{K}$ holds the inner product of the mapped data points (rows of matrix $\boldsymbol{\Phi}$), or -from another point of view- the outer product of the mapped variables (columns of matrix $\boldsymbol{\Phi}$).

3 Generalized Kernel Formulation

This section is aimed at formulating a model and cost function for a multipurpose data representation. To establish our model, let us consider an output data matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, being $d \leq D$ formed by N data points denoted by $\boldsymbol{x}_i \in \mathbb{R}^d$, with $i \in \{1, \ldots, N\}$, as well as by d variables denoted as $\boldsymbol{x}^{(\ell)} \in \mathbb{R}^N$ with $\ell \in \{1, \ldots, d\}$.

Also, let us assume an orthonormal projection matrix $\boldsymbol{W} \in \mathbb{R}^{D_h \times d}$, such that $\boldsymbol{W} = [\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(d)}]$ and $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_d$, where $\boldsymbol{w}^{(\ell)} \in \mathbb{R}^{D_h}$ and $\boldsymbol{I}_d$ is a d-dimensional identity matrix. Since $\boldsymbol{W}$ is orthonormal, elements $\boldsymbol{w}^{(\ell)}$ represent a d-dimensional base and can then generate a new space by means of a linear combination in the form: $\boldsymbol{x}^{(\ell)} = \boldsymbol{w}^{(\ell)}\boldsymbol{\Phi}$. So, the output matrix becomes $\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi}$. Here, in order to add an offset effect, we consider a whole latent variable model as $\boldsymbol{x}^{(\ell)} = \boldsymbol{w}^{(\ell)}\boldsymbol{\Phi} + b_\ell \boldsymbol{1}_N$. Such a model can be expressed in matrix terms as:

$$\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{1}$$

where b_l is a bias term, and $\boldsymbol{b} = [b_1, \ldots, b_d]$, $\otimes$ denotes Kronecker product, and $\boldsymbol{1}_N$ accounts for a N-dimensional all ones vector. Both PCA and SVM, in their simplest formulations, involve an energy term regarding the data matrix. Unlike conventional formulations that starts with a known input matrix, we pose a latent variable model, being unknown both variables (output and mapped data matrix) as well parameters (bias term and projection matrix). By incorporating a weighting matrix $\boldsymbol{\Delta} = \operatorname{Diag}(\delta_1, \ldots, \delta_N)$, the energy term regarding $\boldsymbol{X}$ can be written as $\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top$. Then, a functional in terms of the generalized matrix M-norm [17] can be expressed as:

$$\frac{1}{N}\operatorname{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top) = ||\boldsymbol{X}||^2_{(1/N)\boldsymbol{\Delta}}. \tag{2}$$

From another point of view, if we define a weighted output data matrix as $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{d \times N}$ as $\widetilde{\boldsymbol{X}} = \boldsymbol{X} \operatorname{Diag}(\delta_1^{1/2}, \ldots, \delta_N^{1/2})$, the functional $\operatorname{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top)$ can also be directly seen as an energy term, so: $\operatorname{tr}(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^\top)$. Our model can be determined by means of a primal-dual formulation as described below.

Primal Formulation: Recalling the functional given in equation (2) and the othonormality condition of projection matrix, we can write the following optimization problem:

$$\max_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{N}\operatorname{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top), \quad \text{s. t.} \quad \boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_d, \quad \boldsymbol{X} = \boldsymbol{\Phi}\boldsymbol{W} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{3}$$

which can be relaxed as

$$\max_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{2N}\operatorname{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top \boldsymbol{\Gamma}) - \frac{1}{2}\operatorname{tr}(\boldsymbol{W}^\top \boldsymbol{W}), \text{s. t.} \quad \boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{4}$$

where $\boldsymbol{\Gamma} = \operatorname{Diag}([\gamma_1, \ldots, \gamma_d])$ is a diagonal matrix holding regularization parameters.

Dual Formulation: To solve problem (4), we form the corresponding Lagrangian of problem stated in equation (4), as follows:

$$\mathcal{L} = \frac{1}{2N}\operatorname{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^{\top}\boldsymbol{\Gamma}) - \frac{1}{2}\operatorname{tr}(\boldsymbol{W}^{\top}\boldsymbol{W}) - \operatorname{tr}(\boldsymbol{A}^{\top}(\boldsymbol{X} - \boldsymbol{W}^{\top}\boldsymbol{\Phi} - \boldsymbol{b}\otimes\mathbf{1}_N^{\top})), \tag{5}$$

where matrix $\boldsymbol{A} \in \mathbb{R}^{N\times n_e}$ holds the Lagrange multiplier vectors, that is, $\boldsymbol{A} = [\boldsymbol{\alpha}^{(1)}, \cdots, \boldsymbol{\alpha}^{(n_e)}]$, being $\boldsymbol{\alpha}^{(l)} \in \mathbb{R}^N$ the l-th vector of Lagrange multipliers. Solving the Karush-Kuhn-Tucker (KKT) conditions on (5), we get:

$$\begin{aligned} &\frac{\partial\mathcal{L}}{\partial\boldsymbol{X}} = \mathbf{0} \Rightarrow \boldsymbol{X} = N\boldsymbol{\Delta}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}, && \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}} = \mathbf{0} \Rightarrow \boldsymbol{W} = \boldsymbol{\Phi}\boldsymbol{A},\\ &\frac{\partial\mathcal{L}}{\partial\boldsymbol{A}} = \mathbf{0} \Rightarrow \boldsymbol{X} = \boldsymbol{W}^{\top}\boldsymbol{\Phi} + \boldsymbol{b}\otimes\mathbf{1}_N^{\top}, && \frac{\partial\mathcal{L}}{\partial\boldsymbol{b}} = 0 \Rightarrow \boldsymbol{b}^{\top}\mathbf{1}_N = 0. \end{aligned}$$

Therefore, by applying Lagrange multipliers and eliminating the primal variables from the initial problem (3), the following eigenvector-based dual solution is obtained: $\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{A}\boldsymbol{\Delta}(\boldsymbol{I}_N + (\mathbf{1}_N \otimes \boldsymbol{b}^{\top})(\boldsymbol{K}\boldsymbol{\Lambda})^{-1})\boldsymbol{K}$, where $\boldsymbol{\Lambda} = \operatorname{Diag}(\boldsymbol{\lambda})$, $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$, $\boldsymbol{\lambda} \in \mathbb{R}^N$ is the vector of eigenvalues with $\lambda_l = N/\gamma_l$, $\lambda_l \in \mathbb{R}^{+}$. Again, $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is a given kernel matrix, satisfying the Mercer's theorem such that $\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} = \boldsymbol{K}$. In order to pose a quadratic dual formulation satisfying the condition $\boldsymbol{b}^{\top}\mathbf{1}_N = 0$ by centering vector $\boldsymbol{b}$ (i.e. with zero mean), the bias term is chosen in the form $b_l = -1/(\mathbf{1}_N^{\top}\boldsymbol{\Delta}\mathbf{1}_N)\mathbf{1}_N^{\top}\boldsymbol{\Delta}\boldsymbol{K}\boldsymbol{\alpha}^{(l)}$. Therefore, the solution of problem (4) is reduced to the following eigenvector-related problem:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{\Delta}\boldsymbol{H}\boldsymbol{K}\boldsymbol{A}, \tag{6}$$

where matrix $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ is the centering matrix that is defined as $\boldsymbol{H} = \boldsymbol{I}_N - (1/(\mathbf{1}_N^{\top}\boldsymbol{V}\mathbf{1}_N))\mathbf{1}_N\mathbf{1}_N^{\top}\boldsymbol{\Delta}$. Imposing a linear independency constraint on Lagrangian vector multipliers, $\boldsymbol{A}$ might be chosen as an orthonormal matrix. In consequence, a feasible solution is to estimate $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ as the spectral decomposition of a centered weighted kernel matrix $\boldsymbol{\Delta}\boldsymbol{H}\boldsymbol{K}$ -eigenvector and eigenvalue diagonal matrix, respectively. Finally, the output data matrix can be calculated as follows:

$$\boldsymbol{X} = \boldsymbol{A}^{\top}\boldsymbol{K} + \boldsymbol{b}\otimes\mathbf{1}_N^{\top}. \tag{7}$$

Given this, the solution is determined by the spectrum of a centered weighted kernel matrix and a bias vector defined so that the centering condition is ensured. In the following sections, we show how this solution can be applied for both dimensionality reduction and spectral clustering.

3.1 Dimensionality Reduction Perspective

Latent data matrix $\boldsymbol{X}$ is given by the linear model $\boldsymbol{W}^{\top}\boldsymbol{\Phi} + \boldsymbol{b}\otimes\mathbf{1}_N^{\top}$, which clearly involves a linear combination. If we seek for a low-dimensional representation of input data $\boldsymbol{Y}$,just estimation $\boldsymbol{X}$ with a low-rank version of $\boldsymbol{W}$. Such a estimation of the reduced matrix can be performed on the dual problem solution by using some eigenvectors from $\boldsymbol{A}$.

Weighted Kernel PCA: Given that the optimization is done under a maximization criterion, the eigenvectors associated with the largest eigenvalues should be selected. In this sense, final dimension d indicates how many eigenvectors are to be considered. Indeed, the eigenvalues of the centered weighted kernel defines the explained variance, so that the final dimension can be estimated with respect to it. Then, our generalized kernel model represents a weighted kernel PCA formulation when using a low-rank representation of matrix $\boldsymbol{W}$, being then able to embed a D-dimensional data matrix $\boldsymbol{Y}$ into a low-dimensional resulting matrix $\boldsymbol{X}$.

Kernel PCA: To yield conventional kernel PCA, the model should be considered as linear projection in the form $\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi}$. Since d is clearly less than d_h, a low-rank version of $\boldsymbol{\Phi}$ is then $\widehat{\boldsymbol{\Phi}} = \boldsymbol{W}\boldsymbol{X}$. So, we can write a functional to be minimized as $\frac{1}{N}||\boldsymbol{\Phi} - \widehat{\boldsymbol{\Phi}}||_F^2$, which has a dual problem given by:

$$\max_{\boldsymbol{X}} \ \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{K}\boldsymbol{X}), \quad \text{s. t.} \quad \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_d, \tag{8}$$

as widely explained in [13]. Therefore, a feasible solution is when $\boldsymbol{X}$ are the eigenvectors associated with the d largest eigenvalues. As well, this formulation can be seen as a generalized Weighted PCA when using a Mahalanobis distance regarding any positive-semidefinite matrix [13, 23]. Since kernel PCA is derived under the assumption that matrix $\boldsymbol{\Phi}$ has zero mean, centering becomes necessary. To satisfy this condition, we can normalize the kernel matrix with:

$$\begin{aligned} \boldsymbol{K} \leftarrow & \boldsymbol{K} - \frac{1}{N}\boldsymbol{K}\mathbf{1}_N\mathbf{1}_N^\top - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{K} + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{K}\mathbf{1}_N\mathbf{1}_N^\top \\ &= (\boldsymbol{I}_N - \mathbf{1}_N\mathbf{1}_N^\top)\boldsymbol{K}(\boldsymbol{I}_N - \mathbf{1}_N\mathbf{1}_N^\top). \end{aligned} \tag{9}$$

3.2 *Clustering Perspective*

Notice that the primal formulation given in (4) can be seen as a least-squares SVM. Then, our model should be able to provide information about the clusters immersed in data matrix. Since no supervised information is used, grouping process is fully unsupervise.

Kernel Spectral Clustering: Suppose that the output holds non-encoded information about centroids or prototypes for each cluster. Then, output data points should be represented in low dimension $d = K - 1$, being K the assumed number of clusters. Because each cluster is represented by a single point in the $K-1$-dimensional eigenspace, such that those single points are always in different orthants due also to the KKT conditions, we can encode the eigenvectors considering that two points are in the same cluster if they are in the same orthant in the corresponding eigenspace [19]. Then, a code book can be obtained from the rows of the matrix containing the $K - 1$ binarized leading eigenvectors in the columns, by using $\text{sign}(\boldsymbol{x}^{(\ell)})$. Then, matrix $\overline{\boldsymbol{X}} = \text{sgn}(\boldsymbol{X})$ is the code book being each row a codeword. Finally, clusters are formed according to the minimal Hamming distance between codewords within the space of $\overline{\boldsymbol{X}}$. This clustering approach is so-called kernel spectral clustering (KSC), introduced in [19].

Out-of-Samples Extension: The big advantage of this approach is that it can be extended to out-of-samples analysis without re-clustering the whole data to determine the assignment cluster membership for new testing data [19]. In particular, defining $\boldsymbol{z} \in \mathbb{R}^d$ as the projection vector of a testing data point $\boldsymbol{y}_{\text{test}}$, and by taking into consideration the training clustering model, the testing projections can be computed as $\boldsymbol{z} = \boldsymbol{A}^\top \boldsymbol{K}_{\text{test}} + \boldsymbol{b}$, where $\boldsymbol{K}_{\text{test}} \in \mathbb{R}^d$ is the kernel vector such that$\boldsymbol{K}_{\text{test}} = [K_{\text{test}_1}, \ldots, K_{\text{test}_N}]^\top$, where $K_{\text{test}_i} = \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_{\text{test}})$. Once, the test projection vector $\boldsymbol{z}$ is computed, a decoding stage is carried out that consists of comparing the binarized projections with respect to the codewords in the code book $\overline{\boldsymbol{X}}$ and assigning cluster membership based on the minimal Hamming distance [19].

4 Final Remarks

The aim of this paper is to state a generalized formulation able to explain the close relationship between spectral clustering and dimensionality reduction, within a kernel-based framework. Specifically, it has been shown that a least-square-support-vector-machine optimization problem, involving a latent variable model in terms of a high-dimensional representation of input data matrix, yields solutions containing information for encoding cluster assignment, and in turn for representing data matrix embedded in a lower-dimensional space. Furthermore, our formulation provides researchers on spectral, unsupervised pattern recognition methods with a fully matrix notation and formulation to easily understand kernel-based approaches such as KSC and KPCA.

Acknowledgements This work was supported by the Faculty of Engineering from Universidad Técnica del Norte.

References

1. Belanche Muñoz, L.A.: Developments in kernel design. In: European Symposium on Artificial Neural Networks (ESANN 2013). Computational Intelligence and Machine Learning, Bruges, Belgium, pp. 369–378 (2013)

2. Aldrich, C., Auret, L.: Statistical learning theory and kernel-based methods. In: Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods, pp. 117–181. Springer (2013)
3. Wu, Y., Ma, W., Gong, M., Li, H., Jiao, L.: Novel fuzzy active contour model with kernel metric for image segmentation. Applied Soft Computing (2015)
4. Molina-Giraldo, S., Álvarez-Meza, A., Peluffo-Ordóñez, D., Castellanos-Domínguez, G.: Image segmentation based on multi-kernel learning and feature relevance analysis. In: Advances in Artificial Intelligence–IBERAMIA 2012, pp. 501–510 (2012)
5. Langone, R., Alzate, C., Suykens, J.A.: Kernel spectral clustering with memory effect. Physica A: Statistical Mechanics and its Applications (2013)
6. Peluffo-Ordónez, D., Garcia-Vega, S., Langone, R., Suykens, J., Castellanos-Dominguez, G., et al.: Kernel spectral clustering for dynamic data using multiple kernel learning. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2013)
7. Harchaoui, Z., Bach, F., Cappé, O., Moulines, E.: Kernel-based methods for hypothesis testing: a unified view. IEEE Signal Processing Magazine **30**(4), 87–97 (2013)
8. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern recognition **41**(1), 176–190 (2008)
9. Alvarado-Pérez, J.C., Peluffo-Ordóńez, D.H.: Artificial and natural intelligence integration. In: 12th International Conference on Distributed Computing and Artificial Intelligence (DCAI 2016), pp. 167–173. Springer (2015)
10. Alvarado-Pérez, J.C., Peluffo-Ordóñez, D.H., Therón, R.: Bridging the gap between human knowledge and machine learning. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal **4**(1), 54–64 (2015)
11. Peluffo-Ordóñez, D.H., Lee, J.A., Verleysen, M.: Recent methods for dimensionality reduction: a brief comparative analysis. In: European Symposium on Artificial Neural Networks (ESANN). Citeseer (2014)
12. Ham, J., Lee, D.D., Mika, S., Schölkopf, B.: A kernel view of the dimensionality reduction of manifolds. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 47. ACM (2004)
13. Peluffo-Ordonez, D.H., Aldo Lee, J., Verleysen, M.: Generalized kernel framework for unsupervised spectral methods of dimensionality reduction. In: 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 171–177. IEEE (2014)
14. Peluffo-Ordónez, D.H., Alvarado-Pérez, J.C., Lee, J.A., Verleysen, M.: Geometrical homotopy for data visualization. In: European Symposium on Artificial Neural Networks (ESANN 2015). Computational Intelligence and Machine Learning (2015)
15. Peluffo-Ordónez, D.H., Alzate, C., Suykens, J.A.K., Castellanos-Domínguez, G.: Optimal data projection for kernel spectral clustering. In: European Symposium on Artificial Neural Networks. Computational Intelligence and Machine Learning, pp. 553–558 (2014)
16. Wolf, L., Bileschi, S.: Combining variable selection with dimensionality reduction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 801–806, June 2005
17. Peluffo, D.H., Lee, J.A., Verleysen, M., Rodríguez-Sotelo, J.L., Castellanos-Domínguez, G.: Unsupervised relevance analysis for feature extraction and selection: a distance-based approach for feature relevance. In: International Conference on Pattern Recognition, Applications and Methods - ICPRAM (2014)
18. Wolf, L., Shashua, A.: Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. Journal of machine learning **6**, 1855–1887 (2005)
19. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel pca. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(2), 335–347 (2010)

20. Seeland, M., Karwath, A., Kramer, S.: A structural cluster kernel for learning on graphs. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 516–524. ACM (2012)
21. Domeniconi, C., Peng, J., Yan, B.: Composite kernels for semi-supervised clustering. Knowledge and information systems **28**(1), 99–116 (2011)
22. Schölkopf, B., Smola, A.J.: Learning with Kernels (2002)
23. Peluffo-Ordóñez, D.H., Lee, J.A., Verleysen, M., Rodrıguez, J.L., Castellanos-Domınguez, G.: Unsupervised relevance analysis for feature extraction and selection. ICPRAM **2014**, 310–315 (2014)

Fingerprint Orientation Field Estimation Using ROEVA (Ridge Orientation Estimation and Verification Algorithm) and ADF (Anisotropic Diffusion Filtering)

Marco Antonio Ameller Flores and Angélica González Arrieta

Abstract The goal of this paper is to offer a joined approach in fingerprint orientation field estimation, integrating some of the most known techniques like ridge orientation estimation and image filtering, both tested using images from local and public databases. We propose a reliable orientation estimation algorithm [6] and anisotropic image filtering in this paper. To show the applied theory experimental results, we use Matlab for our implementation of the above algorithms. The investigation results showed robustness improving the correct estimation of the fingerprint ridge orientation process.

Keywords Fingerprint · Fingerprint enhancement · Orientation estimation · Orientation enhancement · Anisotropic filtering

1 Introduction

Correct estimation of fingerprint ridge orientation is an important task in fingerprint image processing. A successful orientation estimation algorithm can drastically improve tasks performs such as fingerprint enhancement, classification, and singular point extraction. Gradient-based orientation estimation algorithms are widely adopted in academic literature [4], but they cannot guarantee correctness of ridge orientations.

Fingerprints are the most important biometric identifier and are widely applied in automated fingerprint identification systems, used in large-scale civil identification

M.A.A. Flores(✉) · A.G. Arrieta
Department of Computer Science, University of Salamanca,
Plaza de los Caídos, 37008 Salamanca, Spain
e-mail: {ameller,angelica}@usal.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_29

projects. Fingerprint image consists of alternating pattern of ridges (dark area) and valleys (white area) [7], where ridges denoted by the black curves and valleys are the spaces between two neighbor ridges, see Fig. 1. These directional patterns from various fingerprint features, including singular points (delta and core), represents regional directional makeup and randomly distributed local discontinuities called minutiae (such as ridge end, and bifurcation).

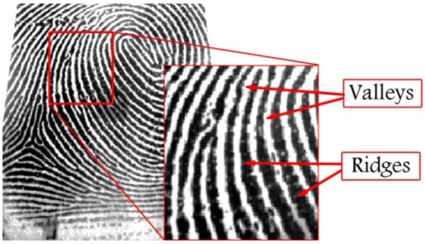


Fig. 1 Fingerprint image.

Developing a reliable fingerprint orientation estimation algorithm [6] is critical to creating a directional field from a fingerprint. A directional field denote the representation of ridge orientations, usually in square blocks, from the original fingerprint; generally used in fingerprint classification and singular points recognition.

The fingerprint orientation estimation algorithm performance is greatly influenced by the image quality, a correct orientation field is important to an Automated Fingerprint Identification System [9] see Fig. 2.

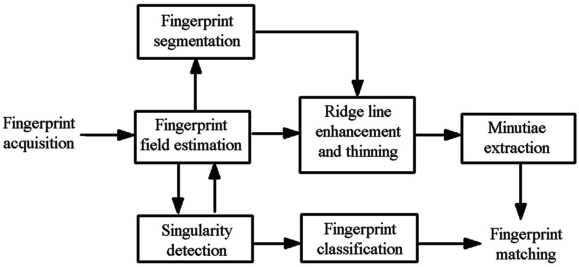


Fig. 2 Automated Fingerprint Identification System (AFIS).

2 Anisotropic Diffusion

The first phase of our approach consist in image enhancement by anisotropic diffusion, this algorithm anisotropically diffuses an image. That is, it blurs over regions of an image where the gradient magnitude is relatively small (homogenous regions) but diffuses little over areas of the image where the gradient magnitude is large.

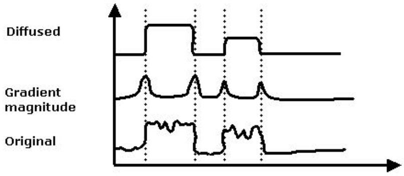


Fig. 3 Gradient anisotropic diffusion.

Note in Fig. 3 smooth homogeneous image regions and retains image edges. We attenuate the diffusion at the edge of signals using a gradient magnitude function [3].

Let I_t a gray-scale image in time t, Consider the anisotropic diffusion described in the following equation [8]:

$$I_t = div(c(x,y,t)\nabla I) = c(x,y,t)\Delta I + \nabla c * \nabla I \quad (1)$$

The equation (1), used to perform anisotropic diffusion, blurring on background noise, while sharpening edges, we indicate with div the divergence operator, and with ∇, and Δ the gradient and laplacian operators with respect to the scale variables, $c(x,y,t)$ is the spatial scale function, namely the diffusion coefficient which is a nonnegative monotonic decreasing function, I_t is the derivative of I with respect to time t, t is the time of thermal diffusion. The value of diffusion coefficient $c(x,y,t)$ will directly influence the filtering effect and in general, diffusion coefficient $c(x,y,t)$ is given a value of the norm of a vector $E(x,y,t)$, which has the following definition.

Let $E(x,y,t)$ be such an estimate: a vector valued function defined on the image which ideally should have the following attributes:

1. $E(x,y,t) = 0$ in the interior of each region.
2. $E(x,y,t) = Ke(x,y,t)$ at each edge point, where e is a unit vector normal to the edge at the point, and K is the local contrast (difference in the image intensities on the left and right) of the edge.

If an estimate $E(x,y,t)$ is available, the conduction coefficient $c(x,y,t)$ can be determined as:

$$c(x,y,t) = g(\|E\|) \quad (2)$$

Exists several possible choices for $g(\bullet)$, the simplest being a binary valued function, we use the edge estimate $E(x,y,t) = \nabla I(x,y,t)$ the choice of $g(\bullet)$ is restricted to a subclass of the monotonically decreasing functions.

$$g(\|\nabla I\|) = \frac{1}{1+(\|\nabla I\|/K)^2} \quad (3)$$

The constant K controls the sensitivity to edges and is usually chosen experimentally, in our case 0.25, in the Fig. 4 it is possible to see the anisotropic diffusion in an image sample block.

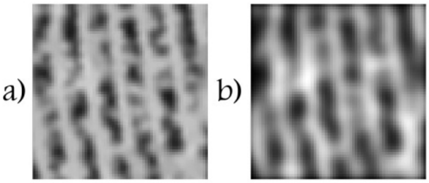


Fig. 4 Anisotropic diffusion, a) Original b) Diffused.

3 Ridge Orientation Estimation and Verification Algorithm (ROEVA)

The second phase of our orientation estimation algorithm ROEVA, is a rule based algorithm that can differentiate blocks with parallel ridge flows from those with non-parallel ridge flows.

Several algorithms are based on gradients that requires more computing power than ROEVA taken from the article [6], which presents an idea of low computational consumption can also provide orientation field and frequencies of a fingerprint, the paper presents the basics and not complete idea of the estimation process field orientations, so this article presents details and results of the algorithm.

A block with parallel ridge flows (105° orientation) illustrated in Fig. 5 (b). The line intensity values that is orthogonal to the ridge flows modelled as a sinusoidal wave which has the same frequency as the ridges and valleys. The wavelength of the sinusoidal wave considered as the “ridge width” *w*. If the ridge orientation miscalculated, the derived wavelength will then be longer than *w*. For example, if the ridge flows orientation is wrongly calculated as (45° orientation) see Fig. 5 (a), the line intensity values will form another sinusoidal wave with wavelength *l* that is longer than the ridge width w. Here we use “ridge length” to refer to the wavelength derived from such a sinusoidal wave. Note that the ridge length will be larger than (when the ridge orientation miscalculated) or equal to (correct ridge orientation) the ridge width *w*.

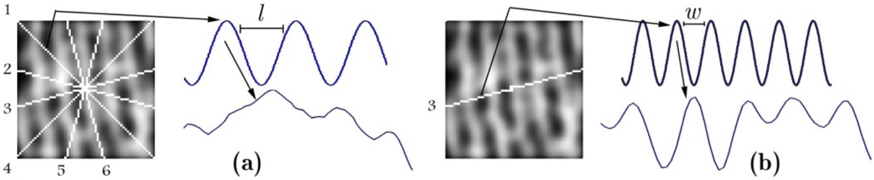


Fig. 5 Ridge orientation estimation (a) Six orientations to histogram examination; (b) Ridge flows with ideal corresponding histogram.

To blocks containing singular points we get irregular results, and nonsinusoidal patterns from blocks where no ridges.

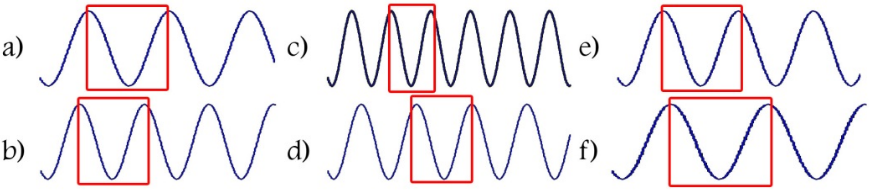


Fig. 6 Six orientations histograms.

A certain block contain at least a sinusoidal wavelength illustrated in Fig. 5 (b), otherwise it is an uncertain block none of six estimations it is present see Fig. 6, then ROEVA is defined step by step as follows [6]:

1. Divide each image into $p \times p$ pixel blocks where p is an odd number.
2. For each block, calculate the ridge lengths, L_i, $i = 0, \ldots, N$, where the orientation of L_i is set to $i \times (180/N)$. Let L_{min} be the minimum of L_i, and L_{max} the maximum.
3. Examine whether L_{min} and L_{max} are unique from 0° to 180°, ridge lengths from L_{min} to $L_{min} + 90$ monotonically increase, and ridge lengths from L_{max} to $L_{max} + 90$ monotonically decrease. If all conditions above are fulfilled, mark the block as certain block and set orientation of L_{max}. Otherwise, mark this block as an uncertain block.

In Fig. 7 shows a block of a fingerprint of 64 x 64 pixels and angle vectors labeled, to extract the gray levels at each angle.

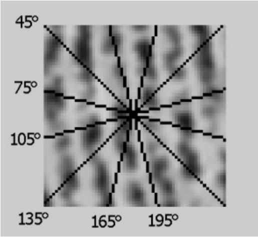


Fig. 7 Sampling vectors and angles.

We use the following steps to extract the orientation and frequency from sinusoidal wave of block:

1. For each vector use a low pass filter.
2. Estimate the high and low peaks.
3. Clean redundant peaks.
4. Estimate the variance between high and low peaks between peaks.
5. Calculate the wavelengths and number of waves.
6. Obtain the best approach from vectors using peaks variance and number of waves; refer to Fig 6 c).

In Fig. 8 it is possible to see the best sinusoidal wave approach (blue line) and peaks (red line), graph from Fig. 5 b).

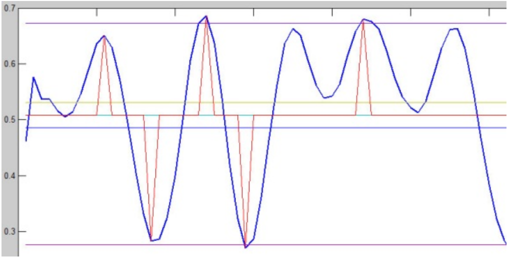


Fig. 8 Best vector approach, sinusoidal wave in blue, peaks in red.

To best sinusoidal wave in our case 105°, the orientation it is orthogonal to above angle (15° or 195°) see Fig. 9.

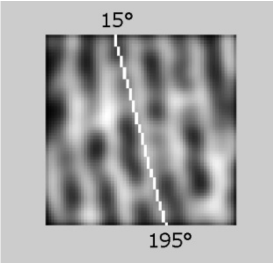


Fig. 9 Angle orthogonal to sinusoidal wave.

To represent a directional field, the range of the direction angles is $(0,\ \pi)$ or $(0^\circ,\ 180^\circ)$ then in our case a valid direction is 15°; the Fig. 10 shows results of four fingerprint blocks.

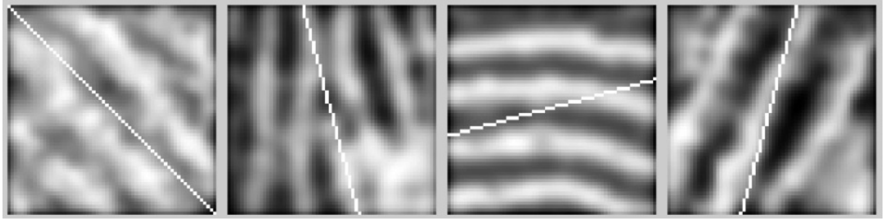


Fig. 10 Results of 4 blocks with different orientation and quality.

4 Experimental Results

In this section, we describe experimental results of fingerprint orientation field estimation using ROEVA and ADF using three different databases.

1) Local database of 150 fingerprint images taken with Digital Persona U. are. U 4500 Reader.
2) Public database of 80 fingerprint images DB2 from [1].
3) Public database of 128 fingerprint images from [11].

The aim for these experiments is to show that the orientations and frequency field estimation combining both algorithms significantly improve the accuracy results.

The results obtained in complete fingerprint images show an acceptable estimation of the orientations and frequency fields.

In Figures 11, 12 and 13 c) shows a better orientation field generated after applying ADF + ROEVA (Anisotropic Diffusion Filter + Ridge Orientation Estimation and Verification Algorithm).

On Table 1 for each image we have valid blocks (certain), true directions estimated and false wrong directions; Table 2 for ten images we have true directions estimated; qualified by simple observation and manual rate.

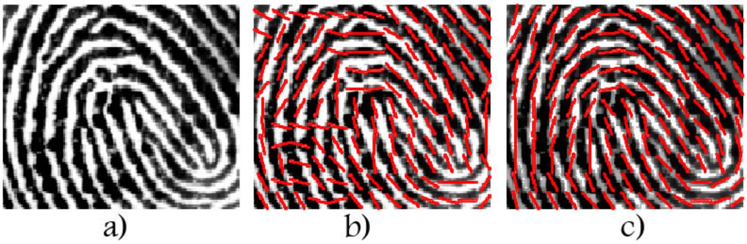


Fig. 11 Result in database 1); a) Original b) ROEVA c) ADF + ROEVA.

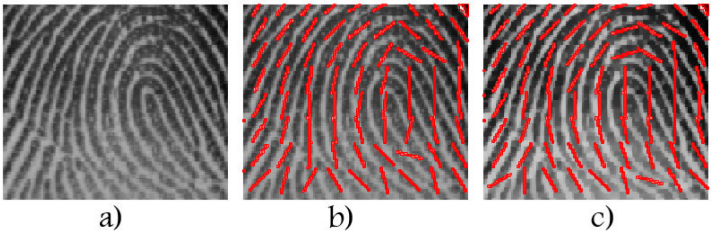


Fig. 12 Result in database 2); a) Original b) ROEVA c) ADF + ROEVA.

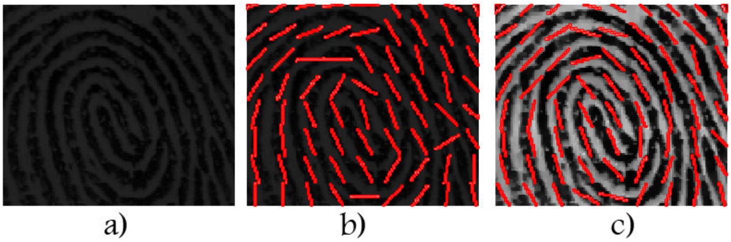


Fig. 13 Result in database 3); a) Original b) ROEVA c) ADF + ROEVA.

Table 1 Results on a single image from each database.

	Valid	ROEVA			ADF + ROEVA		
DB	Blocks	True	False	Percent	True	False	Percent
1)	780	450	330	57%	760	20	97%
2)	289	250	39	86%	256	33	88%
3)	187	101	86	54%	172	15	91%

Table 2 Results on 10 images from each database, aceptable direction blocks.

DB	Images	ROEVA	ADF+ROEVA
1)	10	50%	80%
2)	10	70%	72%
3)	10	61%	76%

5 Conclusions and Future Work

We have proposed a combined method to get orientation and frequency fields in fingerprint images, this combination provides consistent improvement, is therefore crucial to estimate the orientation correctly, since the computation of properties like singular points is depending on the orientation and frequency fields, this approach offers an alternative method to estimate orientation and frequency fields.

The proposed method is illustrated by examples in detail, and results using images from three different databases shows improvements with a low computing power than other methods like a "Least mean square orientation estimation algorithm" proposed in [5] and “Orientation estimation based on gradient” in [2].

Studying the recent literature on orientation modelling suggests that a solution is better with a suitable analytic expressions which result in a more specific and more general model than the existing ones[10].

As future work, is to achieve the development of a robust system to analysis, identification and recognition of fingerprint images with various methods including method proposed.

References

1. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: FVC2004 database DB2. Handbook of Fingerprint Recognition, 2nd edn. Springer, London (2009)
2. Zhu, E., Yin, J., Hu, C., Zhang, G.: A systematic method for fingerprint ridge orientation estimation and image segmentation. Pattern Recognition (2006)
3. Weickert, J.: Anisotropic Diffusion in Image Processing, B.G. Teubner Stuttgart (1998)
4. Lee, K.-C., Prabhakar, S.: Probabilistic Orientation Field Estimation for Fingerprint Enhancement and Verification. In: Biometrics Symposium (2008)

5. Hong, L., Wan, Y., Jain, A.K.: Fingerprint image enhancement: Algorithm and performance evaluation. IEEE Trans. Pattern Analysis. Machine Intelligence (1998)
6. Liu, L., Dai, T.-S.: A Reliable Fingerprint Orientation Estimation Algorithm (2011)
7. Nitika, Gill, N.S.: Fingerprint Recognition Techniques: A Critical Review. International Journal of Computer Science and Management Studies (2013)
8. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. University of California (1990)
9. Ratha, N.K., Chen, S.Y., Jain, A.K.: Adaptive flow orientation-based feature extraction in fingerprint images. Pattern Recognition (1995)
10. Ram, S.: Fingerprint Ridge Orientation Modeling, Graz University of Technology Institute for Computer Graphics and Vision (2008)
11. UPEK Fingerprint Database, April 12, 2012. http://www.advancedsourcecode.com/PNGfingerprint.rar

Instance Level Classification Confidence Estimation

Tuomo Alasalmi, Heli Koskimäki, Jaakko Suutala and Juha Röning

Abstract Often the confidence of a classification prediction can be as important as the prediction itself although current classification confidence measures are not necessarily consistent between different data sets. Thus in this paper, we present an algorithm to predict instance level classification confidence that is more consistent between data sets and is intuitive to interpret. The results with five test cases show high correlation between true and predicted classification rate, i.e. the probability of assigning the correct class label, thus proving the validity of the proposed algorithm.

Keywords Classification · Confidence · Model uncertainty

1 Introduction

Machine learning is an excellent tool to model some phenomenon without knowing the exact model behind it when we do have some data describing it. For example, in predictive modelling a variable of interest is predicted based on a model that was build based on training data. The prediction can be a continuous value, i.e. regression, or a discrete, categorical value, i.e. classification. The predictive model is, however, never perfect and it is important as the user of the model to understand just how reliable the predictions are. Nevertheless, current methods for instance level classification confidence estimation are not consistent between data sets. Also, missing values are common in real-world data sets and decrease confidence in the predictions. In this paper, we will present an algorithm to estimate instance level classification confidence that is much more consistent and also intuitive to interpret.

T. Alasalmi(✉) · H. Koskimäki · J. Suutala · J. Röning
University of Oulu - Biomimetics and Intelligent Systems Group,
P.O. Box 4500, 90014 Oulu, Finland
e-mail: {tuomoala,hejunno,jaska,jjr}@ee.oulu.fi

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_30

2 Background

In almost all modelling studies, model performance is assessed in some way. Common measures such as classification rate, sensitivity, specificity, and receiver operating characteristic (ROC) curve evaluate the model performance on a whole test data set and do not make a difference between individual instances of predicted data points. On the other hand classification confidence estimation of individual instances is less common.

It has been stated that it is valuable to know how confident one is about a solution provided by an intelligent system [1] and several confidence measures have been proposed for e.g. k-Nearest Neighbours (kNN) [1, 2, 3] and for Naive Bayes and Support Vector Machine (SVM) classifiers [2]. But using just the numeric output of kNN, Naive Bayes, Neural Networks, or SVM classifiers are not very well correlated with classification confidence [2].

The most common measure of classification confidence is a posteriori probability estimate of a classifier. The disadvantage is that it only takes into account the a posteriori probability of the predicted class. There is an improved uncertainty measure, presented in Equation 1, that takes into account the a posteriori probability of the predicted class and the number of classes [4]. Even this measure, however, is not linearly correlated with classification rate [5]. Instead, the measure behaves differently with different data sets which can be seen in Figure 1. Each data point in the figure represents the percentage of correctly classified samples of 200 test data set samples with similar uncertainty and the average uncertainty of those samples. It is apparent that the uncertainty measure is not straightforward to interpret because a similar uncertainty value can indicate drastically different confidence in different contexts. To demonstrate different contexts and different ranges of confidence values, different data sets and artificially introduced missing values are used in this work. Missing values are also common in many real-world data sets making this approach natural.

$$U = 1 - \frac{\max_{i=1,\ldots,n}(p_i) - \sum_{i=1}^{n}(p_i)/n}{1 - 1/n} \in [0, 1] \tag{1}$$

In Equation 1, U stands for the uncertainty of the classification prediction, p_i for the a posteriori probability of class i, and n for the number of classes. The measure is scaled to the range [0, 1] higher value meaning more uncertain or less confident prediction.

3 Methods

In this section we will present the developed algorithm for estimating classification confidence as the probability of the prediction belonging to correct class. We refer to this as predicted classification rate.

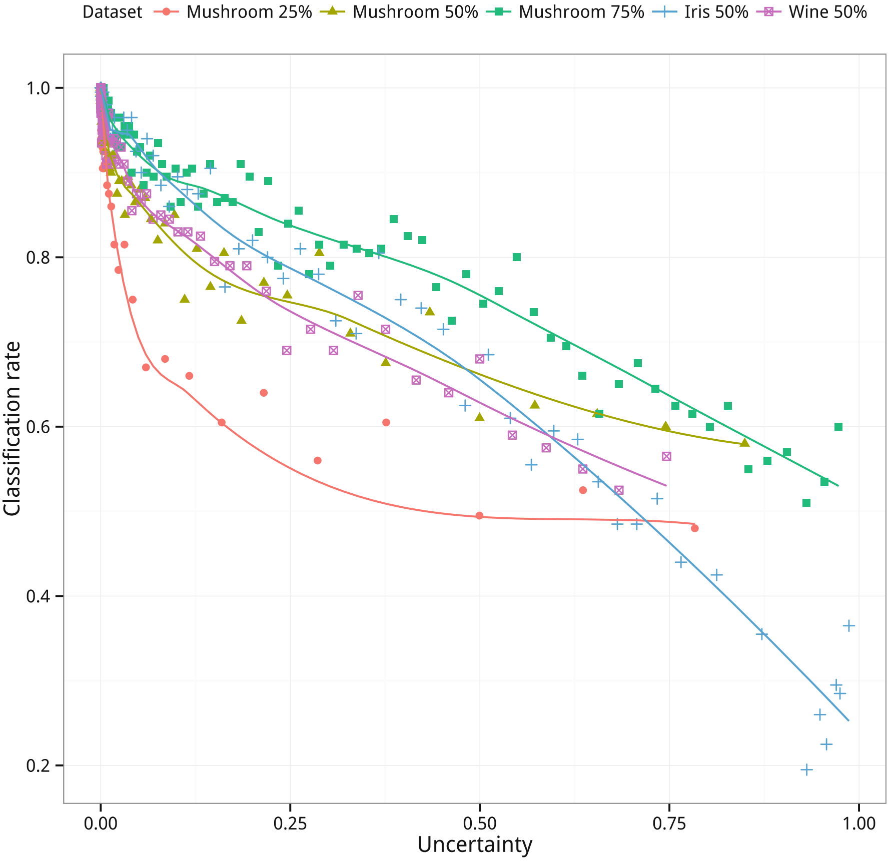


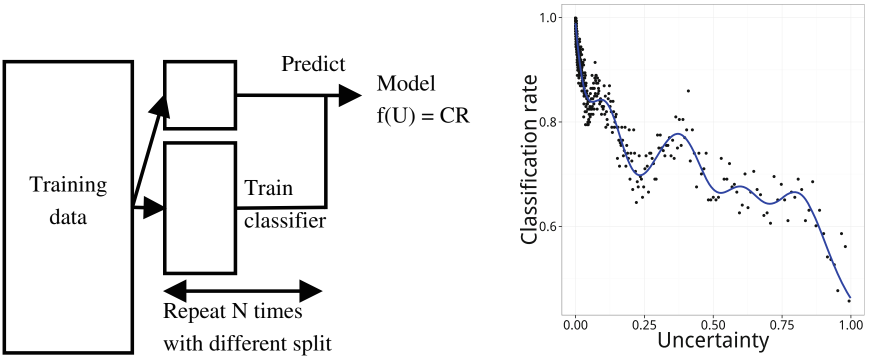
Fig. 1 Classification rate of Naive Bayes classifier as a function of estimated uncertainty in different datasets and with different number of missing data values.

Obviously, a single prediction is either right or wrong and classification rate cannot be determined for a single prediction. Therefore, to build a model for classification confidence, a large enough number of data points with a prediction are needed. With smaller data sets the data can be generated by repeating a cross validation scheme with a split to training (70%) and test data sets (30%) until a large enough number of data points are generated. Each round the training data set is used to train a classification model and using this model the test data points are predicted and the uncertainties estimated with Equation 1 for each data point.

Next step is to sort the predicted data points according to their uncertainty value. Using this order of the data points, a number of neighbouring data points at a time are merged as a data sample for which classification rate and average uncertainty value are calculated. In other words, predictions with similar uncertainty values are grouped together and classification rate for each group is determined. This results in a set of samples that can be used to model classification rate as a function of uncertainty.

Because of a non-linear relationship between uncertainty and classification rate, an SVM epsilon regression model was chosen to model the relationship. Default gamma ($1/N_{variables}$, i.e. 1 in this case) and epsilon (0.1) parameter values were used as they result in a good fit without over fitting. The gamma parameter can be tuned to fit the data at hand [6], but in our experience, with just one predictor variable, this tends to result in overfitting. As suggested in [7], the epsilon parameter value was fixed, as stated above, and the cost parameter value was selected from values on the log_2 scale between 2^{-2} and 2^{11} with 10-fold cross validation selecting the value resulting in the lowest mean squared error between the folds.

Figure 2 summarizes the proposed algorithm for building the model for classification confidence estimation. This model can then be used to estimate the confidence of previously unseen data.



(a) Flow chart of the algorithm to estimate instance level prediction confidence. U stands for uncertainty an CR stands for classification rate.

(b) The resulting support vector regression model $f(U) = CR$.

Fig. 2 The algorithm to estimate classification confidence.

4 Experiments

To validate the confidence model produced by the algorithm, an experiment was set up as follows. Different data sets were divided into separate training (70%) and test data sets (30%). The training data set was used to both train a classifier and to train a model for estimating classification confidence as described above. 200000 data points were generated for building the confidence model. They were then grouped in samples of 200 data points resulting in 1000 samples that were used to fit the confidence model. The test data points were predicted using the trained classifier and the confidence estimates, i.e. predicted classification rates, were also calculated for the test data points. Naive Bayes classifier was selected because of its robustness to missing values, overall good performance, and the ability to estimate a posteriori probabilities.

The tests were repeated with five data sets: Mushroom [8] with either 25%, 50%, or 75% of data values missing completely at random (MCAR), Iris [9] with 50% MCAR, and Wine [10] with 50% MCAR. The data sets were chosen to represent different characteristics. Iris and Wine have a small number of data points but they differ in the number of features. Mushroom, on the other hand, has much more data points. These differences in data set characteristics are interesting here because the proposed algorithm needs to generate a large number of data points to fit the confidence model. All the data sets also have relatively even class priors so that estimated classification rate is a sensible measure of classification confidence.

Data values were removed to artificially increase the uncertainty of the predictions so that the predictions would contain data points from a wider spectrum of uncertainties. With one of the data sets, increasing the number of missing values was used to evaluate the performance of the proposed algorithm in easier and more challenging conditions for the classifier. The test was simulated with a different split to training and test data sets until at least 200000 predicted data points for each data set were generated. The predicted data points were then sorted based on their confidence. Using this order of the data points, 200 neighbouring, i.e. with similar confidence values, data points at a time were merged as a data sample for which classification rate and average confidence value were calculated. This resulted in at least 1000 samples per data set consisting of their true and predicted classification rates. A linear regression model was fit to these data to study how well the developed algorithm performed.

To investigate how much chance affected the fit of the confidence model, the training was repeated several times with a different split into training and test data sets and the resulting SVM regression models were compared.

5 Results

Results of the experiment investigating the effect of chance to the regression model fit can be seen in Figure 3 where the experiment was repeated with 10 different splits with the Mushroom data set when 50% of the data had been removed. The width of the confidence interval and the maximum deviation from the mean reflect the number of training samples residing in that area of input space. With a high percentage of the uncertainty values residing close to zero in this case, the variation between models from different splits is minimal in this area but increases significantly as the density of training data decreases. In other words, the model is more robust in areas of input space where there is more training data available.

The linear models fitted to the resulting predicted and true classification rates from each of the data sets can be seen in Figure 4. The intercept and slope terms of the models as well as statistics describing the model qualities are presented in Table 1.

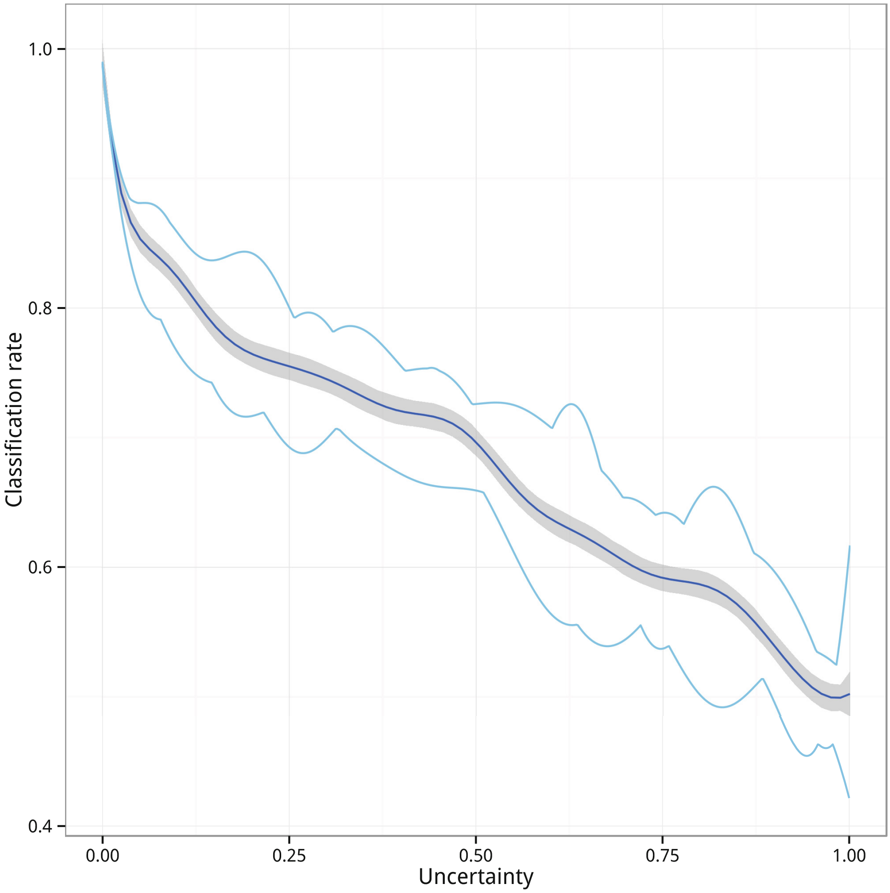


Fig. 3 SVM regression model of the classification rate as a function of uncertainty within the training data set: mean, 95% confidence interval, minimum, and maximum of 10 different training and test data set splits.

The predicted classification rates are good estimates of the true classification rates. This can be seen from the intercept and slope terms which are close to the ideal model (0 and 1, respectively) and from the high percentage of the response variable variation that is explained by the models (R^2). Residual standard deviation (σ) of the models are small indicating that the predicted values do not vary much around the mean. The variation was also approximately normally distributed. These properties are clear also in the figure. Data set sizes vary from a mere 150 data points to thousands of data points but this has no obvious effect on the results.

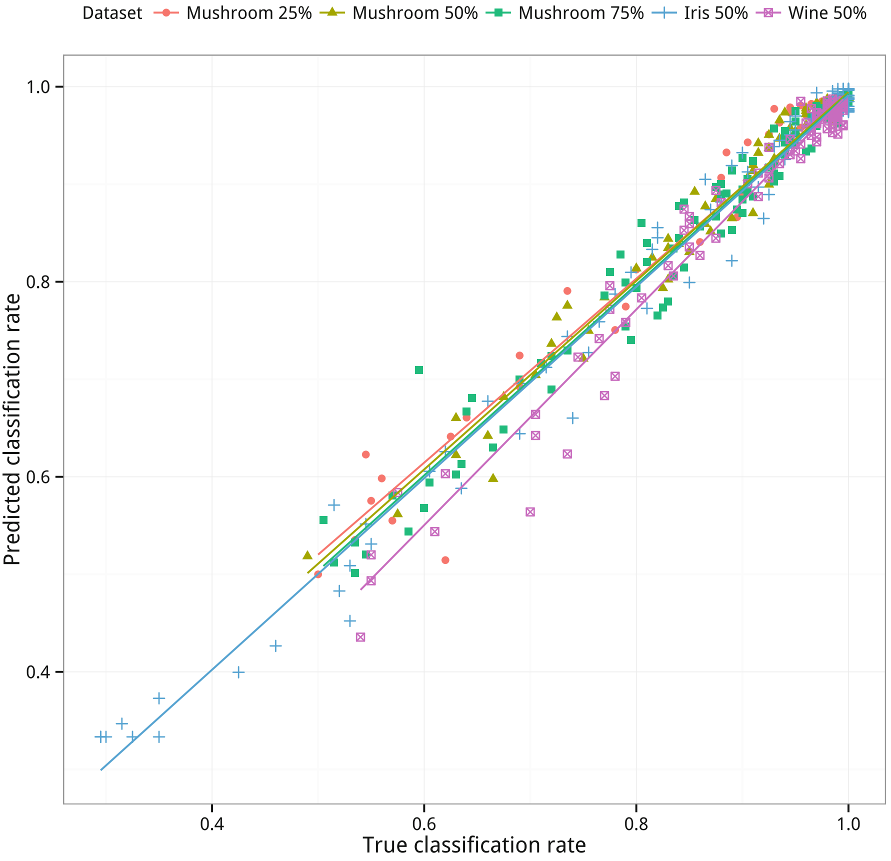


Fig. 4 Predicted and true classification rates of the test samples.

Table 1 Linear model intercept and slope terms and R^2 and σ of the models for predicted versus true classification rate.

Data set	Size	Intercept	Slope	R^2	σ
Mushroom 25% MCAR	8124	0.04860	0.94324	0.9735	0.01898
Mushroom 50% MCAR	8124	0.02746	0.96689	0.9784	0.01549
Mushroom 75% MCAR	8124	0.01459	0.97811	0.97	0.02345
Iris % 50 MCAR	150	0.00939	0.98195	0.9868	0.02382
Wine % 50 MCAR	178	-0.1131	1.10589	0.9687	0.023

6 Conclusions

In this paper, we have presented an algorithm to estimate the confidence of classification results on instance level that is consistent between different data sets, unlike a posteriori derived uncertainty. Our novel algorithm presents the confidence as predicted classification rate, i.e. as a probability that the prediction is correct, so that the results are intuitive to interpret. The algorithm worked well on all five tested cases and the performance was not affected by the size of the data set.

Acknowledgments The authors would like to thank InfoTech Oulu, Jenny and Antti Wihuri foundation, and Tauno Tönning foundation for financial support of this work.

References

1. Cheetham, W., Price, J.: Measures of solution accuracy in case-based reasoning systems. In: 7th European Conference on Case- Based Reasoning (ECCBR 2004), pp. 106–118. Springer (2004)
2. Delany, S.J.: Generating Estimates of Classification Confidence for a Case-Based Spam Filter. In: Proceedings of the 6th International Conference on Case-based Reasoning, pp. 177–190 (2005)
3. Orozco, J., Rudovic, O., Roca, F.S., Gonzalez, J.: Confidence assessment on eyelid and eyebrow expression recognition. In: 8th IEEE International Conference on Automatic Face Gesture Recognition, FG 2008, pp. 1–8, September 2008
4. Gonçalves, L.M.S., Fonte, C.C., Júlio, E.N.B.S., Caetano, M.: A method to incorporate uncertainty in the classification of remote sensing images. International Journal of Remote Sensing **30**(20), 5489–5503 (2009)
5. Alasalmi, T., Koskimäki, H., Suutala, J., Röning, J.: Classification uncertainty of multiple imputed data. In, : IEEE Symposium Series on Computational Intelligence: IEEE Symposium on Computational Intelligence and Data Mining (2015 IEEE CIDM), Cape Town, South Africa, December 2015
6. Caputo, B., Sim, K., Furesjo, F., Smola, A.: Appearance-based object recognition using svms: Which kernel should i use? In: Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, Whistler (2002)
7. Kuhn, M., Johnson, K.: Applied predictive modeling. Springer (2013)
8. Lincoff, G.H.: The audubon society field guide to north american mushrooms. Technical report (1989)
9. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics **7**(2), 179–188 (1936)
10. Forina, M.: PARVUS - An extendible Package for Data Exploration, Classification and Correlation (1991)

Optimal Design and Deployment of Wireless LANs Based on Evolutionary Genetic Strategy

Tomás de J. Mateo Sanguino and Francisco A. Márquez

Abstract This paper presents a design and deployment tool for wireless local area networks (WLAN) based on the IEEE 802.11 standard. The problem for optimal WLAN deployment has been addressed using an evolutionary genetic strategy. The algorithm starts with an initial population of nodes defined within a map and tries to search the best location for access points (AP). The flexibility of the algorithm allows improving the distribution of APs based on the analysis of the RF signal loss by distance and obstacles as criteria. This algorithm has been fully integrated within a previous tool called WiFiSim, which allows to study various parameters and design problems in the PHY and MAC layers of the OSI model (e.g., the hidden node problem, throughput, channel utilization, frame collisions, delays, and jitter). This enables engineers to have a complete tool for the study, design and deployment of customized Wi-Fi networks.

Keywords Genetic algorithms · IEEE 802.11 modeling · Wi-Fi simulator · Wireless network design · Engineering education

1 Introduction

Nowadays, wireless connectivity is established in all aspects and fields, offering flexibility and mobility to users. With the proliferation of wireless devices in recent years, low cost services combined with high data rate transmission —such as for voice and video streaming— have made WLANs to be applied to different sectors. Thus, the need for large bandwidths makes critical time constraints, and

T. de J. Mateo Sanguino(✉)
Department of Electronic Engineering, Computer Systems and Automatics,
University of Huelva, Huelva, Spain
e-mail: tomas.mateo@diesia.uhu.es

F.A. Márquez
Department of Information Technologies, University of Huelva, Huelva, Spain
e-mail: alfredo.marquez@dti.uhu.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_31

requires robust and reliable networks to meet the needs for throughput. Consequently, the optimal design and implementation of WLANs is required for the proper transmission of information and to ensure that business needs are adequately covered [1]-[4].

In previous works [5, 6] the authors presented a new educational tool designed for the generic simulation and study of WLANs, the Wireless Fidelity Simulator (WiFiSim). This tool improved the teaching and learning of courses on computer networking by means of the simulation of the behaviour and performance of communication protocols based on the IEEE 802.11 standard. The interest of this educational tool —tested by teachers, students and professionals on computer networks— lies in the realism of its simulations, which provide a high level of interactivity and visual information with easy-to-interpret results through a configurable and intuitive GUI.

At present, there is a large number of commercial tools to address the design of wireless networks. This includes WiTuners^{TM} [7], AirMagnet Survey$^{®}$ [8], ZonePlanner^{TM} [9], Ekahau Site Survey^{TM} [10], Red-Predict [11], LANPlanner$^{®}$ [12], RingMaster$^{®}$ [13], Cisco WCS$^{®}$ [14], and Cindoor [15], among others. Despite of the considerable catalogue on tools, many of them lack an educational part to assist designers. That is, a specific module for the study and simulation of the network operation.

In this paper we propose a further step beyond the study and design of WLANs: the problem of optimally deploying APs to cover wireless clients within a network. To achieve this goal we propose an optimization algorithm based on evolutionary strategy, thus allowing to solve the weaknesses of the current study and design tools in the state-of-the-art. To this end, we have integrated the developed module into WiFiSim, now called WiFiSim Extension. On the one hand, it facilitates designers the implementation process and automatically provides optimal solutions closer to the best network design. On the second hand, the tool allows at the same time to study WLANs and avoid operation problems (*e.g.*, the hidden node problem, throughput, channel utilization, frame collisions, packet delay, queue length and delay, medium access delay, and jitter).

To do this, Section 2 describes the problem for the optimization in WLAN deployment. Section 3 is devoted to describe the model developed for the optimization of AP locations based on a genetic algorithm. Section 4 develops a real case study and finally Section 5 presents some concluding remarks.

2 Deployment of WLANs: The Optimization Problem

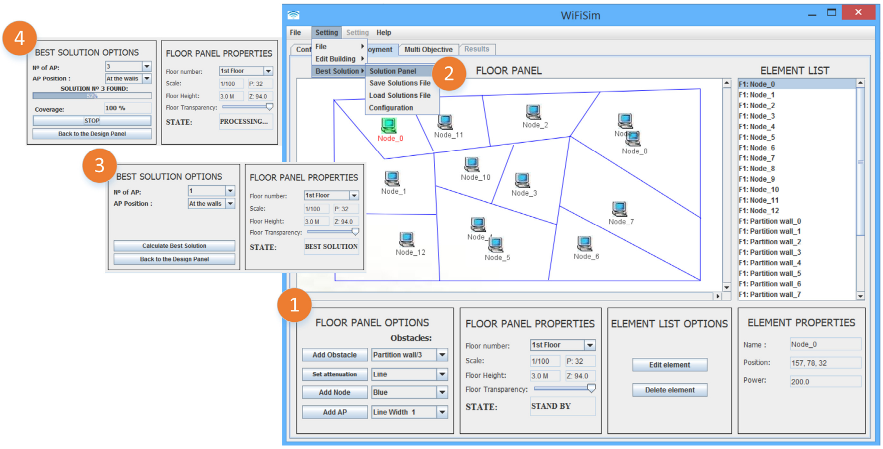
There are many elements that need to be taken into consideration when undertaking the difficult task of designing a WLAN (*e.g.*, environment configuration, power coverage, number of users, flow data rate, site specific user demands, etc.). Consequently, the multitude of parameters that influence the WLAN design requires a wide study.

In order to overcome the demanding tasks imposed by a network engineer, an extension to support the genetic algorithm into the original WiFiSim tool was developed in Java^{TM} with the Eclipse framework. WiFiSim is a software for the design and study of WLANs, not a tool for network deployment [5, 6]. WiFiSim Extension allows designers to graphically describe the environment where the WLAN is to be deployed and specify the user constraints. It is, at this point, where the new module takes care of automatically generating a set of possible AP positions to satisfy the user demands.

2.1 Environment with User-Demand Specifications

The most basic requirement prior to accomplish any automatic design and optimization task is to specify where the WLAN should be deployed. This essentially consists in defining a building by its floor structure with minimal effort (*i.e.*, rooms, doors, corridors, windows, etc.). Each floor plan is used as an input for a RF model that computes the electromagnetic propagation throughout the environment. The WLAN performance is highly dependent on its environment configuration, among other parameters. Therefore, the WLAN planning has a major impact on the network throughput. For increased realism, walls can comprise various materials that influence signal attenuation as part of a large database (*e.g.*, partition walls, concrete walls, joist, etc.). Hence, the WLAN environment can be conveyed by a combination of walls, roofs and even obstacles (Fig. 1).

The next step is to specify additional constraints on the WLAN (*e.g.*, number of APs, power signal, and position at walls or anywhere). As a complement, target areas and restricted areas can be defined by special materials or masks to specifically study areas of interest (*e.g.*, hospitals, offices, shopping centres, etc.) where WLANs may interfere with other devices or suffer from security issues.



Nº of AP: 3
AP Position : At the walls
SOLUTION Nº 3 FOUND: 52%
Coverage: 100 %

Floor number: 1st Floor
Scale: 1/100 P: 32
Floor Height: 3.0 M Z: 94.0
Floor Transparency:
STATE: PROCESSING...

Nº of AP: 1
AP Position : At the walls

Floor number: 1st Floor
Scale: 1/100 P: 32
Floor Height: 3.0 M Z: 94.0
Floor Transparency:
STATE: BEST SOLUTION

F1: Node_0
F1: Node_1
F1: Node_2
F1: Node_3
F1: Node_4
F1: Node_5
F1: Node_6
F1: Node_7
F1: Node_8
F1: Node_9
F1: Node_10
F1: Node_11
F1: Node_12
F1: Partition wall_0
F1: Partition wall_1
F1: Partition wall_2
F1: Partition wall_3
F1: Partition wall_4
F1: Partition wall_5
F1: Partition wall_6
F1: Partition wall_7

Partition wall/3
Line
Blue
Line Width 1

Floor number: 1st Floor
Scale: 1/100 P: 32
Floor Height: 3.0 M Z: 94.0
Floor Transparency:
STATE: STAND BY

Name : Node_0
Position: 157, 78, 32
Power: 200.0

Fig. 1 Example of an environment designed with WiFiSim Extension

2.2 Grid Structure for AP Candidates

Although WiFiSim Extension allows to manually insert several APs in a scenario, it is not feasible to expect the designer suggests all the possible candidate positions with success, especially in complex environments. To automatically find a solution for the WLAN design problem, all the possible combinations for the required APs should be evaluated. The potential search space for such a problem is enormous and the required computational time for such an approach would be unacceptable, especially in large installations. A study on the different techniques used for this optimization problem and the justification for why using a metaheuristic approach can be found in [18].

To automatically generate global optimal solutions, our algorithm considers the geometry of the environment and the specific user constrains. The algorithm is based on a cell structure where objects (*i.e.*, APs, nodes and obstacles) not only have a physical position but also an associated RF signal power. This information is provided from WiFiSim to the optimization process and used as part of the genetic algorithm to evaluate the quality of a suggested solution. In order to perform the grid structure, Wifisim Extension computes the signal attenuation by distance and obstacles for each cell, and the reachability between nodes for each possible location of the APs.

Left side in Figure 2 shows an example of a square floor plan according to a 1:50 scale where the tool computes the solution for the AP candidates at the walls. These are colored in red, whilst the pink circle is an optimal solution, green cells stand for areas where the wireless nodes need signal coverage, and white cells are non-accessible areas. It is assumed that the client devices are evenly distributed within the requested area.

3 Evolutionary Genetic Model for Optimal Deployment

An approach based on a coding scheme has been exploited for one of the most used genetic algorithms (GA), a steady state GA [16, 17]. The basic principle is to begin with a selection of candidate solutions or individuals (*i.e.*, positions for a specific number of APs) and evolve these solutions by using selection, crossover, mutation and replacement with subsequent generation improvements.

The implemented algorithm automatically generates two grids. One grid with the possible physical positions throughout the environment where an AP can be placed. A second grid with the power coverage and level of each cell for all the possible placements of APs based on the PHY layer of the IEEE 802.11a/b/g/n standard used and the signal loss due to obstacles and distance. Thus, each candidate AP not only has a physical position but also an associated coverage intensity map generated through the appropriate propagation model.

Steady State Genetic Algorithm. The specific GA implemented for this research operates in steady state as opposed to generational GAs. The main difference is the number of members after each generation that are evaluated by the target function. For a generational GA, all of the members of a population are evaluated. Their fitness values are evaluated, sorted and then each member of the population is replaced by a crossover and mutation tournament scheme. All of the new members are evaluated again by the target function, except perhaps for the best performer if an elitism strategy is employed. For a steady state GA, after evaluating the initial population by the target function, only one member is replaced by a tournament scheme using crossover and mutation for each generation. That new member is evaluated by the objective function and then it replaces the worst performer of the previous generation. This can lead to quick and very accurate convergence since that member immediately becomes part of the mating pool making possible an early shift toward an optimal fitness [17].

The main drawback of the steady state GA is that it does not have the large number of random guesses that the generational GA can obtain. For the steady state GA only one member of the gene pool is replaced for each generation and that member is the worst performer of the previous generation. Since the member that is created is composed by two good members from the previous generation, the steady state GA can quickly converge to a good solution. However, if none of the members of the initial population are good members the steady state GA can only rely on the mutation of one member each generation to find a good fitness. For this reason, a specific strategy of selection and replacements, and two types of mutation were included to enable more exploration or exploitation for each new member.

Coding Scheme, Population Size and Initial Population. Before starting, the algorithm must know the details of the chromosome such as for example its size (right side in Fig. 2). In this paper, we used an integer coding scheme where m is the number of parameters I_m (*i.e.*, the number of APs requested by the user). Each integer takes values in the interval [1, n], being n the number of possible AP positions.

$$\mathrm{C} = (I_1, \ldots, I_m) \mid I_m \in \{1, n\} \tag{1}$$

The population size will be generated as a function depending on the size of the building. This step is done before executing the algorithm as follows:

$$\mathrm{S} = (a \,/\, m) * k \tag{2}$$

where a stands for the number of accessible cells by APs, m stands for the chromosome size (*i.e.*, the number of APs that users request), and k is a multiplication factor. In our case, this value was set to 3 and determined through experimentation. Once the population size is determined, the next step is to initialize the chromosome of the GA. The initial population obtains all genes randomized into the interval [1, n].

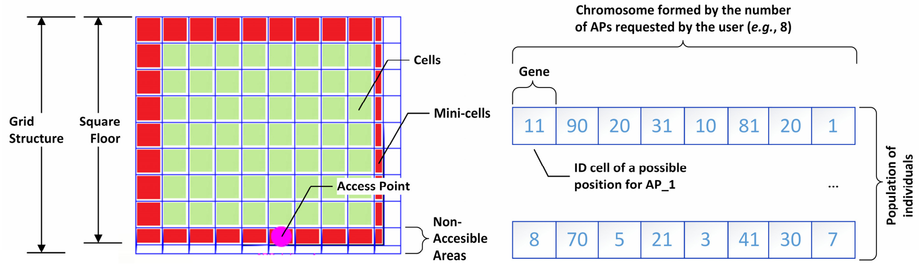

Grid Structure
Square Floor
Cells
Mini-cells
Access Point
Non-Accesible Areas
Chromosome formed by the number of APs requested by the user (e.g., 8)
Gene
11 90 20 31 10 81 20 1
ID cell of a possible position for AP_1
...
8 70 5 21 3 41 30 7
Population of individuals

Fig. 2 Example of the grid structure (left) and chromosome (right) used to compute candidate positions for APs

Objectives. The targets to maximize the algorithm are the followings: *fitness_1*, which consists of the sum of all the cells covered; *fitness_2*, meaning the sum of the power level reaching the cells. To determine the best individual of the population is as follows: one individual with better *fitness_1* will be the best. If more than one individual has the same *fitness_1*, the tiebreaker will be based on the best *fitness_2*.

Selection. A tournament selection is used in order to select the parents.

Crossover and Mutation Operators. The crossover operator employed is a one-point crossover. We keep in mind that a gene cannot be repeated in a chromosome. This way, two individuals are obtained by combining the two offspring generated from parents. For each of them, the mutation operator changes a gene value at random with probability P_m and determines other possibility in [1, *n*].

The genetic algorithm can use two possible types of mutations indicated by a user-defined parameter. With the first mutation (*i.e.*, disruptive) the process will be conducted with a probability of 2%. The second mutation (*i.e.*, non-disruptive) makes a special mutation with a probability of 10%. In this case, a random gene of the chromosome is selected to change its value by a random value of ± 5% within the range [1, *n*].

The system determines two possible types of replacements also indicated by the user configuration: 1) by parents, where two children and two parents are evaluated and compete between them in order to keep the best individuals within the population; 2) by tournament, where the children obtained will compete against the two worst individuals to keep the best two. The mechanism of selection, mutation and replacement will allow managing the convergence of the algorithm (*i.e.*, exploration and exploitation). The algorithm should stop after a number of iterations set by the user or by a stop button in the simulation tool.

4 Case Study for the Deployment of a Wi-Fi Network

The practical case regarding the optimum WLAN deployment was conducted in a typical office environment consisting of a 73 x 40 m^2 building with rooms, corridors and halls (Fig. 3). The idea was to compare the proposed design obtained against the actual design. According to the floor plan, the building has three Cisco Aironet 1100 APs (*i.e.*, red circles in the picture) configured for the OpenUHU wireless LAN.

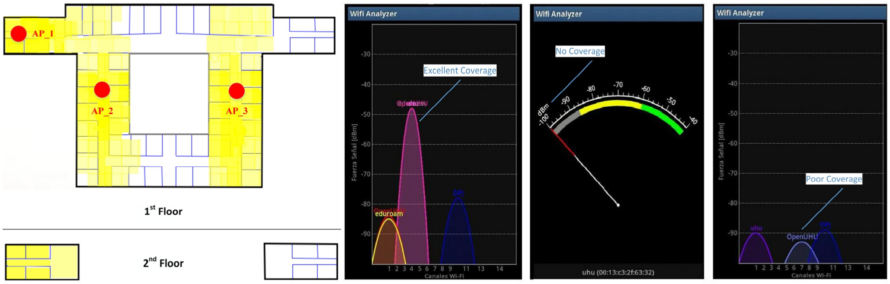


Fig. 3 Methodology to obtain the real coverage of the building and the power levels

An analysis on the APs distribution, their power levels and the type of obstacles involved (*e.g.*, a loss signal of 95% because of external contour walls and 8% due to internal Pladur® walls) was conducted to obtain a coverage map. The followed methodology consisted in using the Wifi Analyzer application to obtain the power levels at the building cells. They were classified from areas with excellent power signal (left plot) to areas with no coverage (central plot). Some other areas were covered even if the signal arrived with a low/medium power (right plot). As a result, we obtained a total coverage of 70.85% stating that the APs were not placed in the best location possible, therefore we faced a new WLAN design after a non-optimal solution.

To this end, we conducted different tests with the WiFiSim Extension module on the experimental environment using a 1:100 scale, thus resulting in 132 cells and 24 mini-cells. The tests consisted in finding better distributions for a WLAN composed by three and four APs at the walls (Fig. 4). The selected parameters were APs with the IEEE 802.11a/b technologies, both types of disruptive and non-disruptive mutations, type of replacement set for parents and tournament size of 5%.

The experiments were completed with an Intel® Core™ i7 (2.6 GHz, 16 GB RAM) under Windows 64-bit operating system and the execution times were between 2s and 20s, which increases exponentially for larger scales. The criterion to stop the simulation was to press the stop button when reached a result with no better coverage afterwards. In the worst case we obtained a solution with a coverage of 80% for three APs. Although this distribution reached more signal levels than the actual WLAN installation, it was not enough to reach all the areas of the building. So we tried different combinations of the GA settings to achieve better

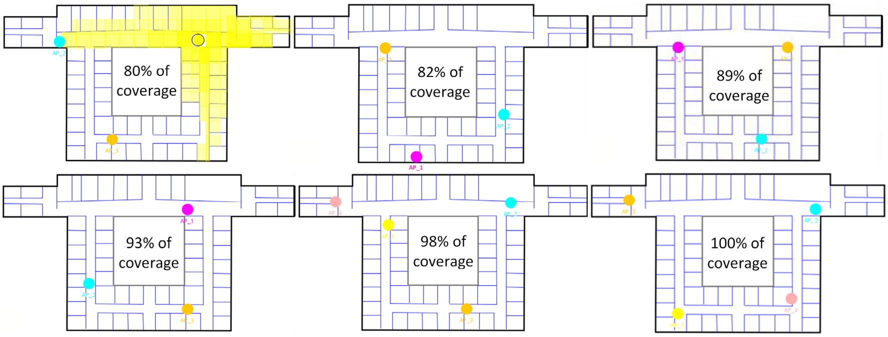


Fig. 4 Solutions with different combinations of the steady state GA for three and four APs

distributions by: 1) improving the APs deployment (82%, 89% and 93%), and 2) increasing the number of APs (98% and 100%). This suggests that although the algorithm may not always find the best solution, it would be very close to the optimum in most of the cases.

5 Conclusions

Wireless communications have been widely imposed over conventional wired networks thanks to advantages such as the low cost, easy installation, mobility and range. However, Wi-Fi networks also have disadvantages such as the considerable decrease in signal power due to technology, distance and obstacles.

This work is focused on the study of these factors to provide adequate and guaranteed designs in the planning, optimization and deployment of WLANs. To do this, we have implemented a complete design tool that obtains sub-optimal solutions very close to the best —by using a genetic algorithm— considering the largest covered area and maximum strength signal as decision criteria.

After an experimental analysis carried out on a real environment, the results show that our tool can be useful not only in the fields of engineering education but also for professionals in networking. That is, to consider how to optimally distribute APs and provide better coverage throughout a building. Videos and a downloadable free version of the tool are available at www.uhu.es/tomas.mateo/wifisim/wifisim.htm.

References

1. Rodrigues, R.C., Mateus, G.R., Loureiro, A.A.F.: On the design and capacity planning of a wireless local area network. In: Proc. NOMS, pp. 335–348 (2000)
2. Chandrashekar, K., Janes, P.: Optimal design of wireless local area networks (WLANs) using simulation. In: Proc. 28th IEEE MILCOM, pp. 1–7 (2009)
3. Vasudevan, S.: A simulator for analyzing the throughput of IEEE 802.11b wireless LAN systems. Master thesis. Virginia Polytechnic Institute and State University (2005)

4. Reineck, K.M.: Evaluation and comparison of network simulation tools. Master thesis. University of Applied Sciences Bonn-Rhein-Sieg and Fraunhofer-Gesellschaft (2008)
5. Mateo Sanguino, T.J., Serrano López, C., Márquez Hernández, F.A.: WiFiSiM: an educational tool for the study and design of wireless networks. IEEE Trans. Education **56**(2), 149–155 (2013)
6. Mateo Sanguino, T.J., Márquez Hernández, F.A., Serrano López, C.: Evaluating a computer-based simulation program to support wireless network fundamentals. Computers & Education **70**, 233–244 (2014)
7. Kruys, J., Qian, L.: RF spectrum, usage and sharing. In: Sharing RF Spectrum with Commodity Wireless Technologies, ch. 3, pp. 3–14. Springer, Netherlands (2011)
8. Phifer, L.: Managing WLAN Risks with Vulnerability Assessment. White Paper. Core Competence, Inc. (2008)
9. Wireless, R.: Smart Positioning Technology (Rel. 1.4). Technical Report (2012)
10. Vanhatupa, T.: Wi-Fi Capacity Analysis for 802.11ac and 802.11n - Theory & Practice. White Paper. Ekahau, Inc. (2013)
11. Belloul, B.: Smart Meter RF Surveys – Final Report. Technical Report. Red-M (2012)
12. Andrusenko, J., Burbank, J., Ward, J.: Modeling and simulation for RF propagation. In: IEEE Global Communications Conf., Design & Development Forum (2009)
13. Arendt, D., Morawski, M., Zajaczkowski, A.: Planning of the 802.11/abgn computer networks with the aid of the specialized web service. In: Towards Modern Collaborative Knowledge Sharing Systems, ch. 9, pp. 157–171. Springer, Heidelberg (2012)
14. Arfe, A., Deguy, P., Guillot, L., Le Guilly, T., Louge, R.: Android Application for Aalborg University. Master Thesis, Aalborg University (2011)
15. Torres, R.P., Valle, L., Domingo, M., et al.: CINDOOR: an engineering tool for planning and design of wireless systems in enclosed spaces. IEEE Antennas and Propagation Mag. **41**(4), 11–22 (1999)
16. Holland, J.H.: Adaptation in natural and artificial systems. The MIT Press (1992)
17. Vavak, F., Fogarty, T.: Comparison of steady state and generational genetic algorithms for use in non-stationary environments. In: Proc. IEEE Intern. Conf. Evolutionary Computation (1996)
18. Gibney, A.M., Klepal, M., Pesch, D.: Agent-based optimization for large scale WLAN design. IEEE Transactions on Evolutionary Computation **15**, 470–486 (2011)

Minaturized Safety PLC on a Chip for Industrial Control Applications

Ali Hayek and Josef Börcsök

Abstract In this paper a safety programmable logic controller as a single chip solution is presented. The presented control system is based on a certified application specific integrated circuit for safety-critical applications according to the safety standard IEC 61508 second Edition, meeting the safety integrity level SIL3. Furthermore, SIL3 compliant operating system and middleware are also briefly presented in this paper. A further main focus of this paper is led on the graphical programming in application development is provided by graphical integrated development environment for configuring and programming the safety controller. The presented safety solution is freely configurable and programmable using graphical interface and achieves flexible, compact, low-cost and yet advanced process control and automation for safety related applications.

Keywords Safety systems · Control systems · Real-time systems · Systems-on-chip

1 Introduction

Programmable logic controllers (PLCs) [1] have been a principal part of industrial process control for decades. Especially in safety-related applications, it is of great importance to integrate robust, reliable and safe PLCs. Typically, standard PLCs used in safety-related applications are configured in a redundant performance. A primary PLC for the execution of the main functionality and a redundant PLC which is used as monitoring element to support a safe and orderly shutdown in the failure case. Additionally, further redundant system elements and several monitoring modules are needed. In summary, designing safety-related systems using

A. Hayek and J. Börcsök(✉)
Chair for Computer Architecture and System Programming, Department of Electrical Engineering and Computer Science, University of Kassel, Kassel, Germany
e-mail: {ali.hayek,j.boercsoek}@uni-kassel.de

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_32

standard PLCs requires tremendous additional design, validation and certification efforts. Therefore, specific safety PLCs are nowadays available that considerably reduce these efforts. Safety PLCs provide a complete certified safety-related solution for the use in several industrial applications.

Driven by the ongoing challenge to reduce the overall system size and power consumption and especially the system costs, a lot of engineering efforts were being invested in miniaturizing systems into single chip solutions. Most recently, due to the further development of safety standards, such as the standard IEC 61508 ([2] and [3]), safety-related solutions on a single chip are now the latest state of technology in the field of safety-related applications.

In this paper, a single chip safety PLC solution is presented, which addresses the safety integrity level SIL3 according to the standard IEC 61508 at hardware and software level. The hardware design of the chip architecture was designed according to the second edition of the standard IEC 61508 for safety-related integrated circuits with on-chip redundancy, as already published in [4]. Since software and application platform are key elements of every safety-related system, an appropriate approach for the different needed layers is presented. As shown in Fig. 1, the lowest layer is the SIL3 certified hardware platform without any additional software. A complying safety-related operating system is the first software layer which mainly performs the basic tests of the hardware modules. The next proposed layer represents a safety-related middleware which handles all low-level access to the target platform for the user application ([5] and [6]). More recently, PLCs are programmed using graphical application software. Therefore, the next layer in the proposed approach is the graphical integrated development environment (IDE) conforming to the standard IEC 61131 [7].

The key feature of the proposed approach is to free the developer of PLC applications from highly complex safety-related details. From developer´s point of view accessing safety PLCs at hardware low-level is given, but it demands detailed knowledge in safety-related programming and all related safety requirements. Due to the middleware with underlying safety-related operating system the user is not intended to take care of safety-related aspects and is enabled to write safety-related software nearly as straightforward as non-safe applications. With the proposed graphical IDE, user applications might be programmed similarly to standard PLCs. Thus, using the whole package, from the target platform up to the graphical IDE, application development is enormously sped up. A safety application still needs to be developed in conformance with appropriate standards, but the effort for the implementation and certification process would be minimized.

The main contribution in this work is adding support for the IDE to the middleware according to the proposed approach to build a safety PLC with all needed functionalities. While the graphical IDE is provided by an industry partner, all other hardware and software layers are designed within own research work with industrial partners.

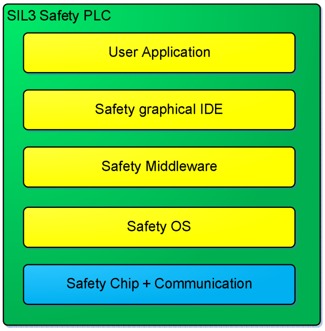


Fig. 1 Complete SIL3 environment

2 Safety Programmable Logic Controller

2.1 Controller Architecture

The target hardware platform of the safety PLC is a miniaturized SIL3 compliant safety system-on-chip which integrates all features of a PLC on a single chip. This reduces the number of required components for safety applications and improves system reliability. A more detailed description of this architecture which has been developed in cooperation with an industry partner can be found in [4]. Fig. 2 gives a general overview about the system architecture of the safety PLC, which consists of two subsystems: a redundant system (safe system) and a single-core system intended for communication (COM system). Both subsystems are connected interference-freely. In addition, both processor systems may trigger an interrupt in the other subsystem. Both subsystems contain processor cores with own data and program memories, digital inputs and outputs, as well as diverse communication interfaces. The COM system act as black channel for safe communication between the safe system and field devices for safety-related applications utilizing its communication interfaces. In the black channel approach it is up to the communication end-points to implement the safety layer [2], the operating system and safe system already provides safe communication.

2.2 Safe Operating System

According to the proposed approach, the first software layer for the safe system is a SIL3 compliant safety-related operating system. The major advantage of using a safe operating system is providing features that will make it easier to design safety into the user application. In the proposed safety PLC the main tasks of the safe operating system for covering SIL3 compliance according to IEC 61508 will be CPU check, memory testing, memory protection and testing of further monitoring elements required for the safety functions. The safe operating system executes these hardware tests at start-up and during runtime. After initialization, it provides a fixed cycle time. All tasks are executed sequentially within one cycle. If cycle time is exceeded, or an error occurs, the operating system initiates the safe state for the target platform. On the black channel side, the COM system has its own non-safe operating system. This can be a standard embedded operating system, and it conducts mainly functions of the communication interfaces.

2.3 Safe Middleware

The safe middleware comprises all safety-related functionality of the target platform, and all low-level access is encapsulated by the middleware. The middleware is built up in modules and each independent module controls a single functionality of the target platform. These modules represent the inputs and outputs, as well as

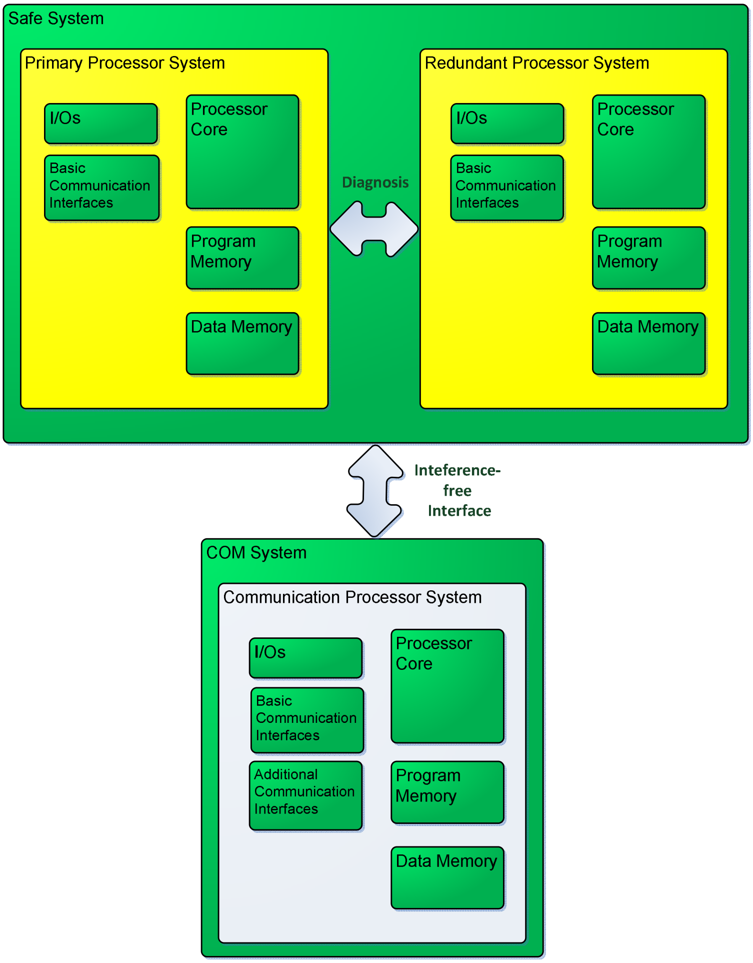

Safe System
Primary Processor System
I/Os
Basic Communication Interfaces
Processor Core
Program Memory
Data Memory
Diagnosis
Redundant Processor System
I/Os
Basic Communication Interfaces
Processor Core
Program Memory
Data Memory
Inteference-free Interface
COM System
Communication Processor System
I/Os
Basic Communication Interfaces
Additional Communication Interfaces
Processor Core
Program Memory
Data Memory

Fig. 2 Safety PLC block diagram

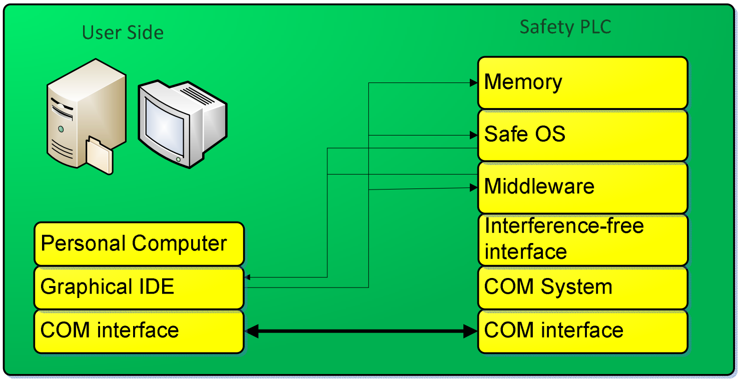


Fig. 3 Interaction between User and the Safety PLC

peripheral communication of the safe system. Thus, the modules represent low-level drivers for all safety-related functionalities. A detailed treatise of the concept and validation of the SIL3 compliant middleware is presented in [5] and [6]. The required program memory is kept as small as possible by only linking modules which are required by the application. This is important, as the middleware is designed to be used with a miniaturized safety PLC with a compact CPU core.

The middleware is called in each cycle of the safe operating system. At first call, instances of all modules, used by the application, are created. Only one instance is allowed for each module in order to prevent concurrent access to registers of the target platform. In addition to function calls by the application, each module comprises a function that checks safety aspects related to this module. This function is called at least once per cycle by the middleware (but may also be called by the user application). The middleware modules ensure that all parameters used for function calls are within the specified ranges. Otherwise the middleware request the safe operating system to enter the safe state.

3 Graphical IDE

In this section the graphical user interface for the programming of the safety PLC is presented (detailed literature in [8]). First, the main target group of end users is introduced. This is important, to identify possible impacts. Furthermore, a detailed description of the implementation and features of the graphical IDE is presented. Finally, the results are shown with a couple of simple examples in order to prove the feasibility of the proposed approach.

3.1 Target Group of End Users

Designed to meet SIL3 requirements according to the standard IEC 61508 the proposed safety PLC solution is intended to be used for any SIL3 application. In this context, the target user group of interests includes every company, institution or service provider who targets a SIL3 certification conforming to the standard. However, many end users are daunted by the prospect of having to comply with the lengthy and complex nature of such certification processes, especially due to the time-to-market aspects. End users are interested in having a SIL3 certified system with wide-ranging features and rapid development process. However, with the proposed approach provides a safety chip as a hardware platform with safety and communication features. Once these features are sufficient for the user´s application, the chip solution can be targeted as a safety PLC solution. Normally end users will access to the hardware using the middleware which is modelled in C++. However, this requires engineers or technicians with good C++ background, as well as time and engineering efforts to understand the internal system structure, such as: Cycle time of operating system, error handler of hardware and software components or SIL3 certification of the application code itself. In this context, the proposed graphical IDE layer in our approach aims to present a user-friendly interface, where the issues mentioned above are carried out by the different levels of the presented approach. Thereby a PLC engineers will not need to use C++ and can use a PLC familiar programming interface.

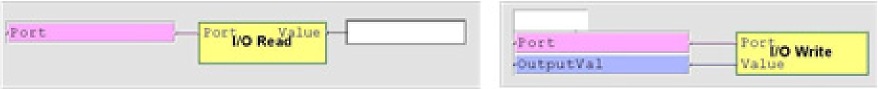


Fig. 4 Graphical IDE function blocks for I/O READ and I/O WRITE

The graphical programming concept is based on function blocks. A function block represents a single functionality of the safety PLC, such as AND gate or an INPUT. A list of function blocks is stored in dedicated libraries using function blocks, and the user can design a system behavior according to his needs. After designing the system, the function block diagram is compiled and downloaded to the target platform.

3.2 Structure and Functionality

In this subsection, the interaction between the graphical IDE and the lower layers is described, as shown in Fig. 3. Here is important to mention, that the complete handling of the communication with the graphical IDE is performed using the middleware. Another IDE from other parties can be ported through the same process. The graphical IDE is usually installed on a PC. The communication between the PC and the safety PLC is depending on the needed communication interface. In case of the presented system, both serial interface UART and Ethernet are possible. The main tasks of the graphical IDE are downloading the generated code to the safety PLC and performing specified online-tests along with the programming of the application. A more detailed explanation of online-tests will be given in the following subsection. In the following, the user side and safety PLC side in Fig. 3 are described in details.

User Side

As already mentioned above, the end user should only know about the function blocks. These blocks are designed for the specific safety PLC and are linked with C++ classes of the middleware. Each module or functionality of the PLC is represented by a class with own methods and attributes. For example an INPUT/OUTPUT module is represented by two blocks I/O-READ and I/O-WRITE for specifying the desired block type as shown in Fig. 4. Before running the application, all blocks are analyzed and the graphical IDE will link the application program and a precompiled middleware library. Afterwards, the generated machine code will be downloaded to the hardware platform. Before that, the graphical IDE performs checksum calculation and adds a timestamp for safety and reliability aspects. Via communication interface the complete data package is sent to the safety PLC.

Safety PLC Side

On the safety PLC side, messages from the graphical IDE will be sent to the safe system interference-freely via the COM system. There, the middleware will perform the checksum calculation and store the timestamp, and the machine code will be stored and executed. In this regard, it should be noted that the safe operating system is not loaded with the application. It is loaded and tested regardless of the graphical IDE being ready to receive and check the application program. It should be mentioned in this context that the memory areas of the safe system for safe operating system, middleware and user application are separated, in a way that they can be updated without interfering other parts.

3.3 Online-Tests

An important module for the operation and monitoring of industrial systems is the integration of simulation and test modules into the graphical IDE. With the integrated online-test option the user is allowed to check and manipulate the state of dedicated modules of the safety PLC during run-time using the graphical IDE. The example in the next section will show more about the execution of the online-tests.

3.4 Simple Example Applications

After presenting the structure and functionality of the graphical IDE a simple example application is given in this section. This example is kept simple to show the technical feasibility of the presented approach. However, all functionalities of the safety PLC are covered by the graphical IDE and can be used as functional blocks. The first example in Fig. 5 shows a simple control of digital outputs using digital inputs. After building the application block diagram using the functional blocks and downloading code to the target hardware, the user can monitor changes on safe digital inputs and outputs. The application program consists of three LEDs (outputs) and three switches (inputs) connected to the safety PLC. The application monitors states of switches and combines states of switches with software input. As result the LEDs are turned on or off. In the function block diagram, the LEDs are represented by the 3 blocks on the right side labeled LED_0 to LED_2. The software switches are labeled var_0 to var_2 and they are clickable. The states of the hardware switches are represented by SW_0 to SW_2. A blue wire represents an inactive state; a red wire represents an active state. Furthermore, the user can change the value of the variable of the application running on the target-platform. He can do that online, i.e. in the IDE while the test is running. In the example above, setting var_0 to "false" would inhibit the propagation of SW_0 to LED_0. In Fig. 6 a second example of a simple RS-Flip-flop is shown. Using the SET and RESET inputs a memory function can be realized.

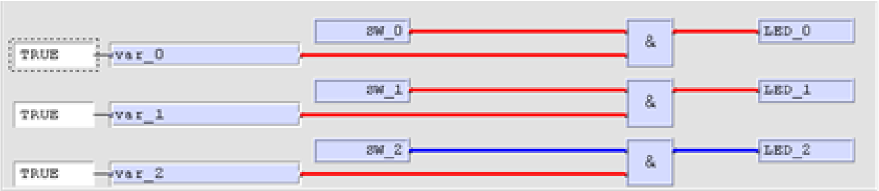


Fig. 5 A part of a sample application for I/O-Control

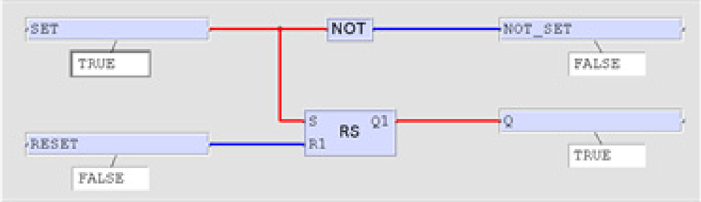


Fig. 6 A sample application for an RS-Flip-Flop

4 Conclusion

In this paper, a single chip safety PLC solution was presented. A complete hardware and software platform for SIL3 industrial applications is introduced. The proposed approach divides the software into several layers: a safe operating system, a middleware and a graphical user interface. An example for a simple application showed the feasibility of the proposed approach. The main advantage of the proposed approach is that it substantially reduces the development time and the complexity of SIL3 applications. Moreover, dividing the software in modular layers designed for safety related applications facilitate the certification process of the overall system.

References

1. Petruzella, F.D.: Programmable Logic Controllers, 5th edn. McGraw-Hill Education (2016). ISBN-13: 978-0073373843
2. International Electrotechnical Commission, IEC, International Standard: 61508 Functional safety of electrical/ electronic/ programmable electronic safety-related systems Part 1-7, Geneva (1999-2010)
3. Lloyd, M.H., Reeve, P.J.: IEC 61508 and IEC 61511 assessments - some lessons learned. In: 4th IET International Conference on Systems Safety 2009 Incorporating the SaRS Annual Conference, pp. 1–6 (2009)
4. Hayek, A., Machmur, B., Schreiber, M., Börcsök, J.: Safety-related system-on-chip architecture for embedded computing applications. In: European Safety and Reliability Association Annual Conference, Amsterdam (2013) (in press)
5. Schreiber, M., Delic, E., Hayek, A., Börcsök, J.: Concept for a SIL3 middleware encapsulating safety-related aspects of applications for an 8051-based SIL3 multi-core system-on-chip. In: 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO), pp. 81–84 (2013). ISBN: 978-953-233-076-2
6. Delic, E., Schreiber, M., Hayek, A., Börcsök, J.: Validation of a SIL3 middleware for safety-related system-on-chips. In: 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO), pp. 85–90 (2013). ISBN: 978-953-233-076-2
7. International Electrotechnical Commission, IEC, International Standard: 61131 Programmable Controllers Part 1-7, Geneva (2000-2013)
8. Delic, E., Löser, K., Schreiber, M., Hayek, A., Börcsök, J.: SIL3 Graphic integrated development environment for a safe system-on-chip. In: Proceedings 12th WSEAS, December 10-12, 2013, Budapest, Hungary. ISBN: 978-960-474-349-0

Random Forest Based Ensemble Classifiers for Predicting Healthcare-Associated Infections in Intensive Care Units

María N. Moreno García, Juan Carlos Ballesteros Herráez, Mercedes Sánchez Barba and Fernando Sánchez Hernández

Abstract Surveillance and prevention of infections acquired in the hospital environment is an important challenge in the current health systems given the great impact of these kind of infections on patient mortality as well as on sanitary costs. Data analysis can contribute to make easier these tasks by means of identification of risk factors and prediction of infection acquisition. This work is focused on the study of infections acquired in intensive care units by means of data mining models. In this context we have to deal with the problem of building reliable classifiers from imbalanced datasets. This is addressed with an ensemble based approach. The aim of the proposal is to overcome some drawbacks presented by other usual strategies.

Keywords Healthcare-associated infections · ICU infections · Data mining · Ensemble classifiers · Imbalanced data

1 Introduction

Healthcare-associated infections (HAI) are mainly acquired over the hospital stay. Between the fourth and fifth part of HAI are diagnosed in intensive care units

M.N.M. García(✉)
Department of Computing and Automation, University of Salamanca, Salamanca, Spain
mmg@usal.es

J.C.B. Herráez
Intensive Care Unit, University Hospital of Salamanca, Salamanca, Spain

M.S. Barba
Department of Statistics, University of Salamanca, Salamanca, Spain

F.S. Hernández
School of Nursing and Physiotherapy, Prehospital Emergency Services,
University of Salamanca, Salamanca, Spain

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_33

(ICU), especially device-associated infection (DAI), those presenting a greater impact on patient progress. All of them are associated to invasive devices altering natural defense barriers and favoring the transmission of pathogens, which often have high rates of antimicrobial resistance and are part of the ICU flora.

Implementation of surveillance and prevention measures have reduced the incidence rate of infections in ICU as well as their adverse effects. To make such measures more effective it is important to manage a great quantity and variety of information and usually this task takes long time to physicians. Data mining techniques can provide support for data processing, not only in order to improve the efficiency but also in order to find valuable patterns in data for prediction, unable to be discovered by using alternative procedures as statistical techniques, which are usually used only to indentify key indicators but not to build predictive models. In this work data mining techniques are used to find out the most important HAI risk factors and to identify patients more susceptible to infections regarding their characteristics, treatment, invasive devices used and other information concerning their stay in the ICU.

The study was carried out with data from 4616 patients gathered in the ICU of the University Hospital of Salamanca (Spain) over the years 2007-2013. Only 311 patients out of the whole group included in the study acquired device-associated infection, which represent 6.7% of the total. The application of classification algorithms to imbalanced datasets as the involved in this work presents serious drawbacks since good global accuracy can be achieved but the precision for the minority class can be low. Oversampling the minority class records or undersampling the majority class records are two common approaches to deal with imbalanced datasets, but they have important drawbacks. Undersampling may discard potentially valuable data, while oversampling artificially increases the size of the data set and, as a result, the computational cost of inducing the models. In addition, the replication of existing examples in the minority class causes overfitting problems [Hulse and Khoshgoftaar].

The aim of the present study is to introduce a new way of addressing the imbalanced data classification by applying the suitable machine learning algorithms, such as ensemble classifiers, in order to overcome the problems mentioned previously.

The rest of the paper is organized as follows: Section 2 presents an introduction to the problem of building classifiers from imbalanced datasets and the most common strategies to deal with it. In section 3, usual measures to validate classifiers in imbalanced data contexts are described. Section 4 encloses the basis of the classification algorithms used in this work, with special focus on ensemble classifiers. Section 5 is devoted to explain the proposed approach and to describe the dataset used in the study. The results are reported in section 6. Finally, the conclusions are given in the last section.

2 Background

Classification from imbalanced datasets has been the focus of intensive research since it represents an important obstacle in the supervised learning. It takes place when there is a big difference between the number of instances of every class under study. In these situations the precision for the minority class is usually significantly lesser than the precision for the majority class; therefore, predictive models are not valid even when they present an acceptable accuracy. That is, the classifier can achieve a high percentage of instances correctly classified but the percentage of instances belonging to the minority class that are correctly classified can be very low. Additionally, the minority class is usually the most interesting and misclassification of its instances is the least desirable.

This problem is very common in medical fields such as studies of mortality [Moreno et al., 2014], about outcome of treatments [Martín et al., in press] or medical diagnosis [Nahar et al., 2013], between others.

There are several methods to deal with this problem. They can be organized in the following categories [López et al., 2013]:

- Data resampling: involves oversampling or undersampling procedures to modify the training set by creating or eliminating instances in order to obtain a balanced distribution of the instances belonging to each class.
- Algorithmic modification: implicates the modification of the learning algorithms to make them more suitable for processing imbalanced data.
- Cost-sensitive learning: takes into account the misclassification costs, so that the different kind of misclassifications are treated in a different way.

Some drawbacks have been described for the three approaches. Oversampling the minority class records or undersampling the majority class records are two common preprocessing methods used to deal with imbalanced datasets, but they have important weaknesses. Removing potentially valuable data is the main drawback of undersampling the majority class, while oversampling the minority class can cause overfitting problems as well as an increasing of the computational cost of inducing the models [Hulse et al., 2007]. The two last approaches are less used due to some difficulties in their application. Adapting every algorithm to imbalanced data demands a great effort and sometimes provide worst results than resampling techniques. On the other hand, cost sensitive learning usually requires domain experts to give values to the cost matrix containing the penalties for the different types of misclassification [López et al., 2013], thus it is difficult to find out the most suitable values.

3 Validation of Classifiers in Imbalanced Data Contexts

Cross validation is the most used method for validating classifiers. It is an effective method for approximating the error that might occur when a classifier is used to classify unlabeled data. In many research works the validation of classifiers is carried out only by means of examining their accuracy, that is, the percentage of correctly classified instances. However, that measure is not appropriate in imbalanced data contexts since in these scenarios machine learning algorithms can achieve an acceptable global accuracy but the precision for the minority class can be very low. Therefore, accuracy can be complemented with other metrics that provide additional error perspectives, especially when evaluating binary decision problems. In these cases, the examples are classified as either positive or negative and the output of the classifier can belong to one of the following four categories: True positives (TP) are positive instances correctly classified, false positives (FP) are negative instances classified as positive, true negatives (TN) are negative instances correctly classified and false negatives (FN) are positive instances classified as negative. Given this information, it is possible to define some validation metrics such as precision, recall, F-measure or area under the Receiver Operator Characteristic (ROC) curve.

4 Classification Algorithms

The algorithms applied in this study were both simple and ensemble classifiers. As simple classifiers, the J48 tree and Bayesian networks were tested. The ensembles used were Random Forest, Bagging, AdaBoost and Random Commitee.

A decision tree consists of a set of conditions that are organized in a hierarchical structure, so that the final decision can be determined following the true conditions from the root to the leafs of the tree. The induction of the tree is carried out by means of a process in which the examples are separated depending on the evaluation of certain conditions related to the values of the attributes. The mathematical model used to select the attributes influencing the classification as well as the attribute values involved in the conditions is based on the entropy provided by the attribute. The J48 algorithm is an advanced version of C4.5 [Quinlan, 1993], one of the most known and used tree induction methods. J48 is an information gain based method with pruning procedures that use rules.

Bayesian networks (BN) are probabilistic graphical models where nodes represent random variables and edges represent conditional dependencies between the variables. Two nodes representing conditionally independent variables are not connected each other. The learning process for a dataset D lies in finding, among all possible graphs, the graph G that better represents the set of dependencies/independencies between data. The problem is NP-hard, so that is not feasible an exact solution and it is necessary to resort to heuristic search methods. They consist of establishing a quality metric, which represents the adaptation of a Bayesian network to a specific dataset, and finding a solution that maximizes this metric by means of an optimization procedure. Some search algorithms are TAN, BAN, K2, etc. TAN (Augmented Naïve-Bayes) and BAN (Bayesian Networks Augmented Naïve-Bayes) are based in the simplest BN, Naïve Bayes, but they involves sophisticated graphical model to deal with the no realistic assumption of attribute independence. K2 is a function based on uniform prior scoring for learning and evaluating Bayesian networks [Cooper and Herskovits, 1992].

Multiclassifiers combine several individual classifiers induced with different basic methods or obtained from different training datasets with the aim of improving the accuracy of the predictions. The methods for building multiclassifiers can be divided in two groups. The first, also named ensemble classifiers, such as Bagging [Breiman, 1996], Boosting [Freund and Schapire, 1996] and Random Forest [Breiman, 2001], induce models that merge classifiers with the same learning algorithm, but introducing modifications in the training data set. The second type of methods, named hybrids, such as Stacking [Wolpert, 1992] and Cascading [Gama and Brazdil, 2000], create new hybrid learning techniques from different base learning algorithms.

Bagging is the acronym for Bootstrap AGGregatING. The method induces a multiclassifier that consists on an ensemble of classifiers built on bootstrap replicates of the training set. Among the different ways of combining the outputs of the classifiers in an ensemble (abstract level, rank level, measurement level…), abstract level is the type used by bagging. Given a set of labels Ω, a set of classifiers D and examples $x \in \mathcal{R}^n$ to be classified, in this approach each classifier Di produces a class label $s_i \in \Omega, i = 1, \dots, L$. Thus, for any example $x \in \mathcal{R}^n$ to be classified, the outputs of a L classifier are given by a vector $s = [s_1, \dots, s_L]^T \in \Omega^L$. The label outputs of the classifiers can be represented as binary vectors $\left[d_{i,1}, \dots, d_{i,c}\right]^T \in \{0,1\}^c, i = 1, \dots, L,$ where $d_{i,j}$ = 1 if D_i labels **x** in ω_j, and 0 otherwise, then, the final choice of the class is carried out by majority vote [Kuncheva, 2004].

Boosting is a multiclassifier of the same kind of Bagging, however, this method assign weights to the outputs of the induced single classifiers from different training sets (strategies). The weight of a strategy s_i represents the probability that s_i is the most accurate of all of them. In an iterative process, the weights are updated by increasing the weight of strategies with the correct s_i prediction and reducing the weight of strategies with incorrect predictions. In this way the multiclassifier is developed incrementally, adding one classifier at a time. The classifier that joins the ensemble at step k is trained on a data set selectively sampled from the training data set Z. The sampling distribution starts from uniform, and progresses in each k step towards increasing the likelihood of worst classified data points at step k – 1 [Kuncheva, 2004]. This algorithm is called AdaBoost which comes from ADAptive BOOSTing. This algorithm presents the advantage of drive the ensemble training error to zero in very few iterations [Kuncheva, 2004].

Random Forest [Breiman, 2001] can be considered a multiclassifier similar to Bagging since it involves the induction of an ensemble of tree classifiers, each of which produces its own output. The induction of each tree is produced from a subset of the original data set chosen independently (with replacement) and with the same distribution for all trees in the forest. For classification problems, the most popular class obtained by simple vote is chosen as the final outcome.

Random Committee algorithm is used to induce an ensemble where each base classifier is built using a different random number seed. The final prediction is generated by averaging probability estimates over the individual base classifiers.

5 Data Mining Study

As commented before, the main problem to be addressed is the treatment of imbalanced data. The behavior of classifiers used in an individual way sometimes fails with imbalanced datasets, achieving very low precision in the classification of minority class examples. In this work an alternative way of dealing with this drawback is proposed aiming at avoiding some of the disadvantages of usual approaches.

The proposal is based on the use of ensemble classifiers that build several hypothesis from different datasets following a resampling strategy. However, this strategy differs from the used in the classical treatment of imbalanced problems since it is not focused on one particular class, so that the outcome is not biased to a specific class of instances. In addition, ensemble methods have potential capacity to minimize overfitting problems. Adaboost has the capacity to avoid overfitting problems and reduce errors at the same time in spite of the progressively increasing complexity of the induced classifiers [Kuncheva, 2004]. This fact is essential to deal with the imbalanced data problem avoiding one of the main weaknesses of the oversamping strategy.

According to Kuncheva, the key of the good behavior of classifiers ensemble is the diversity provided by different training sets. When training sets are generated from Bootstrap sampling, significant improvements are achieved mainly when the base classifier is unstable, that is, small changes in the training set should lead to large changes in the classifier output. This is the scenario that takes place when working with imbalanced data. On the other hand, majority vote properties assure the improvement of the single classifiers results if the outputs were independent and classifiers had the same individual accuracy. Outputs of Bagging cannot be considered completely independent since the training samples are formed from the same set by taking bootstrap replicates, however Bagging improves accuracy of single classifiers due to the bias-variance decomposition of the classification error [Kuncheva, 2004].

Based on the above premises several ensemble classifiers were applied to the available data using different base classifiers. The dataset used in the study comprises information about 4616 patients hospitalized in the ICU of the University Hospital of Salamanca. We focused on predictive factors of infections, thus some attributes as days of stay or death were discarded since they are not known until the end of the stay. The attributes used in the learning process were the following: Gender, Acute Physiology and Chronic Health Evaluation (APACHE II), Emergency surgery, Immunosuppression, Neutropenia, Immunodeficiency, Mechanical Ventilation, Central Venous Catheter (CVC), Urinary Catheter, Parenteral nutrition, Patient origin, 48 hours of antibiotic treatment, Previous surgery, Extrarenal depuration.

6 Results

Several classification algorithms were applied to induce models that allow to predict acquisition of infection by ICU patients. In order to do a comparative study of results, two kind of algorithms were used for inducing both single classifiers and multiclassifiers, specifically, ensemble classifiers. These were tested with several base classifiers. The results of some algorithms providing very poor results (accuracy lesser than 60%) have not been reported.

Ten-fold cross-validation was used in the validation of all classifiers. Given the fact that accuracy is not a suitable evaluation measure for imbalanced data contexts, several metrics introduced in section 3 have been obtained. They are TP rate, FP rate, precision, recall (sensitivity), F-measure and ROC area. F-measure was computed with β parameter set to 1. Table 1 shows the values of these measures. In order to appreciate in a better way the differences between the results of the classification algorithms, precision, recall and F-measure are represented in Figure 1.

Table 1 Results provided by different classifiers

Algorithm	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
J48	93.96%	94.0%	70.5%	92.7%	94.0%	92.7%	60.10%
Random Forest	94.65%	94.60%	53.20%	93.90%	94.60%	94.10%	81.30%
Bayes Net-K2	91.72%	91.70%	59.40%	91.60%	91.70%	91.60%	82.10%
Bayes Net-TAN	93.31%	93.30%	68.20%	91.90%	93.30%	92.30%	84.90%
AdaBoost-J48	94.25%	94.30%	48.70%	93.80%	94.30%	94.00%	84.20%
AdaBoost-Random Forest	**95.26%**	95.30%	50.50%	**94.70%**	**95.30%**	**94.70%**	81.70%
Bagging-J48	94.13%	94.10%	68.10%	93.00%	94.00%	93.00%	80.70%
Bagging-Random Forest	**95.10%**	95.10%	56.00%	**94.50%**	**95.10%**	**94.40%**	**87.90%**
Random Committee-Random Tree	94.63%	94.60%	50.80%	94.00%	94.60%	94.20%	81.90%
Random Committee-Random Forest	94.97%	95.00%	50.50%	94.40%	95.00%	**94.50%**	**86.80%**

The best values of all of the metrics except ROC area were achieved by AdaBoost when Random Forest was used as base classifier. However, Bagging with Random Forest achieved a significantly better value of the ROC area, the best of all classifiers, with a very slight worsening of the other metrics. In order to analyze the behavior of classifiers regarding the classification for the instances of every class, the precision for both classes was also examined. Figure 2 shows the values of accuracy, precision of class YES (the minority class) and precision of class NO. Bagging with Random Forest reached a significant better precision for the minority class than the other classifiers without hardly descending the precision of the majority class, thus, the lesser difference between the precision of both classes was accomplished by this algorithm. Therefore, after the analysis of different quality metrics, we can conclude that the ensemble Bagging, when uses Random Forest as base classifier, is the most suitable for the classification of healthcare-associated infections from the data used in this study. Another conclusion derived of this results is the fact that Random Forest used with any multiclassifier improves the accuracy and the precision of the minority class with respect to other base classifiers. Consequently, the combination of two ensemble classifiers seems to be effective for dealing with the imbalanced data problem.

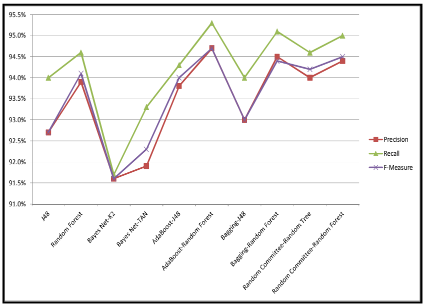


Fig. 1 Values of Precision, recall and F-measure for the tested classifiers

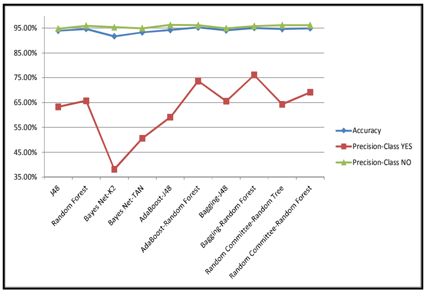


Fig. 2 Values of accuracy, precision of the class YES and precision of the class NO for the studied classifiers

7 Conclusions

In this work, the data from 4616 patients hospitalized in ICU have been processed with different data mining algorithms aiming at predicting device-associated infections in future patients. It was necessary to address the problem of building classification models from imbalanced datasets since only 6.7% of the patients in the dataset presented nosocomial infection while 93.3% did not acquired any infection. Our proposal is focused on the application of ensemble classifiers. The purpose is to avoid problems as overfitting derived from the use of resampling strategies as well as some difficulties in the implementation of other approaches as algorithm modification and cost-sensitive learning.

The results proved that multiclassifiers showed better behavior than single classifiers, especially Bagging and Boosting when Random Forest was used as base classifier. Bagging with Random Forest achieved the best value of area under the ROC curve, the best precision for the minority class and the smallest precision difference between the two classes. Random forest was also the best base classifier for all multiclassifiers. The study carried out has revealed that the combination of ensemble classifiers is an effective way to address the imbalanced data problem and can be an alternative approach to the usual procedures.

References

1. Breiman, L.: Bagging predictor. Machine Learning **24**(2), 123–140 (1996)
2. Breiman, L.: Random Forests. Machine Learning **45**(1), 5–32 (2001)
3. Chawla, N.V.: Data Mining for imbalanced datasets: an overview. In: Maimon, O., Rokach, L., (eds.) Data Mining and Knowledge Discovery Handbook, pp. 853–867. Springer (2005)

4. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning **9**, 309–347 (1992)
5. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings 13th International Conference on Machine Learning, pp. 148–156 (1996)
6. Gama, J., Brazdil, P.: Cascade Generalization. Machine Learning **41**(3), 315–343 (2000)
7. Hulse, J., Khoshgoftaar, T., Napolitano, A.: Experimental perspectives on learning from imbalanced data. In: Proceedings of the 24th International Conference on Machine Learning, pp. 935–942 (2007)
8. Kuncheva, L.I.: Combining pattern classifiers: Methods and Algorithms. John Wiley & Sons (2004)
9. Lopez, V., Fernandez, S., Garcia, A., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences **250**, 113–141 (2013)
10. Martín, F., González, J., Sánchez, F., Moreno, M.N.: Success/failure prediction of non-invasive mechanical ventilation in intensive care units. Using multiclassifiers and feature selection methods, Methods of Information in Medicine (in press). doi:10.3414/ME14-01-0015
11. Moreno, M.N., González, J., Martín, F., Sánchez, F., Sánchez, M.: Machine learning methods for mortality prediction of polytraumatized patients in intensive care units. In: Dealing with Imbalanced and High-Dimensional Data. Lecture Notes in Computer Science, vol. 8669, pp. 309–317. Springer (2014)
12. Nahar, J., Imama, T., Tickle, K.S., Chen, Y.P.: Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. Expert Systems with Applications **40**, 96–104 (2013)
13. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning (1998)
14. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo (1993)
15. Wolpert, D.H.: Stacked Generalization. Neural Networks **5**, 241–259 (1992)

Fast Intelligent Image Reconstruction Algorithm for ECT Systems

Wael A. Deabes and Hesham H. Amin

Abstract Electrical Capacitance Tomography (ECT) has more attention in the last few decades due to its importance in many industrial and medical processes. Research has various directions in this field such how reconstruct accurate images of the object under consideration, hardware implantation of both the recognition system and/or the image viewing devices. In this paper, a novel single-stage intelligent approach is designed for reconstructing images that describe the materials distribution of the multi-phase flow in industrial pipelines. The proposed algorithm utilizes Fuzzy Inference System (FIS) to overcome the nonlinear response of the ECT system. The proposed algorithm is fast since it does not need solving the forward problem to update the sensitivity matrix. The reported results show that the proposed FIS image reconstruction algorithm has high accuracy and promising technique.

Keywords ECT · Image reconstruction · FIS · Multiphase flow

1 Introduction

Recently, the level of interest in the multi-phase flow measurements, which used in the chemical and petroleum industries, has increased rapidly [1]. Multi-phase flow is

W.A. Deabes · H.H. Amin(✉)
Computer Science Department, University College, Umm Al-Qura University,
Makkah, Kingdom of Saudi Arabia
e-mail: {wadeabes,hhabuelhasan}@uqu.edu.sa

W.A. Deabes
Computer and System Department, Faculty of Engineering,
Mansoura University, Mansoura, Egypt

H.H. Amin
Computer Systems Department, Faculty of Engineering, Aswan University, Aswan, Egypt

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_34

defined as the flow of more than one material along a closed or open pipe. Two phases flow of gas, liquid or solid is common. Thus, the study of multi-phase flow measurement techniques are crucial to understand and quantify its parameters [2]–[4].

The electrical tomography (ET) techniques are the most important non-intrusive sensing modalities in the multi-phase flow measurements [5]. The ET systems are remarkable because real-time imaging, safe and suitable for different vessel sizes [6]. Moreover, sensors-based ET systems are prominent and have been broadly utilized to characterize the parameters of the multi-phase flow phases. Generally, ET is based on an array of the electrical sensors placed along a pipe axis or a vessel circumference. Information obtained by non-intrusive techniques is either two-dimensional (2D) cross-sectional concentration profiles or 3D volume images [7]. Electrical Capacitance Tomography (ECT) is one of these ET techniques. It is fast, non-intrusive, low cost, robust, and safe technique for imaging multiphase flow [8]. In an ECT system, sensors are arranged around an object of interest that has an unknown distribution of heterogeneous dielectric materials or material phases with significant electrical permittivity contrast. These sensors measure capacitance in between them [2]. The ECT system requires solving two computational problems. First is the forward problem that determines the mutual capacitance between the mounted sensors after the change of the material distribution by solving the partial differential equations governing the sensing domain. Second is the inverse problem that calculates the material distribution from the known capacitance measurements [2], [9]. Solutions of the inverse problem are also known as image reconstruction algorithms. These Algorithms are responsible for mapping a vector of the M measurements to an image of N unknown pixel values. The relationship between the material distribution and the tomography measurements is nonlinear, which makes the image reconstruction a challenging task. However, there are two types of the image reconstruction algorithms, non-iterative and iterative ones. Iterative image reconstruction algorithms are superior to non-iterative ones since they are able to overcome the nonlinearity of the system and generate images that are more accurate. But the iterative algorithms consume long time since it usually goes through both solving the forward problem and the inverse problem.

Various research have been done to get both high accuracy and real time reconstructed images [10][11]. The algorithms used various methodologies to reconstruct the shape of the flow of the materials in the pipelines. These techniques may be classified into Algebraic Reconstruction Techniques and Optimization Reconstruction Techniques [8]. Unfortunately, research is scant of using intelligent techniques such as Artificial Neural Networks, fuzzy systems and machine learning in general [12]. However, many techniques are available for solving the inverse problem, most of them are characterized by a tradeoff between the reconstructed image quality and the computational complexity of the employed algorithms [13].

Another tradeoff related to increasing the number of sensors which provides more information to build exact images, but this slow down the system and impose poor signal-to-noise ratio [5]. Thus, developing an accurate and fast image reconstruction algorithm to overcome these nonlinearities is very crucial [8]. Therefore, this work develops a novel image reconstruction method based on Fuzzy Inference System (FIS) for ECT systems that yields solutions that are more accurate than other explicit approaches, and without increasing of the computational cost. The proposed method, termed “single-stage fuzzy”, provides improved image construction in both time and resolution, making it an attractive choice for ECT where real-time imaging is required. The accuracy and computational ease of the proposed method makes it an ideal algorithm for ECT systems.

2 Electrical Capacitance Tomography (ECT)

Generally, an ECT system is composed of three basic components: (1) a capacitance sensor, (2) a data acquisition system, and (3) a computer system for reconstructing and displaying images [14]. The ECT sensor consists of an array of electrodes mounted on the periphery of the process vessel to be imaged. There is an earthed screen around the ECT sensor to prevent any external electrical field noise and disturbance. Solving of the forward problem depends on known permittivity distribution and the boundary electrical potential to estimate the electrical response. Later, the inverse problem uses the solution of the forward problem to estimate an approximation for the permittivity distribution. Each electrode in the ECT sensor works in turn as a transmitter and the rest of the electrodes act as receivers [6]. Therefore, for N electrodes there are N(N-1)/2 independent captured capacitive measurements. For example, the number of measurement for 12 electrodes equals 66. The measured capacitances are proportional to permittivity of the material distribution through the potential field inside the area of interest. Therefore, the image of the permittivity distribution can be reconstructed by solving the inverse problem derived by the captured capacitance and the potential field.

Forward model [8] used for simulating the response of the ECT sensor for known materials distributions inside the imaging region and determining the capacitance measurements. The solution of the forward problem is also known as the sensitivity matrix. Fig. 1 shows the finite element model of the ECT sensor consisting of 12 electrodes and 2204 elements. Each element of this matrix represents the response of the ECT sensor after changing the permittivity in the corresponding pixel from the low value to the high one. In order to calculate the sensitivity of the sensor the Finite Element Method (FEM) is applied. Fig. 2 shows the different sensitivity matrix for measurements between electrodes separated by 0, 1, 2, 3, 4, and 5 electrodes. Dark areas represent the highest effective elements, yellow ones give medium effect, green contain elements of low effect, while blue do not affect the response.

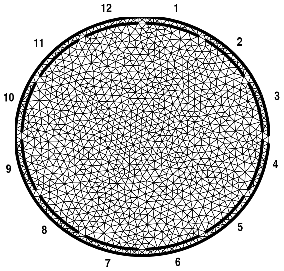


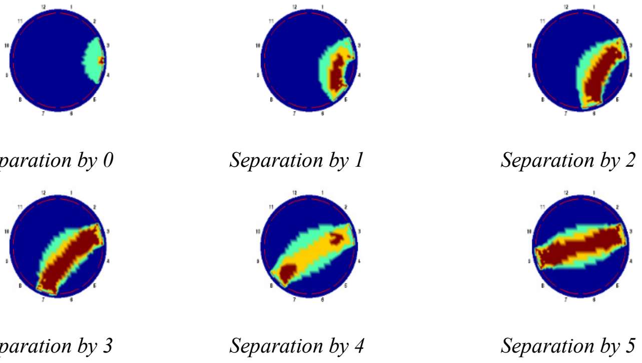
Fig. 1 FEM of the ECT sensor

3 Fuzzy Image Reconstruction Algorithm

The development flowchart of the proposed single-stage fuzzy system for reconstructing the flow images in ECT is shown in Fig. 3. The development of the fuzzy system starts with determining the sensitivity areas of the different pairs of electrodes utilizing a linear sensitivity matrix. Generally, the proposed fuzzy system is comprised of two main steps which building fuzzy membership functions for the inputs and outputs of the system and generating fuzzy rules base.

The proposed fuzzy algorithm for reconstructing images is faster than the iterative nonlinear techniques such as algorithms stated in [8] since it produces accurate images in just one step. Moreover, it does not consume time to update the sensitivity matrix as in the iterative techniques. An FEM is built to simulate the response of the ECT sensor and to solve the forward problem just once. The block diagram of the fuzzy inference system for the ECT is shown in Fig. 4. The development of the fuzzy system starts with determining the sensitivity areas of the different pairs of electrodes utilizing the linear sensitivity matrix. The sensitivity area in front of each sensor is divided into regions, and then the FIS calculates the material distribution in each area based on the capacitance measurement of that corresponding sensor. Based on the resulting sensitivity, for each electrode pairs, each pixel is assigned to a sensitivity class (e.g., “zero”, low, medium, high). For any given electrode pair, many of the pixels are considered to have no effect on the mutual capacitance means they are assigned to class zero. Furthermore, any given pixel will be in various sensitivity regions for multiple electrode pairs, i.e., a given pixel can be in a high sensitivity region for a first electrode pair (e.g., electrodes 1 & 2), a high sensitivity region for a second electrode pair (e.g., electrodes 1 & 3), and a medium sensitivity for a third electrode pair (e.g., electrodes 1 & 4).

To describe the FIS inputs and outputs, the inputs to the fuzzy system, defined as (in1, in2. . . ,in66), are the normalized capacitance measurements between the electrodes mounted on the periphery of the imaging area. The proposed ECT system consisting of 12 electrodes. There are 66 inputs for the FIS since the independent measurements between these electrodes is 66. The first input in1 is the normalized capacitance between electrodes 1 and 2 while the last input in66 is the normalized capacitance between the electrodes 11 and 12. On the other hand, the outputs of the fuzzy system stand for the materials permittivity values in each


Separation by 0 *Separation by 1* *Separation by 2*

Separation by 3 *Separation by 4* *Separation by 5*

Fig. 2 Sensitivity patterns for electrode separation by 0, 1, 2, 3, 4, 5.

pixel individually. The imaging area is divided into n pixels, therefore the total number of the outputs is n labeled as (out1, out2,…., outn). out1 is the measure of the permittivity value in pixel 1, and outn is the value in pixel n. All n fuzzy outputs combined in one vector and is considered the estimated image (Gest) of the material distribution.

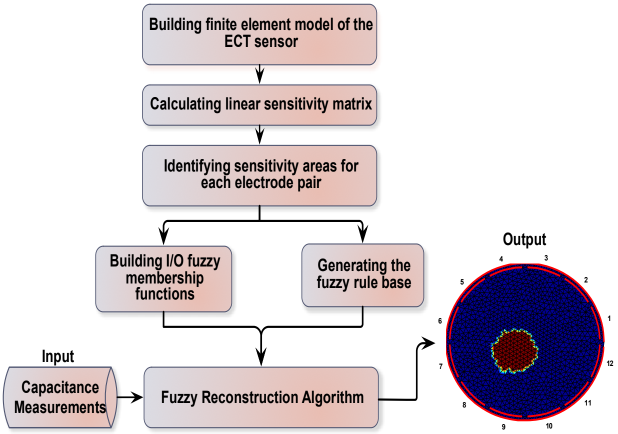

Building finite element model of the ECT sensor
Calculating linear sensitivity matrix
Identifying sensitivity areas for each electrode pair
Building I/O fuzzy membership functions
Generating the fuzzy rule base
Input
Capacitance Measurements
Fuzzy Reconstruction Algorithm
Output
4
3
5
2
6
1
7
12
8
11
9
10

Fig. 3 Flow chart the fuzzy image reconstruction algorithm.

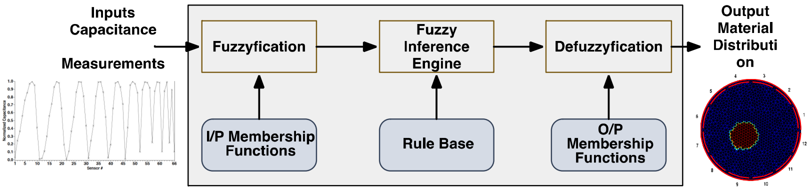


Fig. 4 Fuzzy Inference System (FIS) for Electrical Capacitance Tomography.

The distance between the electrodes is an effective parameter on the shape of the sensitivity area as the sensitivity between each pair of electrodes is proportional to the distance between these electrodes. Therefore, in the sense of capacitance measurements, few elements need to be filled by high permittivity to give high measurement between two near electrodes. The number of elements increases significantly for the electrodes separated by larger distance to obtain the same high measurement. The input membership functions are designed to reveal the characteristic of each input group. Three input membership sets are considered with different membership parameters. Each set is described by four overlapped Gaussian membership functions in range [0 1]. Fig. 5 shows fuzzy membership functions of each set for input i; Zero (Z), Low (L), Medium (M), and High (H), while Fig. 6 shows the output membership functions.

The rule base of the proposed model is built based on the classification of the inputs into groups and on the distance between the probes associated to each measurement. However, every group has different rules to handle the diversity in the sensitivity distribution of all electrodes. The rule base is divided into four classes, where each class handles the inputs associated with one of the input groups. For the 66 inputs, after capturing the indexes of the elements in all the regions for these inputs, the fuzzy rule base for the proposed FIS is generated as follows: There are four rules describing the relationship between every input of the 66 inputs and the elements in each area of the 3 sensitivity areas. The size of the rule base is n rules, since there are 66 inputs and each input has four rules. These four rules for each class are stated in Table 1. In Table 1, Gesth stands for the indexes of elements in the high sensitivity region of input i, Gestm contains the indexes of elements in the medium sensitivity area of input i, and the indexes of elements in the low sensitivity area of input i are represented by Gestl. HN, MN, MP and HP stands for high negative, medium negative, medium positive and high positive, respectively.

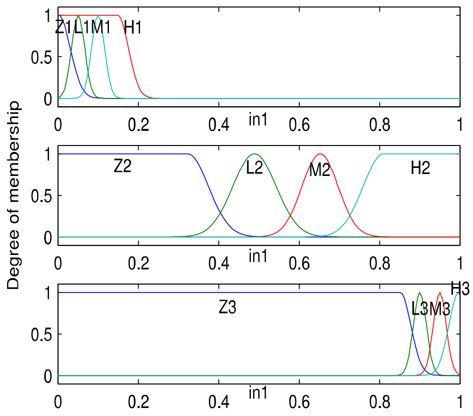


Fig. 5 Inputs membership functions

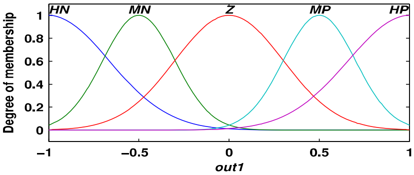


Fig. 6 Outputs Membership Functions

Table 1 Rule Base of the FIS

Class 1 Separation by 4, or 5 electrodes	If in_1 is H_1 then $Gest_l$ is HP, If in_1 is M_1 then $Gest_1$ is MP If in_1 is L_1 then $Gest_1$ is MN, If in_1 is Z_1 then $Gest_1$ is HN
Class 2 Separation by 3, or 2 electrodes	If in_1 is H_2 then $Gest_1$ is HP, If in_1 is M_2 then $Gest_1$ is MP If in_1 is L_2 then $Gest_1$ is Z and $Gest_m$ is MN If in_1 is Z_2 then $Gest_1$ is MN and $Gest_m$ is HN
Class 3 Separation by 1 electrode	If in_1 is H_2 then $Gest_1$ is HP and $Gest_m$ is HP If in_1 is M_2 then $Gest_1$ is MP and $Gest_m$ is MP If in_1 is L_2 then $Gest_1$ is MN and $Gest_m$ is MN and $Gest_h$ is HN If in_1 is Z_2 then $Gest_1$ is HN and $Gest_m$ is HN and $Gest_h$ is HN
Class 4 No Separation	If in_1 is H_3 then $Gest_1$ is HP and $Gest_m$ is HP If in_1 is M_3 then $Gest_1$ is MP and $Gest_m$ is MP and $Gest_h$ is HP If in_1 is L_3 then $Gest_1$ is MN and $Gest_m$ is MN and $Gest_h$ is MN If in_1 is Z_3 then $Gest_1$ is HN and $Gest_m$ is NN and $Gest_h$ is Z

The consequence in each rule is different since the indexes of the elements in the sensitivity region of each input are different. The crisp inputs are fuzzfied based on the membership functions through the calculation of the membership degrees of an input to the antecedents. In the proposed model, the activation degree of a rule is calculated using the Min T-norm operation and its ring strength (conclusion) is computed using the Max-Min implication function. The defuzzification method to get the crisp possibility values from the fuzzy system is computed using the centroid method. This method is considered the simplest of all defuzzification methods. For each output the weighted strengths of the output member function are multiplied by their respective output membership function center points and summed. Pairs of electrodes, viewed as "inputs" in the context of the fuzzy system, are assigned to classes based on their degree of separation as shown in Table 1. Each class of inputs has a different set of fuzzy logic rules. In the case of widely separated electrodes, only dielectric constants for pixels in the highest sensitivity region will be influenced by the mutual capacitance measurement. In the case of electrodes, that are nearby or adjacent electrodes, estimates of dielectric constants for pixels that are in low, medium and high sensitivity regions will be influenced by the mutual capacitance measurement.

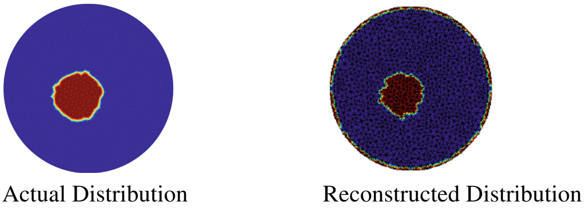


Fig. 7 Results after applying FIS image reconstruction algorithm

4 Results and Discussion

Numerical simulations have been performed to test the feasibility of the proposed algorithm. The proposed algorithm is faster than traditional iterative mathematical algorithms since it is single step algorithm.

A friendly user interface Matlab software package for solving the forward and inverse problem is developed. It calls Gmesh Finite Element software to build the ECT senor and discretize the imaging area into elements. Calculating the sensitivity matrix, identifying the sensitivity regions for each sensor (outputs of the fuzzy system), applying the fuzzy algorithm and plot final images are carried out. An ECT system consisting of 12 electrodes, as shown in Fig. 1, is used in this work. Different material distributions within the sensing region are applied to verify the proposed reconstruction technique. Fig. 7 shows the actual distributions of the materials in the first column, and the estimated images are shown in the second column. Reconstructed images from the fuzzy algorithm are accurate and have sharp edges compared with the actual distribution.

5 Conclusion

In this work, an FIS algorithm for reconstructing the images in the ECT is introduced. The proposed system overcomes the nonlinearity associated with the ECT systems. The rule base of the fuzzy system is build based on the linear sensitivity matrix between each two electrodes. The sensitivity area is divided into regions identify the effect of the elements on the corresponding electrode pair measurements. The inputs and outputs of the FIS are the measured capacitance values and material permittivity value in each pixel respectively. The results demonstrate the feasibility of the proposed fuzzy system to reconstruct the material distribution inside the imaging area.

Acknowledgment This work was supported by the NSTIP strategic technologies program for project #13-ELE469-10 in KSA.

References

1. Wang, A., Marashdeh, Q., Fan, L.-S.: ECVT imaging and model analysis of the liquid distribution inside a horizontally installed passive cyclonic gas–liquid separator. Chem. Eng. Sci. **141**, 231–239 (2016)
2. Rasteiro, M.G., Silva, R.C.C., Garcia, F.A.P., Faia, P.M.: Electrical Tomography: a review of Configurations and Applications to Particulate Processes. KONA Powder Part. J. **29**(29), 67–80 (2011)
3. Wang, A., Marashdeh, Q., Motil, B.J., Fan, L.-S.: Electrical capacitance volume tomography for imaging of pulsating flows in a trickle bed. Chem. Eng. Sci. **119** (2014)
4. Wang, Z., Chen, Q., Wang, X., Li, Z., Han, Z.: Dynamic Visualization Approach of the Multiphase Flow Using Electrical Capacitance Tomography. Chinese J. Chem. Eng. **20**(2), 380–388 (2012)
5. Yang, Y., Jia, J., Mccann, H.: A Faster Measurement Strategy of Electrical Capacitance Tomography Using Less Sensing Data, (2) (2015)
6. Isaksen, Ø.: A review of reconstruction techniques for capacitance tomography. Meas. Sci. Technol. **7**(3), 325–337 (1996)
7. Weber, J.M., Layfield, K.J., Van Essendelft, D.T., Mei, J.S.: Fluid bed characterization using Electrical Capacitance Volume Tomography (ECVT), compared to CPFD Software's Barracuda. Powder Technol. **250**, 138–146 (2013)
8. Yang, W.Q., Peng, L.: Image reconstruction algorithms for electrical capacitance tomography. Meas. Sci. Technol. **14**(1), R1–R13 (2003)
9. Rahman, N.A.A., Rahim, R.A., Bawi, A.M., Leow, P.L., Pusppanathan, J., Mohamad, E.J., Chan, K.S., Din, S.M., Ayob, N.M.N., Yunus, R.R.M.: A Review on Electrical Capacitance Tomography Sensor Development. J. Teknol. **73**(3), 35–41 (2015)
10. Liu, X., Wang, X., Hu, H., Li, L., Yang, X.: An extreme learning machine combined with Landweber iteration algorithm for the inverse problem of electrical capacitance tomography. Flow Meas. Instrum. **45**, 348–356 (2015)
11. Taylor, S.H., Garimella, S.V.: An explicit conditioning method for image reconstruction in electrical capacitance tomography. Flow Meas. Instrum. **46**, 155–162 (2015)
12. Deabes, W.A., Abdelrahman, M.A.: A nonlinear fuzzy assisted image reconstruction algorithm for electrical capacitance tomography. ISA Trans. **49**(1), 10–18 (2010)
13. Ye, J., Wang, H., Yang, W.: Image Reconstruction for Electrical Capacitance Tomography Based on Sparse Representation. IEEE Trans. Instrum. Meas. **64**(1) (2015)
14. Abdelrahman, M.A., Gupta, A., Deabes, W.A.: A Feature-Based Solution to Forward Problem in Electrical Capacitance Tomography of Conductive Materials. IEEE Trans. Instrum. Meas. **60**(2), 430–441 (2011)

Detecting Emotions with Smart Resource Artifacts in MAS

Jaime Andres Rincon, Jose-Luis Poza-Lujan, Juan-Luis Posadas-Yagüe, Vicente Julian and Carlos Carrascosa

Abstract This article proposes an application of a social emotional model, which allows to extract, analyze, represent and manage the social emotion of a group of entities. Specifically, the application is based on how music can influence in a positive or negative way over emotional states. The proposed approach employs the JaCalIVE framework, which facilitates the development of this kind of environments. The framework includes a design method and a physical simulator. In this way, the social emotional model allows the creation of simulations over JaCalIVE, in which the emotional states are used in the decision-making of the agents.

Keywords Multi-Agent Systems · Artifacts · Emotion recognition

1 Introduction

Human beings perceive and manage a wide range of stimuli in different environments. These stimuli interfere in our commodity levels modifying our emotional states. Before each one of these stimuli, humans generate responses varying our face gestures, body or bio-electrical ones. These variations in our emotional states could be used as information useful for machines. To do this, it is needed that machines will have the capability of interpreting in a correct way such variations. This is the reason for the design of emotional models that interpret and represent the different emotions. In this case, emotional models such as *Ortony, Clore & Collins* model [1] and the *PAD (Pleasure-Arousal-Dominance)* model [2] are the most used ones to detect or

J.A. Rincon · V. Julian(✉) · C. Carrascosa
Departamento de Sistemas Informáticos y Computación (DSIC),
Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain
e-mail: vinglada@dsic.upv.es

J.-L. Poza-Lujan · J.-L. Posadas-Yagüe
Departamento de Ingeniería de Sistemas y Automática (DISCA),
Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_35

simulate emotional states. Nevertheless, these models do not take into account the possibility of having multiple emotions inside an heterogeneous group of entities, where each one of such entities have the capability of detecting and/or emulating a specific emotion. According to this, in [3] a social emotional model which includes multiple emotions between humans and software agents was defined. This model was based on the *PAD* model to represent the social emotion of a group.

The need for detecting the emotion of an heterogeneous group of entities can be reflected in the different applications that could be obtained. With the appearance of the different smart devices, ubiquitous computation and ambient intelligent, emotional states turn into valuable information, allowing to develop applications that help to improve the human being life quality.

In this work we propose to employ the social emotion of a group of agents (humans or not) in an AmI application. Concretely, we propose in this paper a system for controlling automatically the music which is playing in a bar. The main goal of the DJ is to play music making that all individuals within the bar are mostly as happy as possible. Each of the individuals is represented by an agent, which has an emotional response according to his musical taste. That is, depending on the musical genre of the song, agents will respond varying their emotional state. Moreover, by varying emotions of the agents will modify the social emotion of the group.

2 Problem Description

The proposed application example is an extension of the developed application presented in [4]. This application is based on how music can influence in a positive or negative way over emotional states [5]. The application example is developed in a bar, where there is a DJ agent in charge of playing music and a specific number of individuals listening to the music. The main goal of the DJ is to play music making that all individuals within the bar are mostly as happy as possible. Each of the individuals will be represented by an agent, which has an emotional response according to its musical taste. That is, depending on the musical genre of the song, agents will respond varying their emotional state. Moreover, varying emotions of each agent will modify the social emotion of the group. The DJ agent plays a song. Once the song has ended, the DJ evaluates the social emotion of the group of listeners that are within the bar. In this way, the DJ agent can evaluate the effect that the song has had the song over the audience. This will help the DJ to decide whether to continue with the same musical genre or not in order to improve the emotional state of the group.

The solution presented has some lacks regarding the right identification of the people's emotions. It has been detected that problems arise due to pub light conditioning and high number of pub clients. Camera identification increases errors when light conditioning is very dark and the number of people increases. So, a novel more customized approach has been integrated as a new kind of artifact, that in a band form is used to perceive the emotion dynamics of each person in the pub. This new information will be correlated with the one obtained by the cameras improving the emotion detection of each individual.

Next section details how this new artifact has been integrated in the previous system, its internal components and how it interacts with the rest of the system.

3 System Proposal

This section explains the different components that constitute the system which describes a way to connect the human beings to an *Intelligent Virtual Environment* (*IVE*). Concretely, the proposed system is structured as shown in Figure 1. Following this figure the different elements of the system are: (i) the *Smart Resource Artifact (SRA)*: this element allows to capture the information of the real world. It has the capability of perceiving and/or acting in the real world; (ii) the *Environment Manager*: this agent has the task to control all the *IVE*, to register each SRA and to know where is each one of the elements composing the *IVE*; (iii) the *Human-immersed agent*: this agent is the virtual representation of the human living in the real word. This agent has the competence of communicating with each human through the *SRA*; and (iv) the *SEtA*: this agent is responsible of calculating the social emotion. This social emotion is obtained from each emotion that was sent by each human-immersed agent. Next subsection details each one of these elements.

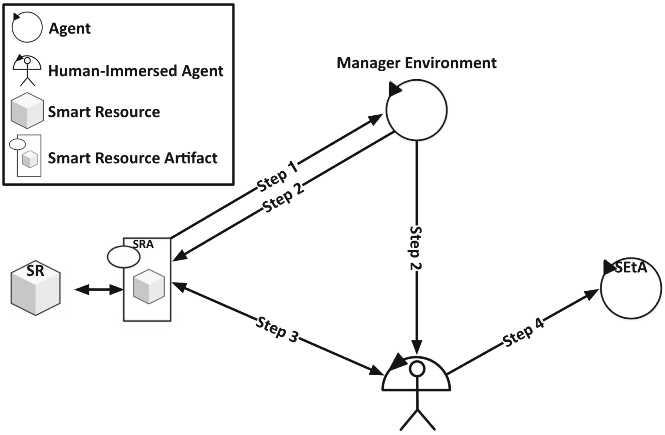


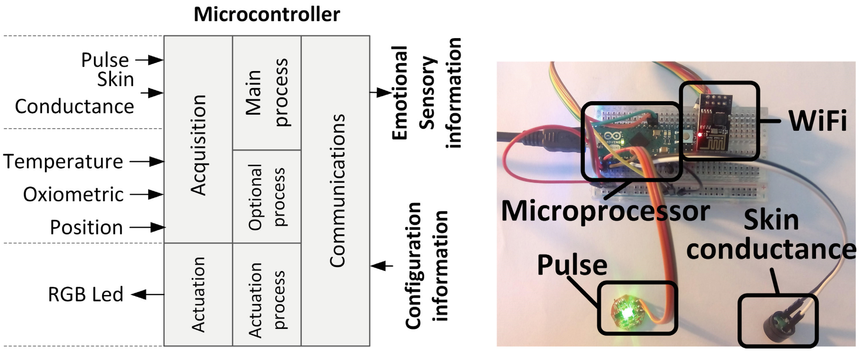
Fig. 1 System's Model Design.

3.1 Smart Resources

Recognizing emotions is easy for a human looking at the face or listening the nuance in the voice of a person. Many different methods to recognize emotions automatically have been studied. One of the more extended approach is based on images and video [6]. Significant improvements have been made using Electroencephalography

(EEG) [7], due to the brain being measured directly. However, both these methods have their associated problems: video image requires high processing load, and a good environment light conditions and *EEG* is an invasive method. Consequently, it is convenient to use other human parameters, like body posture [8] or body biosignals [9].

To measure biosignals it is necessary to use sensors. Sensors transform physical magnitudes in electrical signals. Usually, a simple process can transform the electrical signal, measured by the sensor, in a measure based in a physical unit. Nevertheless, to deduce emotions from units is a hard task [10]. Therefore the sensor hardware and software measurement has an important role to play in the emotion recognition process. Sensors are placed on devices that can measure from different sensor types and different sensors sources, in this case the devices are called smart devices or smart sensors [11]. If there are a lot of smart devices, communications between them, and between smart devices and network servers (for example to store historical measures) is an important challenge. In this latter case, the smart device must increase his connectivity functionality offering services to the rest of the system. Devices are perceived by others as a resource that offers services, so that smart devices change into smart resources, defined in [12]. Figure 2. a) overalls a Smart Resource specifically to bio-signals. Among all the sensors of the Smart Resource, some sensors can be mandatory and others are optionals. Required sensors are those without which clients connected to the Smart Resource, can not detect the emotions, for example, pulse range. Optional sensors, allow clients to increase the reliability of the emotion detected, for example a temperature sensor. Actuators, in smaller quantity that the sensors, are in general optional. Both, sensors and actuators, have a process step to pre-process sensor signals and adapt them to the agents requirements or to convert actuator orders to signals comprehensible for actuators. Finally, a Smart Resource has a communication step that join the real smart resource with its corresponding *SRA*.



(a) (b)

Fig. 2 Biosignal Smart Resource: a) designed and b)implemented.

There are many reasons to use a Smart Resource in the context of biosignals treatment. Some of them can be highlighted. The main advantage is the possibility to select different biosignals and, consequently, adapt the measurements to the user experience. Additionally, a Smart Resource can select the most appropriate biosignal in each moment. For example in a cold environment, temperature sensor can not offer an accuracy value. In this case the Smart Resource does not measure, but estimates the temperature values. Other benefits are the ability to adapt communications characteristics (for example the messages frequency) to the Quality of Service (QoS) parameters or change the processes characteristics (for example processor load or battery consumption) according to Quality of Context (*QoC*) parameters. To test system with real biosignals, a prototype of the Smart Resource has been developed (Figure (2. b)). The Smart Resource consists of an open-hardware pulse rate sensor[1] and an skin conductance (two wires sensor) connected with an Arduino Micro[2] that uses an ESP8266[3] as IEEE 802.11 interface. The main reason to use these components is the integration capacity for a future small bracelet.

Communication between SRAs and smart resources is based on access to services. The smart resource offers an interface to the services as used by its SRA wrapper by following the proposed protocol shown in Fig. 3.

When a smart resource is switched on, it initializes sensors, actuators and communications and then connects to the *environment manager* agent in order to register itself in the virtual environment. For that, *SRAs* only need to know the IP address of the *environment manager* that is public in the system. Once connected, the smart resource sends to the *environment manager* an XML message with its identification, location, and the resources or services it offers. The *environment manager* then creates a SRA associated to the smart resource. With the information of the XML message, the *environment manager* knows how to access the resources/services offered by the registered smart resource. The XML message indicates the base URI (uniform resource identifier) of the smart resource and a different identifier for each resource/service. A resource/service can be applied for by adding to the base *URI* its identifier (for example, the temperature could be obtained with http://192.168.1.14/resources/temperature). Every resource/service has its own URI in the same way that representational state transfer (*REST*) services work (the sw architectural style of the WWW) [13].

SRAs work with resources/services through standard *HTTP* operations such as *GET* (to obtain the value of a resource: for example, the current temperature) or *PUT* (to send data to update a resource: for example, to switch on/off an specific sensor). When an agent needs to access a resource, the agent makes a request to the *environment manager* that knows the associated SRA. The *environment manager* translates the request to the corresponding SRA and then this *SRA* uses the resource URI by sending a *GET* or *PUT* operation to the smart resource. In the case of a *GET* operation, the smart resource responds to the SRA with the content information of the

[1] http://pulsesensor.com/

[2] http://www.arduino.cc/

[3] http://www.esp8266.com/

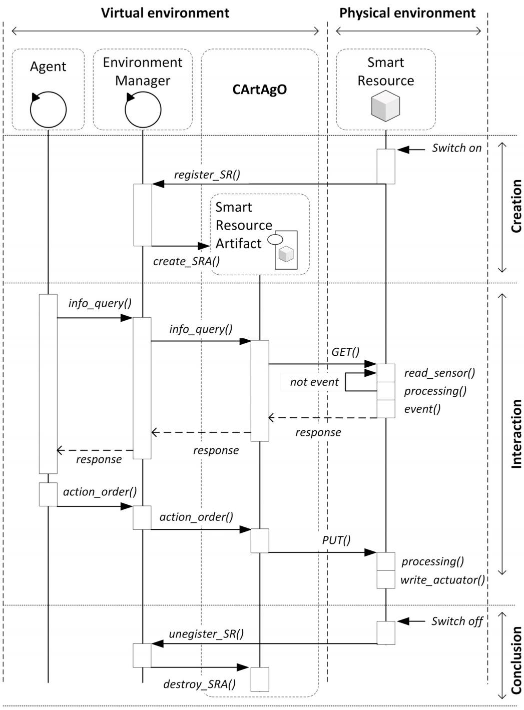


Fig. 3 Protocol to work with a SRA based on a smart resource.

resource by means of an *XML* message and, in the case of a PUT operation, the SRA sends to the smart resource an *XML* message with the information to be updated. In a *GET* operation, the SRA can apply for the smart resource response when an specific event happens (for instance, to receive the value of the temperature only when changes in one grade). For that, the SRA has to add to the *URI*, associated to the *GET* operation, the necessary parameters. These parameters have to include the required event that the smart resource will have to detect before sending the response to the SRA (in the case of the example http://192.168.1.14/resources/temperature?changes=1grade).

3.2 Multi-Agent System

This section describes the three types of agents that forms the proposed MAS. These agents are: the *Environment Manager*, the *Human-immersed agent*, and the *SEtA*. The Human-Immersed agent is mainly in charge of: (i) capture some emotional information from the environment and specifically from a specific individual, this is done by interacting with the real world through the employed *SRA*s. These interactions allow the agent to capture the different bio-signals, that will be used to detect the emotion of a human being. To do this, the agent obtains the signal preprocessing sent by the *SRA*, after this, the agent executes a signal filtering process and obtains a feature vector; and (ii) predict the emotional state of the individual form the processed biosignals. To do this, we take into account that the variation of human emotional states is a modification of our biochemistry [14]. This modification makes that our body and our bio-signal change[15]. In order to analyze these changes and predict emotional states, the Human-Immersed agent employs a classifier algorithm that has been previously trained. The classifier has been trained using a specialized database defined in [16]. This database stores a set of bio-signals obtained after different experiments using a series of music videos and 32 volunteers. The stored data gives us a set of training of 40 x 40 x 8064 entries (corresponding to *video x trial data x channel*). The database has 4 output labels that correspond to the emotional value, composed by an array of 40 x 4 with the following labels: valence, arousal, dominance and liking. This learning capability allows the agent to predict an emotional state according to current bio-signals.

Once the emotion has been obtained, it is sent to the agent which is in charge of calculating the social emotion of the agent group. This agent is called *Social Emotion Agent* or *SEtA*. The main goal of this agent is to receive the calculated emotions from all the human-immersed agents and, using this information, generate a social emotional state for the agent's group (details of how this social emotion is calculated can be seen in [3]). Once this social emotion is obtained, the *SEtA* can calculate the distance between the social emotion and a possible target emotion. This allows to know how far is the agent's group of the target emotion. This can be used by the system to try to reduce that distance modifying the environment. This modification of the environment is made by the *Environment Manager*. This agent is in charge of sending the appropriated actions to the corresponding *SRA*. Moreover, this agent is also in charge of registering all the agents and artifacts, and controlling where are located each one of them.

4 Conclusions and Future Work

This paper presents an agent-based application where humans emotional states are used in the decision-making of the intelligent entities. The application has been developed over the $JaCalIVE$ framework allowing an easy integration of the human in the multi-agent system and a visualization of the system in a virtual environment. The proposed system is able to extract (in a non-invasive way) and to analyze the social emotion of a group of persons and it facilitates the group decision making in order to change the emotional state of the group or only of a subgroup of the agents. The system incorporates automatic emotion recognition using biosignals through the use of smart resources. These smart resources are easily included in the proposed framework using an efficient interaction protocol. Moreover, as future work, we want to apply this system to other application domains, concretely in an industrial one, where it can monitor and simulate the individuals inside a factory.

Acknowledgements This work is partially supported by the MINECO/FEDER TIN2015-65515-C4-1-R and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon.

References

1. Colby, B.N., Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions, vol. 18. Cambridge University Press, May 1989
2. Mehrabian, A.: Analysis of affiliation-related traits in terms of the PAD Temperament Model. The Journal of Psychology **131**(1), 101–117 (1997)
3. Rincon, J.A., Julian, V., Carrascosa. C.: Social emotional model. In: 13th International Conference on Practical Applications of Agents and Multi-Agent Systems. LNAI, vol. 9086, pp. 199–210 (2015)
4. Rincon, J.A., Julian, V., Carrascosa, C.: An Emotional-based hybrid application for human-agent societies. In: 10th Int. Conf. on Soft Computing Models in Industrial and Environmental Applications, vol. 368, pp. 203–214 (2015)
5. Whitman, B., Smaragdis, P.: Combining musical and cultural features for intelligent style detection. In: Ismir, Paris, France, pp. 5–10 (2002)
6. Sun, Y., Sebe, N., Lew, M.S., Gevers, T.: Authentic emotion detection in real-time video. In: Human Computer Interaction, European Conference on Computer Vision, pp. 94–104. Springer (2004)
7. Liu, Y., Sourina, O., Nguyen, M.K.: Real-time EEG-based emotion recognition and its applications. In: Transactions on Computational Science XII, vol. 6670, pp. 256–277. Springer (2011)
8. Coulson, M.: Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. Journal of Nonverbal Behavior **28**(2), 117–139 (2004)
9. Canento, F., Fred, A., Silva, H., Gamboa, H., Lourenço, A.: Multimodal biosignal sensor data handling for emotion recognition. In: 2011 IEEE Sensors, pp. 647–650. IEEE (2011)
10. Haag, A., Goronzy, S., Schaich, P., Williams, J.: Emotion recognition using bio-sensors: first steps towards an automatic system. In: ADS, pp. 36–48. Springer (2004)
11. Meijer, G.C.M., Maria Meijer, C.: Smart sensor systems. Wiley Online Library (2008)
12. Munera, E., Poza-Lujan, J.-L., Posadas-Yagüe, J.-L., Simó-Ten, J.-E., Noguera, J.F.B.: Dynamic Reconfiguration of a RGBD Sensor Based on QoS and QoC Requirements in Distributed Systems. Sensors **15**(8), 18080–18101 (2015)
13. Richardson, L., Amundsen, M., Ruby, S.: RESTful Web APIs. O'Reilly Media, Inc. (2013)
14. Carter, C.S., Porges, S.W.: Carter and Stephen W Porges. The biochemistry of love: an oxytocin hypothesis. EMBO reports **14**(1), 12–16 (2012)
15. Zhao, Q.: A Molecular and Biophysical Model of the Biosignal. Quantum Matter **2**(1), 9–16 (2013)
16. Koelstra, S., Mühl, C., Soleymani, M., Lee, J.S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., Patras, I.: DEAP: A database for emotion analysis; Using physiological signals. IEEE Transactions on Affective Computing **3**(1), 18–31 (2012)

Smart Resource Integration on ROS-Based Systems: Highly Decoupled Resources for a Modular and Scalable Robot Development

Eduardo Munera, Jose-Luis Poza-Lujan, Juan-Luis Posadas-Yagüe, Jose-Enrique Simó-Ten and Francisco Blanes

Abstract Nowadays robots are evolving from using a central computer unit with high computation capability to a distributed system configuration. Most cases present a robot formed with a central unit, which manages and distributes several specific tasks to some embedded systems on-board. Now these embedded systems are also evolving to more complex systems that are developed not only for executing simple tasks but offering some advanced algorithms just as complex data processing, adaptive execution, or fault-tolerance and alarm rising mechanisms. Smart Resources topology has been raised to manage abstract resources which execution relies on a physical embedded hardware. These resources are defined as a list of distributed services that can configure its execution within a context and quality requirements. Therefore this work introduces how a robot can take the advantage by making use of these Smart Resources. In order to provide a more general implementation Smart Resources are integrated into the ROS (Robot Operating System). As a result robots can make use of all the functions and mechanisms provided by the ROS and the distribution, reliability and adaptability of the Smart Resources. Finally, these advantages are reviewed by implementing a Smart Resource into a robot platform that is running ROS. In addition it is also addressed the flexibility and scalability of implementation by combining real and simulated devices into the same platform. As a result it will be summarized all the advantages of this integration and the potential application and upgrades that can be introduced into the system.

Keywords Intelligent sensors · Smart resources · Distributed systems · ROS

E. Munera(✉) · J.-L. Poza-Lujan · J.-L. Posadas-Yagüe · J.-E. Simó-Ten · F. Blanes
University Institute of Control Systems and Industrial Computing (ai2),
Universitat Politècnica de València (UPV), Camino de vera s/n., 46022 Valencia, Spain
e-mail: {emunera,jopolu,jposadas,jsimo,pblanes}@ai2.upv.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_36

1 Introduction

Currently robots are increasing the complexity of their operation. For this reason there have been raised several proposals for easing their development: from system platforms, as robot specific operating systems, to programming languages or dedicated sensors and actuators. The Robot Operating System (ROS) [1] is one of the most wide used software platform for robotics. In the market can be found many robots and devices that are characterized as ROS-ready. There are a wide range of options from whole robots to only some parts such as sensors or actuators (e.g. robotic arms). ROS-ready devices allow robot to be accessed by using standard ROS communication protocol or including a certain set of libraries provided by developers. Despite of this, there is a lack of standardization of the way these devices are managed. Furthermore, some extra capabilities like adaptability, fault-tolerance and quality restrictions are not mandatory, so they can be not implemented.

Smart Resources, described in [2], are developed in order to provide distributed services which have to be executed within a certain requirements by according to the execution context. So, this implementation allows the user to know the performance of the service in any time. Context variation can be configured to trigger adaptation mechanisms to fit the new environment. Alarms will also be raised every time some requirements are not fulfilled, so user can decide if the Smart Resource is performing as expected and define the level of adaptation.

Therefore, considering the convenience to make smart resource as a ROS-ready device, the main objectives to be addressed along this paper are summarized as follows:

- Review and test the advantages of implementing Smart Resources into a robotic system.
- Describe and detail the integration and usage of these Smart Resources to be characterized as a ROS ready device.
- Describe the standard interaction procedure for a ROS ready Smart Resource.
- Test and analyze this implementation on both real and simulate platform.

On the basis of the above, the article describes a method to integrate Smart Resources as smart sensors systems [3]. The article is organized in this order: First section presents a brief review of how sensors systems are included into ROS systems. Next section introduces smart resources used in robotic systems and their convenience for distributed systems. Thereupon, next section applies ROS integration described previously in a specific case study that describes the integration of a distributed smart resource based on a RGBD Camera in a mobile robot. Finally, some conclusions about the integration experiments are presented.

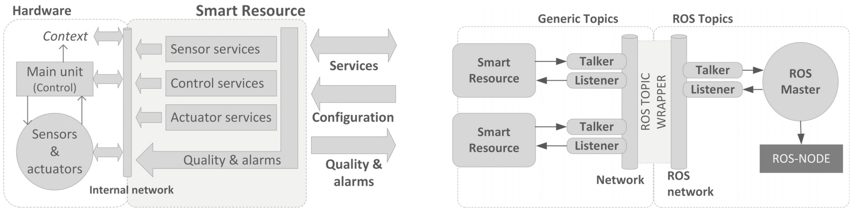
2 Related Work

To distribute sensors and actuator messages between the components of a robot, it is necessary a communication system and some processes to manage all the tasks involved on the robot operation. In the same way that a computer needs an operative system, a robot also need a operative system [4]. From all robot operating systems, ROS [1] has many advantages as a structured architecture based on nodes or a publish/subscribe method to connect nodes between them. ROS is an open source platform, consequently it is easy to modify. Last years, ROS has experimented an important growth [5]. To distribute sensors, actuators and control signals among all clients of the robot or the device (for example, navigation nodes, behavioural nodes, and others), it is necessary a set of communications functions, usually called robot middleware [6]. ROS uses a publish/subscribe based on topics communication system [7]. If the complexity of the nodes increase, the robot nodes, sensors and actuators become a service provider [8]. When the robot or device services are compatible with ROS, the device is defined as ROS-ready robot (or ROS-ready device). Have ROS-ready devices provides a set of advantages to the robot designed and robot user. These advantages include the possibility to make compatible different ROS nodes, the reuse of different developments made by different research groups and companies, and the ability to simulate different devices, robots and scenarios using the multi-robot simulator, Gazebo [9].

There are a great amount of develops ROS ready systems. In [10] is presented a robot that uses ROS to integrate all robot sensors. PhaROS [11] is an architecture that adapts a robot programming language to be used in ROS to program dynamically a robot. ASTRO [12] provides a service architecture oriented to big and complex robot scenarios. ROSbridge [13] provides a bridge to view ROS based systems from non-ROS users, usually web services and Internet. These are just a few examples that ROS can be used in a wide range of robotics and automation system fields. In all the cases described above, ROS is accessed by means a method to translate ROS messages to devices functionalities. Paper proposes to offer devices as resources that interchange ROS messages by means similar topics, these kind of devices are known as ROS Ready devices and it is necessary that devices offer their functionality by means of services.

3 Smart Resource Integration

Smart Resources have been introduced as a suitable option for a wide range of applications. Therefore the main advantages of implementing Smart Resources into any robot architecture is addressed along this section.Furthermore, integration into the ROS is also detailed as depicted in Fig. 1. With this integration, Smart Resources are available in every ROS-based platform in order to ease the configuration and access to the distributed services provided by the Smart Resources.



(a) (b)

Fig. 1 Scheme of: a) Smart Resource for robot. b) Integration with ROS topics.

3.1 Smart Resource for Robotics

In order to introduce Smart Resources for robotic is required to detail how the Smart Resources capabilities can suit the needs of a robot. First of all, the type of service that can be required by any kind of robot is similar to distributed control systems. Therefore, smart resources can provide sensor, actuator and control tasks that will be offered as distributed services. These services can be provided by different nodes, executed in different hardware platforms and supplied by different devices or even simulated devices.

Distributed services provided by Smart Resources are accessed through a network interface. This interface relies on a communication middleware whose main features are: network peer detection, active peer lists management, and a publish-subscriber communication [14]. As any publish-subscriber implementation, communication is driven by topics which can be checked or updated by the network peers. Therefore three main different kinds of topics according to the kind of information can be distinguished: configuration topics for parameterizing the desired tasks execution, the required service, and the quality of the service.

As stated before, Smart Resources offer the capability of adapting to the system which is configured to work within some quality bounds. Every time a Service is requested, it must be configured in order to fulfil the need of the client. Most of these requirements are set in terms of temporal and spatial requirements, information reliability and operation performance.

A certain configuration with certain quality restrictions are defined as a System Profile. System Profiles are pre-programmed in the Smart Resource. A quality requirement is translated into the equivalent (more similar) System Profile. Because of some of these qualities cannot be provided, Smart Resource must try to adapt its execution to suit the requested configuration. Alarms notifies the client every time a quality is not satisfied, being managed as an adaptation event. If minimum requirements can not be satisfied, Smart Resource will execute the most similar performance according to the existing restrictions by offering a best-effort profile. In that case system alarms make the client aware of a non-adaptable event that should be managed as an undesirable or unexpected situation. These non-adaptable events must not be common and are interpreted as a bad design of the Smart Resource or a non-realistic quality requirements set by the client.

Although the service quality adaptation mechanism offers an optimum management of Smart Resource capabilities, the context configuration turns out to be a critical step for increasing the system efficiency. Robots can be designed to perform in a wide range of environments and contexts. More specifically, the mobile robots design usually faces different contexts and dynamic environments and situations. According to this the performance of the required services should be adapted to the context in which the robot is developing its tasks. For this reason Smart Resources can be configured to manage different kind of information according to the needs of the robot, and also modify the quality requirements by changing the System profile.

3.2 ROS Integration

The integration between Smart Resources and ROS systems aims to adapt the service management and supply to be easily accessed through a ROS API turning out as a ROS-Ready device. This is achieved in the communication layer. ROS communications are implemented as a Publish/Subscriber topology, the same used by the communication middleware on the Smart Resources. Therefore an integration layer adapts the service topic information to be managed as ROS topics.

Service information is offered into a data structure understandable for ROS nodes. All the alarm and events mechanisms are also be adapted to manage this data structures. As a result the information off the services can be addressed as any other information provided by a regular ROS node.

4 Experiments: Distributed Smart Resource for Turtlebot Robot

In order to check the integration of the Smart Resources, it is presented a use case in which a ROS-based robot will switch from a centralized design to a decoupled paradigm by implementing the Smart Resource devices. The chosen platform in this use case is the Turtlebot II which can be considered as one of the most spreader ROS-ready robotic platforms. The typical hardware and software configuration is depicted in Fig. 2, including the list of provided services, configuration values, and quality measures of both Smart Resources.

The Turtlebot robot is originally endowed with an on-board RGBD camera (Asus Xtion or Microsoft Kinect) to sense the environment and a Kobuki mobile base for being able to perform a displacement around it. Both devices must be connected to on-board processing unit in order to allow the Turtlebot to manage them. A sensor or actuator management task involves a certain computational cost.

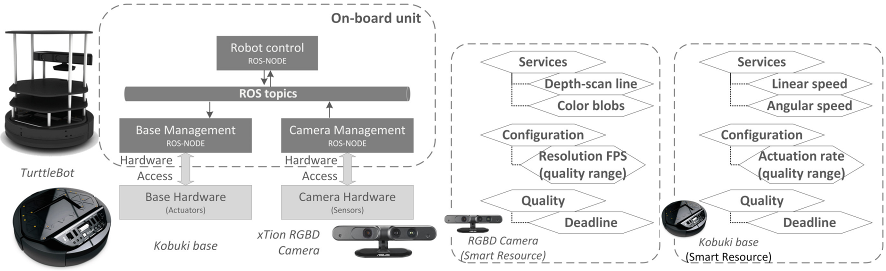


Fig. 2 Turtlebot basic hardware and software set-up.

Although some of this tasks reflect small costs, some others can be set as one of the most resource consuming tasks. One clear example is the data acquisition, management and interpretation of the RGB-D sensor data. As far as more sensors or actuators are added more computation power will be assigned to deal with them. Consequently the computational power available for other control tasks is reduced.

In order to save the on-board processing unit to be used only for robot behaviour and control tasks the sensor and actuator management are distributed. For this reason in these uses case the RGBD camera and the mobile base are decoupled by being implemented as a ROS-ready Smart Resource just as described along this document. As a result the actuation and sensor services, in addition to the usual Smart Resource interface, are accessible through ROS topics. In this use case it is tested the versatility of this implementation by replacing the physical camera and mobile base for simulated ones.

Therefore it is described how the robot behaviour node implemented in the on-board processing unit can perform in the same way since the Smart Resources can provide the same type of service. The set-up for both, the physical and the simulated decoupled design can be reviewed on Fig. 3.

The physical decoupled integration, as showed on Fig. 3.a), makes use of an Asus Xtion RGBD and a Kobouki base connected to their respective BeagleBone Black board as a processing unit characterized as a ROS-ready Smart Resource. The simulated version summarized on Fig. 3.b) make use of a virtual model of the Kobuki and the Asus Xtion camera disposed on the Gazebo [9] simulator. The processing units can be also simulated as different virtual nodes that provide the Smart Resource interface.

In Fig. 4 is displayed the graphical result of the colour service provided by the Xtion Smart Resource in the physical and the simulated implementation. As can be checked, the on-board processing unit node is performing in the same way in both versions as far as the provided services are performing in the way that is expected by offering the required read/write ROS topics.

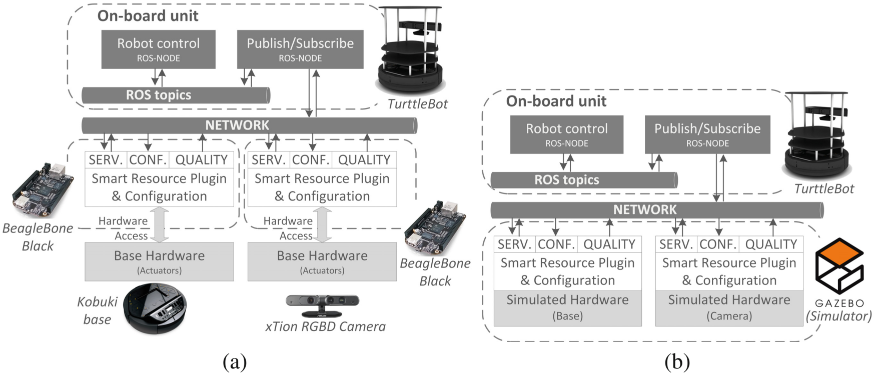


Fig. 3 Turtlebot HW and SW decoupled setup using sensorial Smart resource. a) Physical Smart Resources. b) Simulated Smart Resources.

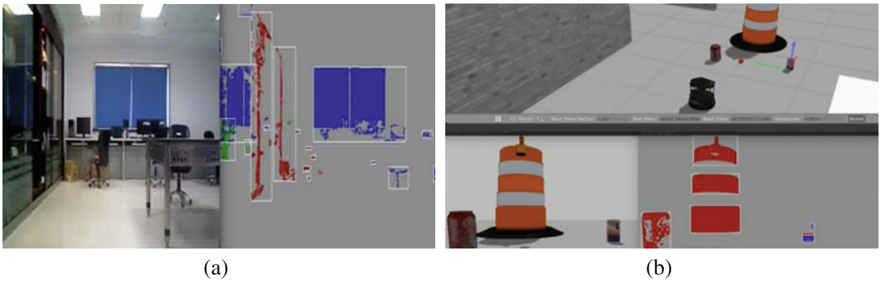


Fig. 4 Xtion Smart Resource: Blob color service real (a) and simulated (b) implementation

5 Conclusions

Since the implementation of ROS into robotic researches and development is increasing everyday the definition of a ROS ready Smart Resource offers a high level of versatility.

Adding these resources into the ROS topology has been proved to provide a standardized way to implement isolated procedures, which can be defined as abstract services that manage high-level information.

Smart Resources also offer a high configuration capabilities and performance monitoring. As a result we obtain a decoupled system in which every resource provides a service adapted to perform in the most optimum way and can be accessed by means of ROS topics, being easily integrated in every ROS based platform and highly reusable.

As a future work, it will be studied how to develop a ROS node to provide a high-level sensor fusion that relies on the information provided by the Smart Resources.

For this goal it will make use of the alarms and reconfiguration mechanisms to increase the fusion reliability without increasing the computational load of the central unit.

Acknowledgements Work supported by the Spanish Science and Innovation Ministry MICINN: CICYT project M2C2: Codise de sistemas de control con criticidad mixta basado en misiones TIN2014-56158-C4-4-P and PAID (Polytechnic University of Valencia): UPV-PAID-FPI-2013.

References

1. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5 (2009)
2. Munera, E., Poza-Lujan, J.-L., Posadas-Yagüe, J.-L., Simó-Ten, J.-E., Noguera, J.F.B.: Dynamic reconfiguration of a rgbd sensor based on qos and qoc requirements in distributed systems. Sensors **15**(8), 18080–18101 (2015)
3. Meijer, G.C.M., Meijer, C.M., Meijer, C.M.: Smart sensor systems. Wiley Online Library (2008)
4. Liu, F., Narayanan, A., Bai, Q.: Real-time systems (2000)
5. Cousins, S.: Exponential growth of ros [ros topics]. IEEE Robotics & Automation Magazine **1**(18), 19–20 (2011)
6. Elkady, A., Sobh, T.: Robotics middleware: A comprehensive literature survey and attribute-based bibliography. Journal of Robotics (2012)
7. Cousins, S., Gerkey, B., Conley, K., Garage, W.: Sharing software with ros [ros topics]. IEEE Robotics & Automation Magazine **17**(2), 12–14 (2010)
8. Remy, S.L., Brian Blake, M.: Distributed service-oriented robotics. IEEE Internet Computing **15**(2), 70–74 (2011)
9. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), vol. 3, pp. 2149–2154. IEEE (2004)
10. Chen, H., Cheng, H., Zhang, B., Wang, J., Fuhlbrigge, T., Liu, J.: Semiautonomous industrial mobile manipulation for industrial applications. In: 2013 IEEE 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER), pp. 361–366. IEEE (2013)
11. Estefó, P., Campusano, M., Fabresse, L., Fabry, J., Laval, J., Bouraqad, N.: Towards live programming in ros with pharos and lrp (2014). arXiv preprint arXiv:1412.4629
12. Saraydaryan, J., Jumel, F., Guenard, A.: Astro: Architecture of services toward robotic objects. International Journal of Computer Science Issues (IJCSI) **11**(4), 1 (2014)
13. Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: Ros for non-ros users. In: Proceedings of the 15th International Symposium on Robotics Research (2011)
14. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.-M.: The many faces of publish/subscribe. ACM Computing Surveys (CSUR) **35**(2), 114–131 (2003)

Multi-Agent Energy Markets with High Levels of Renewable Generation: A Case-Study on Forecast Uncertainty and Market Closing Time

Hugo Algarvio, António Couto, Fernando Lopes, Ana Estanqueiro and João Santana

Abstract This article uses an agent-based system to analyze the potential impacts of variable generation on wholesale electricity markets. Specifically, the article presents some important features of the agent-based system, introduces a method to forecast wind power, and describes a case study to analyze the impact of both wind forecast errors and high levels of wind generation on the outcomes of the day-ahead market. The case study involved two simulations: a base case, where the market closes at 12:00 noon and the bids of a wind producer agent are based on a forecast performed 12 to 36 hours ahead, and an updated forecast case, where the market closes at 6:00 p.m. and the bids of the wind producer agent are based on an updated forecast. The results indicate that wind power forecast uncertainty may indeed influence market prices, highlighting the importance of adaptations to the market closing time.

Keywords Software agents · Agent-based simulation · Electricity markets · Renewable generation · Wind power forecasting

1 Introduction

Electricity markets (EMs) are systems for effecting the purchase and sale of electricity using supply and demand to set prices. EMs differ from their more traditional counterparts because energy cannot be efficiently held in stock, and thus market participants are forced to work with consumption prognoses, which create a number

H. Algarvio(✉) · J. Santana
Instituto Superior Técnico, Universidade de Lisboa, INESC-ID, Lisboa, Portugal
e-mail: hugo.algarvio@tecnico.ulisboa.pt, jsantana@ist.utl.pt

H. Algarvio · A. Couto · F. Lopes · A. Estanqueiro
LNEG-National Research Institute, Est. Paço do Lumiar 22, Lisboa, Portugal
e-mail: {antonio.couto,fernando.lopes,ana.estanqueiro}@lneg.pt

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_37

of risks. In particular, financial risk management is often a high priority due to the substantial price and volume risk that markets can exhibit.

The penetration of stochastic renewable energy sources (RES), or variable generation (VG), such as wind and photovoltaic, has increased significantly in recent years. Large penetrations of VG may reduce market prices due to their near-zero, zero, or negative-bid costs, increase price volatility because of their stochastic nature, and worsen the capacity utilization of conventional resources [1]. There is, therefore, a need to analize the potential impacts of VG on energy markets to determine if existing designs are still effective.

VG has several unique characteristics compared to traditional plants, notably has significant fixed capital costs but near-zero or zero variable marginal costs, has unique diurnal and seasonal patterns, and increases the variability and uncertainty of the net load [1]. A natural way of dealing with this variability and reducing the uncertainty is to use forecasting tools. In the case of wind power, the forecasting techniques have considerably improved over the last decade but still present errors due to the chaotic nature of the atmosphere. Furthermore, the farther ahead the horizon of the forecast, the more difficult it is to predict.

Numerous approaches to forecast wind power have been developed during the past few years and detailed reviews may be found in the literature (see, e.g., [2, 3]). The existing approaches can be classified into the following three categories [2]: statistical models, numerical weather prediction (NWP) models, or a combination of both. NWP models have become an indispensable tool for wind power producers participating in electricity markets. However, one of the main sources of errors in wind power forecasts, even when coupled with statistical approaches, is the data provided by these models. Initial and boundary conditions (ICs) are necessary for numerical calculations as well as digital maps of the terrain topography and roughness. These ICs are usually obtained from global models, such as the global forecast model system (GFS) with six hour intervals, taking into account the assimilation of several different types of meteorological observations. In this work, we consider a K-nearest neighbor method to compute wind power forecasts.

An ongoing study is looking at using software agents to help manage the complexity of competitive electricity markets. In particular, we are developing an agent-based tool enabling market participants to submit their hourly offers to a day-ahead pool energy market [4]. The tool also enables market participants to negotiate the terms and conditions of both forward contracts and contracts for difference [5]. In this paper, we focus on the impact of both wind power forecast errors and high levels of wind generation on the outcomes of the day-ahead market. Since in Portugal and Spain this market closes typically at 12:00 noon (or midday), and supposing wind producer agents that intend to participate in it, there would be a need to perform the inherent power forecasting at least 12 to 36 hours ahead. Accordingly, this paper presents a case study aiming at comparing the results of a simulation (base case) where: (i) the day-ahead market closes normally at 12:00 noon, and (ii) the bids of a wind power producer agent are based on a forecast performed 12 to 36 hours ahead, with a simulation (updated forecast case) where: (i) the day-ahead market closes at

6:00 p.m., and (ii) the bids of the wind power producer agent are based on an updated forecast (i.e., an improved forecast when compared with the wind production data).

The remainder of the paper is structured as follows. Section 2 deals with competitive electricity markets, placing emphasis on the day-ahead market. Section 3 is devoted to wind power forecasting and presents a new forecast model. Section 4 describes some important features of the agent-based simulation tool. Section 5 introduces the case study. Finally, concluding remarks are future avenues of research are presented in section 6.

2 Competitive Electricity Markets

Electricity is typically bought and sold through a two-settlement system involving a day-ahead market (DAM) and a balancing/real-time market (RTM). The DAM clears to meet bid-in load demand for an entire day, one day in advance. The RTM reflects the actual operation of the agents participating in the market. It sets prices and schedules to match the imbalances caused by the variability and uncertainty present in power systems. Many markets also have intermediate scheduling and pricing procedures on the hour ahead or a few hours ahead to facilitate balancing in advance of real time, particularly when the differing conditions from the DAM are apparent.

The pricing mechanism of most day-ahead markets is founded on the marginal pricing theory. Generators compete to supply demand by submitting bids in the form of price and quantity pairs, for example. Similarly, retailers and possibly other market participants submit offers to buy certain amounts of energy at specific prices. Schedules and prices are calculated from the market-clearing engine, and price-quantity pairs are settled for all market participants regardless of their actual performance.

There are two main variations of marginal pricing: system marginal pricing (SMP) and locational marginal pricing (LMP). In SMP, the generation bids are stacked in the merit order, and the market clearing price or spot-price is defined by the intersection of the associated curve with the cumulative load curve. LMP is a more complex variation of marginal pricing. Specifically, the optimization process is now subject to different constraints, such as line load-ability and voltage limits, and can include the supply of losses and other ancillary services necessary to support system operation. Typically, the system operator runs now an optimal power flow procedure that defines the energy price at each bus of the network.

Bilateral trades are defined by privately negotiated bilateral contracts—that is, agreements between buyers and sellers to trade electricity at specified terms—and represent a different form of trading electricity. Either physical or financial, bilateral contracts are typically negotiated weeks or months prior to their delivery and can include the following specifications: starting date and time, ending date and time, price per hour over the length of the contract, megawatt amount over the length of the contract, and range of hours when the contract is to be delivered.

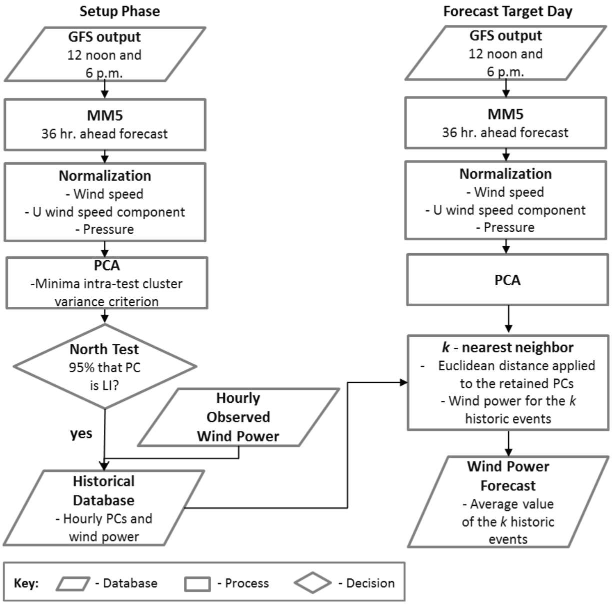


Fig. 1 Flowchart of the wind power forecast methodology

3 Wind Power Forecasting

In this work, we consider the K-nearest neighbor (K-NN) method [6] to obtain the deterministic forecasts using the NWP outputs. The recognition and use of this method—also known as analogous forecast—has increased due to: i) its facility of implementation and good effectiveness, and ii) the fact that it is a non-parametric approach, and thus no prior assumption needs to be taken regarding the distribution of the variables to predict. K-NN has been successfully applied to wind power forecast (see, e.g., [3, 6]).

The K-NN methodology assumes that the atmospheric circulation unleashes the same local impact, and therefore, the forecast of wind power production for a certain time can be obtained from wind production values for K analogous weather patterns from past events. A concern associated with this methodology is the need of a long observation period to determine a reasonable number of similar large-scale events [6]. Thus, we have filtered the freedom degrees—which represent only background noise—using a principal component analysis approach [7]. This approach enables the identification of the main dominant spatial-temporal synoptic variability modes. This was performed by canceling poorly correlated local effects that exist in the principal components (PCs). A sensitivity analysis based on the North test criterion [8] was then performed for each meteorological variable to determine the number of PCs that should be retained. The metric used to estimate the degree of similarity with each past event was based on the Euclidean distance, considering a trajectory matrix that contains the state of the atmosphere on the preceding times:

$$z_t = \sum_{i=1}^{n} \sqrt{\sum_{j=0}^{\bar{t}} \left(x_{i,t-j} - y_{i,j}\right)^2}$$

where z_t is the Euclidean distance, $x_{i,t-j}$ is the vector with the n retained PCs of NWP data for the target hour to forecast (t), $y_{i,j}$ is the vector with the n retained PCs of NWP historical data, and $\bar{t}$ is the number of lags in the trajectory matrix. Figure 1 depicts the flowchart of the methodology employed.

Sensitivity studies using two years of data to explore the impact of selecting different parameters (e.g., the suitable number of K nearest neighbors) were conducted. After several tests, it was decided to use a trajectory matrix with a lag of three hours, allowing to identify the data trajectory based on the first three PCs of wind speed, longitudinal component of the wind, and atmospheric pressure. The suitable number assumed for K was 10, determined according to the normalized root mean square error (NRMSE) and correlation values.

4 Agent-Based System for Electricity Markets

An agent-based system for simulating energy markets is currently being developed using the JAVA programming language. The major components of the system include a graphical user interface, a simulation engine, and a number of domain-specific agents. The graphical user interface allows users to set agent-specific parameters, specify and monitor trading simulations, and gather simulation reports. The simulation engine controls all trading simulations. The agents are computer systems capable of flexible action and able to communicate, when appropriate, with other agents to meet their design objectives. They are equipped with a generic model of individual behavior and can exhibit goal-directed behavior [9].

More specifically, the agents represent typical market participants, including generating companies (GenCos), retailers (RetailCos), aggregators, large and small consumers, and a market operator. GenCos may own a single generating plant or a portfolio of plants of different technologies. RetailCos buy electricity in a wholesale market and re-sell it to customers in a retail market (typically, end-use customers that are not allowed, or do not want, to participate in the wholesale market). Aggregators are entities that support groups of end-use customers in trading electrical energy. Large consumers can take an active role in the market by buying electrical energy in the pool or by signing bilateral contracts (e.g., with producers). Small consumers, on the other hand, can buy energy from retailers and possibly other market participants.

The system supports two key market models: pool trading and bilateral contracting of electricity. A day-ahead market sells energy to retailers and buys energy from generating companies in advance of time when the energy is produced and consumed. The pricing mechanism is founded on the marginal pricing theory—both system

Table 1 Producer agents: technology, fuel type, maximum capacity and marginal cost

Agent	Technology	Fuel Type	Maximum Capacity (MW)	Marginal Cost (€/MWh)
P1	Renewable	Wind	4500	00.00
P2	Thermal	Coal	310	45.00
P3	Nuclear	Nuclear	1180	25.00
P4	Combined Cycle	Gas	830	54.50
P5	Combined Cycle	Gas	990	56.12
P6	Combined Cycle	Gas	1176	57.90

marginal pricing and locational marginal pricing are supported. Bilateral contracts can be either physical or financial obligations. Buyer and seller agents are equipped with a negotiation framework that handles two-party and multi-issue negotiation [10, 11]. Negotiation proceeds through a pre-negotiation phase, an actual negotiation phase, and a post-negotiation phase. Specifically, pre-negotiation is the process of preparing and planning for negotiation, actual negotiation the process of moving toward agreement, and post-negotiation the process of building commitment and implementing the final agreement.

5 Case-Study

This section presents a case study to analyze the impact of both wind power forecast errors and high levels of wind generation on the outcomes of the day-ahead market. The agents are six producers (representing the supply-side) and four retailers (representing the demand-side). Several key features of the producer agents are shown in Table 1, notably the generation technology (wind), the maximum capacity (4500 MW) and the near-zero production cost of P1.

The case study involves two simulations: a base case where the day-ahead market closes normally at 12:00 noon and the bids of P1 are based on a forecast performed 12 to 36 hours ahead, and an updated forecast case where the day-ahead market closes at 6:00 p.m. and the bids of P1 are based on an improved forecast (when compared with the production data). Figure 2 shows the hourly wind production data for P1 in a typical day, together with two wind power forecasts—that represent the bids of P1 at either 12:00 noon or 6:00 p.m.

The wind power forecast errors for the base case (12:00 noon) are naturally higher than the errors for the improved forecast case (6:00 p.m.). Also, the two forecasts consistently underestimate the wind generation profile (see Fig. 2). By comparison, the daily deviations from the production data are lower for the improved forecast case (−87.79 MWh, representing an error of 1.54%) than for the base case

Table 2 Simulation results: hourly market-clearing prices

Hour	Price (€/MWh)		Hour	Price (€/MWh)	
	Base case	Imprv. case		Base case	Imprv. case
1 a.m.	25.00	25.00	1 p.m.	56.34	56.34
2 a.m.	25.00	25.00	2 p.m.	56.34	56.34
3 a.m.	25.00	25.00	3 p.m.	56.34	56.34
4 a.m.	25.00	25.00	4 p.m.	56.34	56.34
5 a.m.	25.00	25.00	5 p.m.	56.34	54.50
6 a.m.	25.00	25.00	6 p.m.	57.90	57.90
7 a.m.	45.00	25.00	7 p.m.	56.11	56.11
8 a.m.	25.00	45.00	8 p.m.	56.11	56.11
9 a.m.	54.50	54.50	9 p.m.	54.50	56.11
10 a.m.	56.11	56.11	10 p.m.	54.50	56.11
11 a.m.	54.50	56.11	11 p.m.	54.50	54.50
12 noon	56.34	54.50	12 midnight	54.50	25.00

(−437.71 MWh, corresponding to an error of 15.81%). The correlation values between the production data and each of the wind power forecasts are positive: 0.90 (base case) and 0.99 (improved forecast case). Furthermore, the normalized root mean square errors are 4.67% (base case) and 1.56% (improved forecast case), i.e., the NRMSE is about 3 times higher for the base case.

The simulation results are presented in Table 2, namely the hourly market-clearing prices for both the base case and the improved forecast case. Clearly, closing the day-ahead market at either 12:00 noon or 6:00 p.m. leads to significant differences in the market-clearing prices over time. In particular, the wind forecast uncertainty

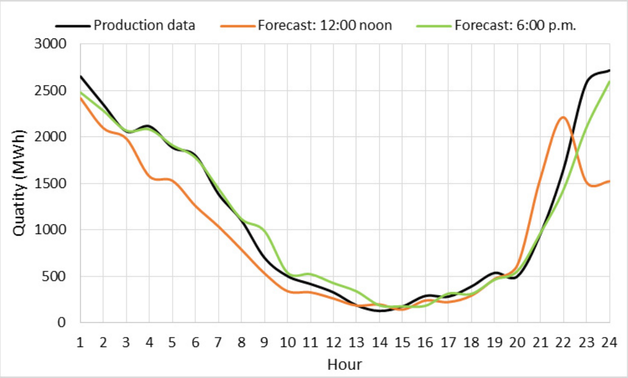


Fig. 2 Producer P1: production data and wind power forecasts

significantly influence the market prices at eight hourly trading intervals—that is, at 7 a.m., 8 a.m., 11 a.m., 12 noon, 5 p.m., 9 p.m., 10 p.m. and 12 midnight.

A final explanatory and cautionary note is in order here. The production data used in this work (see Fig. 2) is based on real production data from a Portuguese wind power plant with an installed capacity of 114 MW. Also, the two wind power forecasts shown in Fig. 2 were obtained directly from forecasts performed for the Portuguese plant. Thus, in order to assess the impact of a high level of wind generation, we considered production data and wind power forecasts for agent P1 (installed capacity of 4500 MW) based on a smaller Portuguese wind power plant, i.e., the real data and the forecasts were increased by a factor.

6 Conclusion

This paper has presented some important features of an agent-based system for simulating energy markets, introduced a method to forecast wind production, and described a case study to analyze the impact of high levels of wind generation on market outcomes. The case study involved two simulations: a base case, where the day-ahead market closes at 12:00 noon, and an improved forecast case, where the market closes at 6:00 p.m. The results indicate that forecast uncertainty may influence market prices, highlighting the importance of adjusting the closing time of the day-ahead market to deal with wind power variability and uncertainty. As this research progresses, we aim at using the system to study in more detail market design with increasing levels of renewable generation.

Acknowledgement This work was supported by "Fundação para a Ciência e a Tecnologia" with references UID/CEC/50021/2013 and PD/BD/105863/2014.

References

1. Ela, E., Milligan, M., Bloom, A., Botterud, A., Townsend A., Levin, T.: Evolution of Wholesale Electricity Market Design with Increasing Levels of Renewable Generation. Report NREL/TP-5D00-61765, Golden, USA, September 2014
2. Giebel, G., Brownsword, R., Kariniotakis, G., Denhard, M., Draxl, C.: The State-Of-The-Art in Short-Term Prediction of Wind Power: A literature Overview. Deliverable Report D-1.2, Project ANEMOS.plus, Contract 038692, DTU (2011)
3. Jung, J., Broadwater, R.: Current Status and Future Advances for Wind Speed and Power Forecasting. Renew. Sustain. Energy Rev. **31**, 762–777 (2014)
4. Vidigal, D., Lopes, F., Pronto, A., Santana, J.: Agent-based Simulation of Wholesale Energy Markets: a Case Study on Renewable Generation. In: 26th Database and Expert Systems Applications (DEXA 2015), pp. 81–85. IEEE (2015)
5. Lopes, F., Rodrigues, T., Sousa, J.: Negotiating Bilateral Contracts in a Multi-agent Electricity Market: A Case Study. In: 23rd Database and Expert Systems Applications (DEXA 2012), pp. 326–330. IEEE (2012)

6. Alessandrini, S., Dav, F., Sperati, S., Benini, M., Delle Monache, L.: Comparison of the Economic Impact of Different Wind Power Forecast Systems for Producers. Advances in Science and Research **11**, 49–53 (2014)
7. Couto, A., Costa, P., Rodrigues, L., Lopes, V., Estanqueiro, A.: Impact of Weather Regimes on the Wind Power Ramp Forecast in Portugal. IEEE Transactions on Sustainable Energy **6**(3), 934–942 (2015)
8. North, G., Bell, T.: Sampling Errors in the Estimation of Empirical Orthogonal Functions. Monthly Weather Review **110**(7), 699–706 (1982)
9. Lopes, F., Algarvio, H. Coelho, H.: Bilateral Contracting in Multi-agent Electricity Markets: Negotiation Strategies and a Case study. In: 10th International Conference on the European Energy Market (EEM-13), pp. 1–8. IEEE (2013)
10. Lopes, F., Coelho, H.: Concession Strategies for Negotiating Bilateral Contracts in Multi-agent Electricity Markets. In: 23rd International Workshop on Database and Expert Systems Applications (DEXA), pp. 321–325. IEEE (2012)
11. Lopes, F., Coelho, H.: Negotiation and Argumentation in Multi-agent Systems. Bentham Science, The Netherlands (2014)

Efficient Results Merging for Parallel Data Clustering Using MapReduce

Abdelhak Bousbaci and Nadjet Kamel

Abstract Data clustering is partitioning data into sub-groups using a distance measure. Clustering a large data amount requires an important execution time. Several works have been proposed to overcome this problem using parallelism. One of the parallel techniques consists in partitioning data and processing each partition apart, the results obtained from each partition are merged to get the final clusters configuration. Using an inappropriate merging technique leads to an inaccurate final centroids and a middling clustering quality. In this paper, we propose two merging techniques to improve the clustering quality.

In a first solution, the results are merged using the K-means algorithm, and in a second one using the genetic algorithm. The results proved the efficiency of the proposed strategies.

Keywords Data clustering · K-means · Parallelism · MapReduce · Results merging · Genetic algorithm

1 Introduction

Data clustering is partitioning data into sub-groups using a distance function such that data in the same group are similar [1]. Clustering a large data amount requires an important execution time. In the literature, several works have been proposed to overcome this problem using parallelism [2] [3] . One of parallelism techniques consists in partitioning data and processing each partition apart [4] [5] [6], using a cluster of machines or a multi-core CPU. In both cases, the results obtained from each partition are merged to obtain the final clusters configuration.

A. Bousbaci(✉) · N. Kamel
LRIA, Computer Science Department, USTHB Algiers, Bab Ezzouar, Algeria
e-mail: abousbaci@usthb.dz

N. Kamel
Computer Science Department, Faculty of Sciences, UFAS Setif, Setif, Algeria
e-mail: nkamel@univ-setif.dz

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_38

The merging techniques have a direct impact on the clustering quality. Using an inappropriate fusion technique, results inaccurate centroids and a middling clustering quality.

One of the existing merging methods, consists in gathering the clusters that overlap in data space, according to a defined threshold. If we have two clusters A and B, and there are data points from one of these two clusters that can belong to the other one, then these two clusters are merged together and constitute a new single cluster. In [4], the authors proposed a parallelization for the k-means algorithm. In their solution, the initial data is divided randomly on several machines and each one processes its own data partition independently. In a final step, the clusters from all the machines that overlap in data space were merged. This method showed efficiency, but the obtained final number of centroids, can differ from the one set in the beginning (*K*). This can affect the clustering quality in the case where the initial chosen number of clusters (K) is the optimal one.

Another existing simple merging technique consists in merging the centroids from the different partitions, such that each centroid from a given partition is merged with the *N-1* nearest centroids from each of the other partitions; where *N* is the number of all the partitions. This merging method is not efficient, according to [7], using such simple merging techniques leads to inaccurate new centroids. If there are two centroids A and B to merge, which have respectively M and N data points in their clusters; with $M > N$, the new centroid will be closer to $A's$ cluster data points. This will affect the performance of the algorithm.

In our previous work (*SPKmMR*) [6], we proposed a parallelization for the Sampling-PSO k-means algorithm (*SPKm*) [8] using MapReduce and shared memory parallelism, and a simple merging technique to obtain the final clusters.

To improve the *SPKmMR* algorithm [6], we propose two different strategies for an efficient final results selection. The first one is based on the K-means algorithm, we obtain the final configuration by clustering the centroids resulted from all the partitions. In the second one, we supposed that involving all centroids from the different partitions may not be the optimal solution, so we proposed to choose the best *K* centroids among the *N* centroids using genetic algorithm; where *N* is the number of all the centroids from the different partitions.

This paper is presented as follows: Section 2 introduces the related works to our contribution. In section 3, our proposed strategies are detailed. Section 4 contains the experimentation and results. Finally, in section 5 we present the conclusion and perspective for future works.

2 Background and Related Work

2.1 K-means

K-means is one of the most used clustering algorithm for its simplicity and efficiency. Several works based on k-means have been proposed. Otherwise its main objective which is clustering data, K-means has been used in several work to perform many tasks in relation with data clustering. In [9], k-means has been used for the initial centroids selection.

In [8], the authors used k-means algorithm for data sampling, by partitioning data into small sub-sets and applying k-means on each one, the resulted centroids from each partition represent the full data set.

2.2 Genetic Algorithm

Genetic algorithm (GA) is a meta-heuristic used in optimization problems. GA has been used in many clustering tasks. It is known for its efficiency in the initial centroids selection problem for the K-means algorithm. In [10], the authors proposed a solution for initial centroids selection based on GA. The algorithm select the K optimal centroids from the initial DataSet, the resulted centroids are then used as the starting clusters' centers for the K-means algorithm. In [11], an algorithm for clustering data has been proposed based only on the GA.

2.3 Parallel Sampling-PSO-Multi-core-K-means Algorithm using MapReduce (SPKmMR)

The algorithm presented in [6] is a parallelization of the Sampling-PSO-K-means algorithm (SPKm). In that work we proposed to parallelize the *SPKm* algorithm using MapReduce and shared memory parallelism. This algorithm can be summarized in the following steps:

- Initial data is divided on the set of machines.
- Apply *SPKm* on each machine using its data partition to find local clusters.
- Merge the clusters from the previous step to get the final solution.

This method showed efficiency and enhanced the rendering of the *SPKm* algorithm, but we aim to improve this algorithm by improving the merging step.

By analyzing each data partition's centroids, we noticed in some cases that the centroids resulted from a single partition give better results than the ones obtained after the merging process. This proved that the used merging technique failed to obtain an optimal solution and using an inappropriate merging technique provides inaccurate and malformed centroids.

Parallel clustering by partitioning is sensitive to the used merging method. We aim to propose an approach to improve the final selection step of our previous work [6]. We propose two techniques for the final selection process. The first one is based on k-means' work cited above, the use of k-means in initial selection and in sampling algorithms inspired us to use it in the merging step. The second approach is inspired from the hybrid genetic-k-means algorithm works cited above. In our case, we use the genetic algorithm to select the K best centroids from the set obtained from the different data partitions. The two proposed solutions will be detailed in the next section.

3 Proposed Approach

We present in this section our two contributions in the merging step for the *SPKmMR* algorithm. Our initial objective was to use k-means algorithm to get the final centroids configuration. This proposition involves all the centroids obtained from the different data partitions, as mentioned earlier, using all the centroids obtained from the different machines may not be optimal because of the random data distribution between the machines in the first step. For this reason we propose a solution where only the best K centroids are chosen in the final selection step using the genetic algorithm. Before detailing these two approaches we can summarize all the process in these steps:

1. Distributing randomly initial data input on the set of machines.
2. The algorithm *SPKmMR* is applied on each machine.
3. N local clusters are obtained from the previous step; with $N = K \times M$; where K is the number of clusters and M is the number of machines.
4. One of the proposed merging strategies is applied to obtain the final clusters configuration.

3.1 K-means Algorithm Merging

In this solution, we use the k-means algorithm for selecting the final configuration. In [8], the authors used the k-means algorithm for sampling the initial data and reduce its size. The resulted centroids from the sampling step can represent all the initial data set, so they applied the PSO-K-means algorithm on the results of the sampling step instead of applying it on the entire data set. In [6], we applied *SPKm* on many data partitions, and K centroids were obtained from each one.

We can consider each K centroids obtained from a data partition as the representative data points of this one, so the ensemble of all obtained centroids can represent as well the entire initial data set.

Thus, at the end of *SPKmMR*, we have N centroids; with $N = K \times M$; where K is the chosen number of clusters and M is the number of machines in the cluster. Finally, instead of applying the overlap technique for merging the centroids, we consider the N centroids as a sample for the entire initial data set. So we apply the k-means algorithm on these centroids to get the final K centroids.

3.2 Genetic Algorithm Merging

The second proposed strategy for the final selection step is based on the GA. As mentioned earlier, involving all the centroids may not be the optimal solution. Some centroids can be malformed due to a bad initialization when applying *SPKmMR*, or due to the random data distribution on the machines as proved in [12]. To avoid this, we propose to select among the set of centroids the K best ones. This can be seen as the k-means centroids initialization problem [10]. Therefore, we propose to use the genetic algorithm to select the best K centroids among the ones resulted from each data partition.In this solution, the genetic algorithm takes as input data the centroids set obtained from the executions of the *SPKmMR* algorithm on the different data partitions, the GA takes also as input data the entire initial data set, it is used when evaluating (fitness calculation) the obtained solutions (chromosomes evaluation) of the GA.

Population and Chromosomes. The population of the genetic algorithm is generated randomly from the centroids ensemble obtained from the *SPKmMR* algorithm executed on the different data partitions. N chromosome are generated, and at each iteration the best N ones are selected. Each chromosome from the population is a vector of K random centroids selected from the centroids ensemble. Each chromosome is a potential clusters solution.

Crossover and Mutation Operations. At each iteration of the genetic algorithm, new chromosomes are obtained by applying mutation and crossover operations on the population. The crossover operation combines two randomly selected solutions and generates two new solutions. The mutation operation in our case consists in changing randomly one of the K elements of a given chromosome and replace it by another one from the set of centroids. The best solutions are selected for the next iteration and so on until the last iteration. At the end of the genetic algorithm, the best obtained solution represents the final clusters configuration.

4 Implementation

To analyze our propositions, we implemented our previous algorithm SPKmMR and the two proposed merging strategies. The tests were done on “The Individual household electric power consumption” data set [13], which contains 2075259 data points with 9 dimensions.

Before staring the clustering process, data is preprocessed. This step is very important to obtain good results. In our case, it consists of cleaning data, eliminating the insignificant attributes, normalize the data values and finally each data point is represented with a numerical vector which fits with the MapReduce data structure (*Key/Value pair*).

4.1 Evaluation

To evaluate a solution we should determine how close the objects of a same cluster are. We use the formula (1) to calculate the average distance between a centroid and the data points of its cluster. The smaller this value is the better is the clustering quality. It is defined as follows:

$$f = \frac{\sum_{i=1}^{k} \left\{ \frac{\sum_{j=1}^{n_i} d(o_i, p_{ij})}{n_i} \right\}}{K} \tag{1}$$

with p_{ij} is the j^{th} data point in the i^{th} cluster; o_i is the center of the i^{th} cluster; $d(o_i, p_{ij})$ represents the distance between the data point p_{ij} and the centroid o_i; n_i is the number of data instances in the cluster C_i and K is the defined number of clusters. To calculate the distance between two data points we used the euclidean distance.

For the evaluation of the solutions (chromosomes) generated by the GA, we also used the formula (1), because evaluating a chromosome means to determine how compact is the clusters configuration contained in this chromosome.

5 Experimentation

To evaluate our proposed solutions, we implemented the following algorithms: K-means, Sampling+PSO+Kmeans (*SPKm*), k-means using MapReduce (*KmMR*) and Sampling+PSO+McK-means using MapReduce (*SPKmMR*).

We tested the proposed merging strategies with the parallel algorithms from the list above (*KmMR*, *SPKmMR*).

To evaluate our propositions, we used a cluster of 5 machines. Each node is equipped with a Dual Core CPU and 2 GB of RAM. The cluster runs on Linux Ubuntu 10.04. We used the framework Hadoop 1.2.1 which is an open source implementation of the MapReduce framework. The experiments were done on the normalized data set “The Individual household electric power consumption”.

Many algorithms have been implemented to test our approach, and each one has its own parameters. For the Sampling+PSO step, the used parameters are the same used in [6]. The table 1 summarizes the k-means algorithm step and the genetic algorithm (*GA*) parameters.

In this section we discuss the obtained results from the different implemented algorithms. To evaluate the clustering quality of the obtained solutions, we use the formula 1.

Table 2 shows the results of the implemented algorithm on “The Household Electrical Consumption” data set.

Table 1 Algorithms parameters

Number of clusters (K) in K-means	50-100 [7]
Number of iterations in k-means	25
Population size in GA	50
Number of generation in GA	100

Table 2 The performances of algorithms

		K = 50	K = 100
sequential algorithms	k-means	0.079233	0.067021
	SPKm	0.073251	0.063955
Simple merging	KmMR	0.077204	0.064242
	SPKmMR	0.071459	0.063574
K-means merging	KmMR	0.071763	0.061751
	SPKmMR	0.070317	0.059569
GA merging	KmMR	0.057541	0.051913
	SPKmMR	0.054363	0.049195

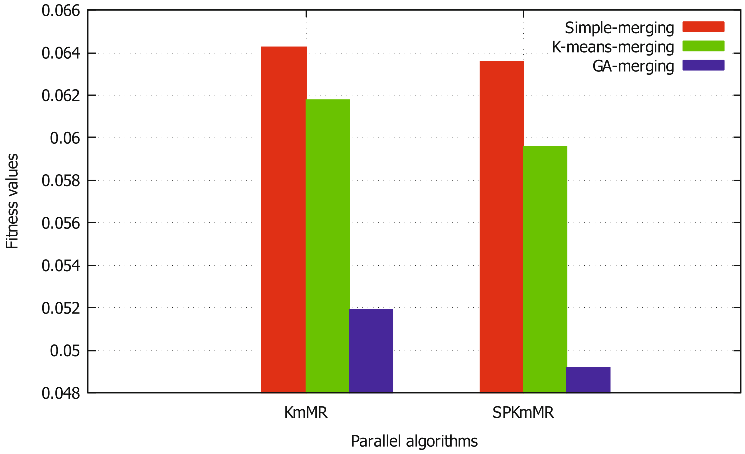
The results show that the proposed merging techniques give better results than the simple merging strategy. The simple merging strategy showed its inability to exploit the centroids obtained from different partition. We can notice this in the case of the *SPKmMR* algorithm where $K = 100$; the fitness value obtained after using a simple merging is equal to 0.063574 whereas the fitness obtained with the sequential algorithm *SPKm* is almost identical to it.

For the other cases of the simple merging strategy, we can see that for each algorithm, the results are slightly better than the ones obtained from those of the sequential versions.

The figure 1 illustrates the obtained results of the implemented parallel algorithms (KmMR and SPKmMR) with the different merging strategies in the case where *K=100*.

We notice that by using the k-means merging strategy we obtained a visible improvement compared to the simple merging strategy on both of the implemented parallel algorithms. We can see also that the GA merging technique gives the best fitness values especially when applied on our previous algorithm (SPKmMR) [6]. Another important point to highlight is that the improvement obtained with the SPKmMR algorithm compared to the KmMR algorithm changes according to the used merging strategy. We can see that the difference in the fitness value between SPKmMR and KmMR is more important when using K-means merging than when using the simple merging strategy. We can notice also that the difference in the fitness value between

Fig. 1 Performances of the merging strategies



SPKmMR and KmMR is the most significant when using the GA merging strategy compared to the two other merging techniques. This is explained by the fact that the GA merging strategy exploits the results generated by the SPKmMR algorithm better than the two other strategies.

6 Conclusion

In this paper, we proposed solutions to improve our work presented in [6]. We proposed to improve the merging step which showed deficiency in some cases. In a first place we proposed to use the K-means algorithm to get the final clusters configuration, next we suggested to use the genetic algorithm for selecting the K optimal centroids and consider them as the final solution. These two methods proved their efficiency by improving the performances of our previous work. In the Map phase of the MapReduce process, data is partitioned randomly on the cluster's machines. As future work, we aim to propose an efficient data partitioning approach to improve the clustering quality of our algorithm.

References

1. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, USA, vol. 1, pp. 281–297 (1967)

2. Ene, A., Im, S., Moseley, B.: Fast clustering using mapreduce. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 681–689. ACM (2011)
3. Guerrieri, A., Montresor, A.: Ds-means: distributed data stream clustering. In: Euro-Par 2012 Parallel Processing, pp. 260–271. Springer (2012)
4. Ferreira Cordeiro, R.L., Traina Junior, C., Machado Traina, A.J., López, J., Kang, U., Faloutsos, C.: Clustering very large multi-dimensional datasets with mapreduce. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 690–698. ACM (2011)
5. Mashayekhi, H., Habibi, J., Voulgaris, S., van Steen, M.: Goscan: Decentralized scalable data clustering. Computing **95**(9), 759–784 (2013)
6. Bousbaci, A., Kamel, N.: A parallel sampling-pso-multi-core-k-means algorithm using mapreduce. In: 2014 14th International Conference on Hybrid Intelligent Systems (HIS), pp. 129–134. IEEE (2014)
7. Cui, X., Zhu, P., Yang, X., Li, K., Ji, C.: Optimized big data k-means clustering using mapreduce. The Journal of Supercomputing **70**(3), 1249–1259 (2014)
8. Kamel, N., Ouchen, I., Baali, K.: A sampling-pso-k-means algorithm for document clustering. In: Genetic and Evolutionary Computing, pp. 45–54. Springer (2014)
9. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: ICML, vol. 98, pp. 91–99. Citeseer (1998)
10. Kwedlo, W., Iwanowicz, P.: Using genetic algorithm for selection of initial cluster centers for the k-means method. In: Artifical Intelligence and Soft Computing, pp. 165–172. Springer (2010)
11. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern recognition **33**(9), 1455–1465 (2000)
12. Hore, P., Hall, L., Goldgof, D.: A cluster ensemble framework for large data sets. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, vol. 4, pp. 3342–3347. IEEE (2006)
13. Lichman, M.: UCI Machine Learning Repository (2013)

Managing the Access to Medical Emergencies Services

Mateus Calado, Luis Antunes and Ana Ramos

Abstract Health is one of the most vibrant areas where Information Systems have been making a revolution in the way services are improving peoples quality of life. However, the optimization of provided health care services is not reaching medical emergency services in Angola, where there still is an incipient use of Information Systems applied to public health care services. We propose a web-based information system in which the core activities are implemented using a multi-agent system. Our goal is to guide the access to emergency care providing a feature of triage and referral of users based on significant personal information, stablished pre-diagnosis and considering available hospital services.

Keywords e-Health systems · Medical emergency services · Multi-agent systems

1 Introduction

In this paper we propose a solution to improve Angolan health care service. Some of the most important health indicators in Angola are below African average [1]. Changes are needed in the organization and functioning of Angolan public health system, taking into account human, technical, technological, logistical and administrative aspects. The variety of these factors suggests the need of a multi-disciplinary action to present a solution.

M. Calado(✉)
Faculdade de Cincias, Universidade Agostinho Neto, Luanda, Angola
e-mail: padoca@fc.uan.ao

L. Antunes
Faculty of Sciences, BioISI Biosystems and Integrative Sciences Institute, University of Lisboa, Campo Grande, Lisboa, Portugal
e-mail: xarax@ciencias.ulisboa.pt

A. Ramos
Departamento de Cincias da Computao, Universidade Agostinho Neto, Luanda, Angola

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_39

Both public and private institutions provide emergency medical services. In Angola private health care is unaffordable to the majority of the population, so people seeking emergency care use public emergency services. These services are available at general hospitals, as well as at specialty hospitals. Public medical emergency services suffer from several constraints such as insufficient and unevenly distributed human resources, insufficient and obsolete facilities and material resources, little coordination between entities and in-hospital services. These constraints, along with poor health conditions of Angolan population lead to overcrowded emergency departments, long waiting times, inefficient treatments, professionals' disincentive and high mortality in emergency rooms. In Angola due to logistic problems and incipient ambulance network, people frequently access emergency care in private transportation. It is not used a referral system and it does not exist an operational emergency center.

2 Our Proposal

Motivated by an earlier study presented in [2] we developed an information system to diagnose and refer patients to an adequate emergency care, thus improving access to emergency services. The aim of this proposal is to present a flexible, simple and intuitive interface that responds to the need of improvement and optimization of the access to emergency services, to facilitate access to information and to enable public health authorities a greater degree of knowledge about patients who are referred to them and a more balanced use of its resources.

This is a web-based system to support public medical emergency activities available on a multi-platform basis, to be used by patients, health professionals in emergency services and professionals in call centers. At the present development stage it is able to find medical emergency care in the province of Luanda, which is the most populated one of Angola.

This prototype provides several features: establishes a pre-diagnosis of patients based on described symptoms and personal data (e.g. age, health risk factors); classifies patients in one of three states (very urgent, urgent and less urgent) according to the urgency of care to be provided; directs patients to medical emergency facilities and informs the medical emergency service to where the patient is referred. Each one of the emergency services has waiting lines corresponding to the patients´s states. In the used approach the core activities are implemented using artificial intelligence agents.

2.1 Agents

There are several definitions of what is an agent, for instance in [3], [4], [5], [6]. To us, an agent is an abstraction of behaviors. It is also the most basic processing unit of an agent platform/environment. This aspect can be compared to objects in object-oriented programming: where objects have methods, agents have behaviors, but agents are autonomous and work in order to achieve objectives. Therefore, an agent has responsibilities, or tasks that it complies through its behavior. The behavior of an agent is build concerning the responsibilities of that agent.

Our agents are modeled using of the characteristics of autonomy, heterogeneity, reactivity, pro-activity and goal orientation in the way they are mentioned in [4]. To satisfy system requirements the following agents where implemented:

- User;
- Provider;
- Operator;
- Coordinator;
- Diagnose;
- Referral;
- Reporting.

Their main tasks are:

- User agent - is an interface agent (considering the classification proposed in [7]) that represents users seeking emergency health care services. The tasks performed by the user agent include aiding the user in using the application interface; requesting the creation or updating of user records; requesting information from the reporting agent on behalf of the user; validating information entered by the user (if necessary collaborating with the reporting agent); deciding to send requests from the user; sending diagnosis data to operator agent; asking the user data needed to make the diagnosis; inform the user about the diagnosis and referral established.
- Provider agent - is also an interface agent, and represents the entities that provide health services such as hospitals, clinics, health centers. Its tasks are to assist the health care provider with the application interface; validate information entered by the health care provider (if necessary collaborating with the reporting agent); request the creation or updating of records from the reporting agent; request information from the reporting agent on behalf of the health care provider and inform the health care provider about patient's diagnosis and referral (in collaboration with the coordinator agent).
- Operator agent - is a software agent responsible for receiving and filtering requests from the user agent. The operator agent asks for additional information required to make the diagnosis of the user; sends diagnosis information to the coordinator agent and requests data from the reporting agent.
- Coordinator agent - is responsible for coordinating the diagnosis and referral. Its tasks are to launch the diagnose agent and give it the information received from the operator agent; to give the operator agent the information requested by diagnose agent; to launch the referral agent and give it the corresponding information about the referral of the patient, namely personal data, location, diagnosis; to decide whether the process of diagnosis and referral is successfully completed; to inform the user agent about the outcome of diagnosis/referral; and to request data from the reporting agent.

– Diagnose agent – it is a collaborative agent responsible for implementing the diagnosis algorithm of patients. Thus, this agent is responsible for establishing a diagnosis; decide if it is necessary to find the user a service in a hospital entity; request additional information to establish a diagnosis; select the services needed to treat the patient according the established diagnosis; assign a level of urgency to the user (triage); send to the coordinator agent information about diagnosis and referral of the patient.
– Referral agent - manages congestion queues of hospitals and is responsible for referring patients to health care services. The referral agent has the tasks of asking the reporting agent additional information about the suitable services for the user (received from the coordinator agent) based upon the diagnosis; manage the congestion queues of hospital services; decide which entity will attend the patient based on the congestion of the hospital and the location of the user; inform the coordinating agent of the hospital/service to which the user is referred.
– Reporting agent - receives and filters requests from the user agent.

2.2 Diagnosis Process

Agents collaborate with each other in order to achieve the proposed objectives (e.g., make a diagnosis, give information to the user, inform the health care provider about referrals of patients. This collaboration is dynamic, to perform a certain task an agent often needs to interact more than once with another particular agent, and this is especially necessary when seeking to establish the diagnosis of a patient.

The diagnosis is modeled as a dynamic process (figure 1) and this results from the fact that hardly all relevant information to make a diagnosis can be given early in the process.

The process for establishing a diagnosis is made by the diagnose agent in 3 steps:

1. Gather information: age, gender, symptoms and health risk factors;
2. To formulate hypotheses of disease;
3. Apply the differential diagnosis to establish a diagnosis. The differential diagnosis is a systematic approaches used in medical practice for, based on evidence, to find out the cause (disease) of the symptoms presented by the patient, among several possible. The diagnose agent can explicitly request information necessary to evaluate the hypothesis of diagnosis (confirming or rejecting this hypothesis). The coordinator agent and operator agent give the requested information to the user agent. The operator agent can request information about the patient´s history to the report agent. The information provided by the user (e.g., indicating more symptoms or explicit risk factors) is used by the diagnose agent to continue the diagnosis process. There is a hierarchy of symptoms: certain symptoms indicate emergency situations.

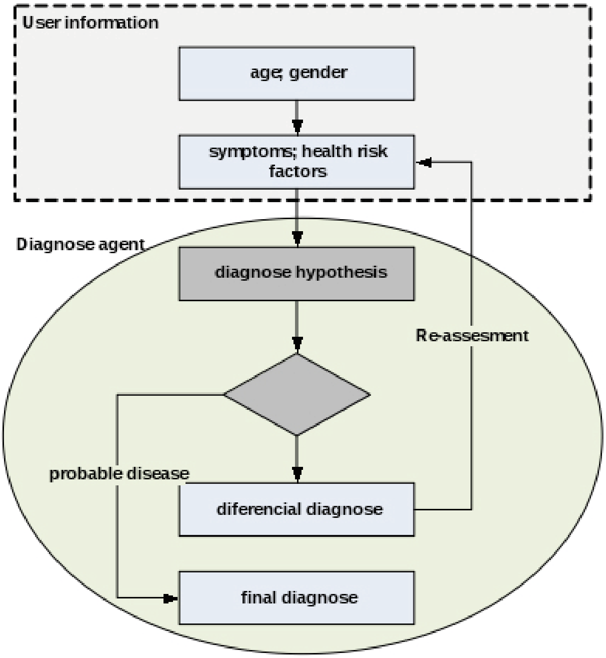


Fig. 1 Diagnosis flow chart.

2.3 *Selecting the Emergency Service*

The health status of the patient, his current location, the queues in emergency medical facilities and the services available will determine the hospital entity to which the patient will be directed. The chosen hospital receives the patient's personal information and the pre-diagnosis established. The order of preference for selecting the hospital is as follows:

1. Identify the hospitals with the necessary services to treat the patient;
2. Calculate patient´s waiting time for emergency medical treatment in each of these hospitals;
3. Validate the nearest hospital with the smallest waiting time;
4. Calculate patient´s distance to these hospitals;
5. Refer the patient to selected hospital.

In each hospital, patients' waiting queues are treated separately, considering the number of patients specifically in the three possible states (very urgent, urgent and less urgent). Whenever a triage is made, our system considers as eligible hospitals those with the smallest waiting time in queues for a severity greater or equal to the one of the given patient.

Figure 2 illustrates a scenario where the patient has been referred to an emergency service. Our system refers the patient to the emergency medical service and registers

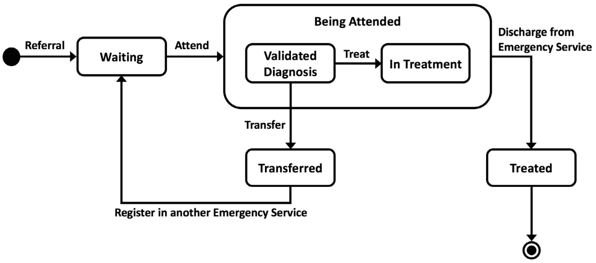


Fig. 2 Statechart of a patient in an emergency service.

Table 1 Simulation's referral results

Patients' condition	Referred patients (percent)
very urgent - general hospital	97
less urgent - medical center	81
specific health conditions	82
second referral	28

the patient in the waiting queue of the emergency service. When the patient is attended, we consider two steps: first a doctor or nurse who decides if the service is suitable for treating the patient evaluates him. If patients' condition is not adequate for the services provided by the hospital, then the patient will be referred through the system to another hospital with the diagnosis updated by the expert, while it proceeds to the 'Transferred' state. Otherwise goes to 'In Treatment' state, that is, emergency medical procedures are carried out. When the patient is discharged from the emergency department, he moves to the state 'Treated'. State transitions 'Attend', 'Treat', 'Transfer' and 'Discharge from Emergency Service' are registered in the system by the emergency department's staff (doctors or nurses). If the patient needs to be redirected to another emergency service, this can be done either automatically by our application or manually by medical staff.

3 Results

To evaluate the adequacy of the system, both simulations were run and users (patients and medical staff) tested the interface. Concerning the results obtained through simulations, we created a list of items that were quantified in order to assess whether the system can respond to current challenges in the access to Angola's hospital emergency services. Two of the studied subjects are the correction of patient's referral considering his severity status and also the existance of necessary services to treat the patient. Table 1 reflects the results obtained in referring patients to emergency services in Luanda. In the observed results, the percentage of very urgent patients referred to general hospitals meets the expectations. A small amount of these patients was not referred to general hospitals because they had a specific disease, having been referred to specialty hospitals. Eigthy one percent of patients with a less urgent state were referred to medical centers, the remaining received counseling and did not have to go to hospitals or medical centers. It was found that 28 percent of patients had to be referred for a second time, wich we consider a large value. This can be explained by the fact that patients do not insert the correct symptoms and this fact leads to an initial inadequate referral.

4 Conclusion and Future Work

In this article we propose the use of multi-agent paradigm to model the fundamental tasks of managing access to hospital emergency rooms, for example, to diagnose and to refer patients. Agents have properties of autonomy, cooperation, pro-activity and are modeled according to the Belief-Desire-Intention architecture proposed in [8]. The next challenges we address are:

- Implement components to deploy some features present in hospital management systems, in particular, allow doctors to actively manage patients' information. Such features are interesting in the Angolan scenario because the use of hospital management systems is very incipient.
- Improving security measures, considering the sensitive and confidential nature of data. Security measures have to provide a defence against attacks such as SQL Injection, Distributed Denial of Service, Cross-Site Scripting, and others.

References

1. World Health Organization - Global Health Observatory Data Repository. http://apps.who.int/gho/data
2. Calado, M., Antunes, L., Ramos, A.: Social simulation for optimization of emergency health services policy. In: IST-Africa Conference Proceedings, pp. 1–8. IEEE Conference Publications (2014)
3. Russell, S., Norvig, P.: Artificial Intelligence A Modern Approach. Prentice Hall, Upper Saddle River (2010)
4. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley and Sons (2009)
5. Maes, P.: Artificial life meets entertainment: lifelike autonomous agents. Communications of the ACM **38**(11), 108–114 (1995)
6. Hayes-Roth, B.: An architecture for adaptive intelligent systems. Artificial Intelligence **72**(1), 329–365 (1995)
7. Nwana, H.S.: Software agents: An overview. The knowledge engineering review **11**(03), 205–244 (1996)
8. Georgeff, M., Lansky, A.: Reactive reasoning and planning. In: Proceedings of the Sixth National Conference on Artificial Intelligence, Seattle, WA, pp. 677–682 (1987)

Real Time User Adaptation and Collaboration in Web Based Cognitive Stimulation for Elderly People

Carlos Rodríguez Domínguez, Francisco Carranza García,
María Luisa Rodríguez Almendros, María Visitación Hurtado Torres
and María José Rodríguez Fórtiz

Abstract Cognitive stimulation helps to improve the performance of daily life activities and life quality of elderly people. Most computer-based applications for cognitive training do not support user adaptation at run-time, and are designed for individual use, not for groups. Adaptation allows performing more effective ecological stimulation programs. The cognitive training in group is beneficial due to social interaction. This paper presents the solutions applied to a cognitive stimulation web application called VIRTRAEL, whose exercises can be adapted to the user at run time and they can be performed in a collaborative way.

Keywords Rule-based system for decision-making · Software engineering · Technology for internet · Middleware · e-Learning · Cognitive training for elderly people

1 Introduction

Research on active aging from different perspectives is the key for understanding how to live with better physical, social and mental conditions [1]. One of the main goals of active aging is maintaining the cognitive functions. In order to achieve this, it is necessary to stimulate abilities such as [2] memory, reasoning, attention, planning and organization, language, etc. In general, the cognitive slope affects the Activities of Daily Life (ADLs) performance since they cause dependence, especially to develop Instrumental Activities (IADLs). They are basic to carry out

C.R. Domínguez(✉) · F.C. García · M.L.R. Almendros · M.V.H. Torres · M.J.R. Fórtiz
University of Granada, Granada, Spain
e-mail: {carlosrodriguez,carranzafr,mrla,mhurtado,mjfortiz}@ugr.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_40

tasks implying the capabilities of decision-making and performing more difficult interactions with the environmental interactions, like managing money, performing housekeeping tasks, etc.

The effectiveness of preventing cognitive decline by means of stimulation in healthy seniors has been demonstrated in several clinical tests [3,4]. Some of those tests are based on the use of computer applications to stimulate the main cognitive areas, proving improvements on those areas and on IADLs [5].

Consequently, ITCs can provide new stimulation and training opportunities to improve the decrease in social, sensory-motor, cognitive and emotional functions in elderly people. Computer-based programs have been designed recently for cognitive stimulation in form of serious games [6].

However, until now, most of them present static, individual exercises that cannot be adapted to the users at run time.

In order to get run time adaptation (i.e., while the user is using the application), his/her interactions must be collected and taken into account to build a personal profile that allows the computer application to automatically adjust the exercises to each individual. Run time adaptations allow [4,7] (a) working the cognitive areas in which the user has more needs, (b) changing the order and difficulty level of the exercises depending on the user scores and requirements and (3) training in an ecological way, presenting scenarios and contents within the scope of his/her interests, to enhance user motivation and comprehension.

In addition to the adaptation, the performing of activities in-group is beneficial for the cognitive stimulation. Structured group activities (1) increase the sense of self-sufficiency; (2) stimulate personal skills; (3) decrease the sense of isolation, since the individuals feel identified with their partners and understood by them; (4) help learning from the peers' responses, and allow, through each one of the individual participation, the shared responsibility of providing results to the whole group [8]. In case of performing activities in a competitive way, the individuals usually increase his/her efforts in order to reach the maximum score, or at least better than his/her peers, which is a stimulus to continue working and improving.

In order to develop group exercises, it is necessary to implement a collaborative system supporting communication, cooperation and coordination. A collaborative system is, inherently, a distributed system whose intention is to provide computer-supported cooperative work (CSCW) [9].

In this paper we present VIRTRAEL (VIRtual platform for evaluation and TRAining of ELderly people). It provides exercises for evaluation and stimulation of cognitive skills, supports run time adaptation and provides mechanisms to support the cognitive training in group (based on adaptive learning systems fundamentals).

The rest of the paper is organized as follows. Section 2 introduces the foundations of this work. In Section 3, we describe our proposal to support adaptation and collaboration in VIRTRAEL. Finally, Section 4 presents some conclusions and outlines some future work.

2 Foundations

Some platforms and tools have been designed to promote an active and healthy aging, in aspects as learning, communicating, networking, health caring, etc. Some examples of related projects are OASIS (http://www.oasis-project.eu), SEACW (http://www.seacw.org) and AGE Platform Europe (http://www.age-platform.eu). Rehacom (http://rehacom.us), Cogmed (http://cogmed.com) and Cognifit (https://cognifit.com) are examples of well-known commercial platforms for cognitive training using ICTs. They offer exercises to train several mental abilities, and some of them can be adapted to the users' needs but not in run-time. Besides, none of them allow to perform group activities. Hulme and Melby-Lervag [10 analysed these three tools (and others) concluding that, in despite of their wide use, none of them is supported by a detailed task analysis by which any improvements could be expected. For that reason, more empirical evidence should be presented in the future.

The Computer-Supported Collaborative Learning (CSCL) are an emerging area of learning sciences concerned with studying how people can learn together with the help of computers [11]. Because of the emergence of CSCL in education, there have been various proposals for improving student learning by defining a space, set of activities and collaboration tools. Some authors [12] claim that such proposals enhance and stimulate cognitive ability in members of the collaborative learning. Some proposals and systems related to this area are CLS-KM [13] or GAOOLE [14].

The Rule-Based systems allow managing and storing information. They are used to make deductions and choices by means of if-then rules that incorporate the knowledge. They adaptively can determine the best sequence of rules to be run and apply them. Some of their uses are component selection, product configuration, quality assurance or troubleshooting. In general, they analyze situations, preferences or requirements and suggest lists of selections, treatments or preventative measures [15].

ECA (event-condition-action) rules are used to model active rules in event-driven architectures. An inference engine detects events and manages rule execution [16]. Rules can be modified at run-time, thus supporting workflow adaptations to meet new requirements [17].

Some Rule-Based systems support ECA model. There is a review of different rule-based approaches in [18] that includes the RuleML mark-up language [19] and Complex Event Processing [20].

The computer-assisted collaborative learning and rule-based systems offer a high potential because allow to take decisions to custom exercises and to track exercises performing, encourage social collaboration. Those must be considered to design systems that stimulate cognitive skills in elder people. The next section presents a particular computer-mediated platform named VIRTRAEL, based on these systems.

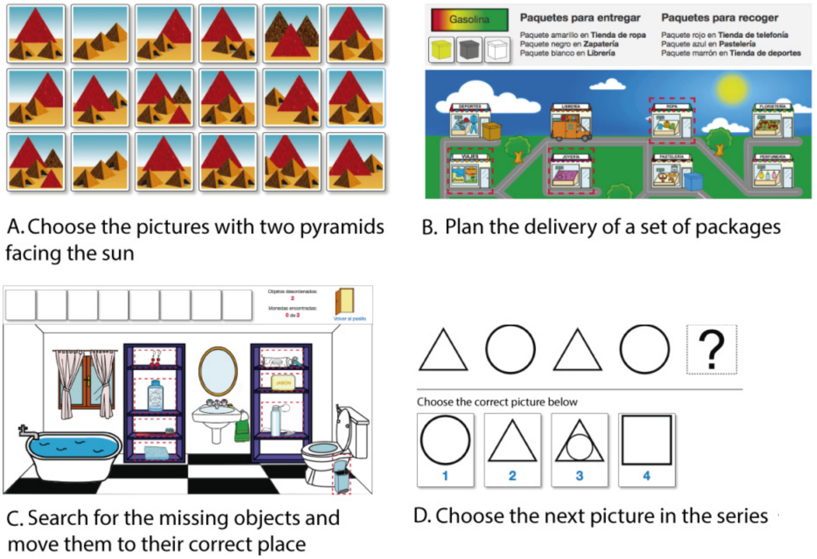
3 Proposal to Support Adaptation and Collaboration

Before to present how to support adaptation and collaboration, we are going to describe the computer-assisted training and evaluating platform for elderly people.

3.1 VIRTRAEL Platform

VIRTRAEL (http://virtrael.es) is a web platform that can be considered a CSCL for elderly people because helps to train cognitive skills in a collaborative way. It is composed by three elements: an exercises area (Fig. 1), a configuration tool, and a communication tool.

The **exercises area** includes 18 different exercises or games distributed in predefined work sessions. Each exercise evaluates or stimulates (depending on the session) one or more of the following skills: memory, planning, reasoning and attention. The exercises require the user to read instructions and, according to them, give an answer by selecting between a set of items (text or images) or writing in some specific cases. During the performance of an exercise, several measures are taken to evaluate the success, playtime, number of failures, omissions and hits. Exercises can be carried out individually or in group. The exercises' difficulty level and user interface are adapted to each user while they are performing. Some of the exercises have been designed as 3D games [21] in order to improve the user experience.


Gasolina
Paquetes para entregar
Paquete amarillo en Tienda de ropa
Paquete negro en Zapatería
Paquete blanco en Librería
Paquetes para recoger
Paquete rojo en Tienda de telefonía
Paquete azul en Pastelería
Paquete marrón en Tienda de deportes
DEPORTES
LIBRERIA
ROPA
FLORISTERIA
VIAJES
JOYERIA
PASTELERIA
PERFUMERIA
Objetos desordenados:
2
Monedas encontradas:
0 de 2
Volver al pasillo
JABÓN
?
Choose the correct picture below
1
2
3
4

A. Choose the pictures with two pyramids facing the sun

B. Plan the delivery of a set of packages

C. Search for the missing objects and move them to their correct place

D. Choose the next picture in the series

Fig. 1 Some exercises included in VIRTRAEL.

The **configuration tool** is used by an administrator to assign end users (patients) to a supervising therapist. Each therapist can also configure which exercises are going to be carried out by each patient, and in which order. The supervisor can track, at real time, the progress of each patient while performing each exercise, querying scores and measures, and deciding if the next planned game should be skipped or not. In the case of performing an exercise in group, the therapist creates the group, decides the kind of collaboration between the partners (cooperative or competitive) and how to play: turns, etc.

The **communication tool** provides both a forum to exchange messages between the stakeholders in the cognitive evaluation and training, and a chat to allow real-time collaboration between end users and therapists. This tool is intended to improve user engagement, decrease the sense of isolation and increase the sense of collaboration with other people.

VIRTRAEL has been designed and implemented as a web tool because the web provides portability, following standards, and then the tool can be used by more people with different operating systems and devices. Another reason is that we can create and run several instances of one or more exercises at the same time, using a mechanism to support that distribution and the communication between them.

3.2 Adaptation

Each exercise has been designed and implemented following the MVC (Model View Controller) architecture.

The model is composed by a XML file that describes the general aspects of an exercise and its elements (images or texts), a directory of resources associated to these elements and the code to define the story board or workflow of the exercise (including if it has different steps with mini-exercises, repetitions of the same exercise with the same or different elements, etc.). There is a different type of XML for each type of exercise; an exercise is one instantiation. The model also includes a database that stores the measures taken when the exercises are performed by each specific user: the time spent to complete each part of an exercise, number of fails, omissions and successes, time between interactions, number of helps required, etc. The persistence of the model must be guaranteed in case the web browser fails, or the user reloads the website, etc. So, we also store the data locally in the user personal device.

The view is a set of different widgets associated to the model. We have prepared widgets for different navigators in which the platform could run.

The controller gets the user interaction, and interconnects the view with the model while the exercise is running.

One of our goals is provide a program of exercises to be performed by an elder without the constant support of a therapist. Then, the platform must be proactive and take the best decision about the adaptations that must be done.

We need to support two kinds of adaptations in the exercises at run-time: (1) user interface due to the device used by the user (view) and (2) contents and structure of the exercise due to the difficulty level shown to the user (model). Both kinds of adaptation require an automatic decision taking mechanism. We have designed an internal rule-based system with a set of “if-then” rules that are triggered whenever certain predicates are satisfied. For instance, if a user repeats an exercise several times and he/she does not correctly complete it in a row, then the next time fewer contents could be shown and an exercise tutorial could be presented.

The predicate of a rule can refer to the user profile (cognitive skills, personal information, etc.), the partial exercise results, the kind of exercise, the amount of repetitions made, the kind of user’s device, etc.

The actions to be carried out if the predicate is satisfied can be: modify any internal property of any exercise to increase or decrease its difficulty and changing its workflow (e.g., the number of possible choices as answers to a question, the amount of help required, the number of on-screen elements or the number of steps) or any part of its presentation (the interaction mode, the widgets to use, colours, typography, etc.).

Regarding the rules referring to user devices, they help providing adapted user interfaces depending on the most appropriate interaction mode for each device, for example touch screen or mouse. The set of rules can be easily changed and, in fact, are currently being modified according to the feedback provided by therapists and users during pilot studies. When a predicate is satisfied and the action is performed, the instance of the exercise running is changed to include the new properties or resources, which influences its model (XML) or its view.

3.3 Collaboration

We have included in the platform a component responsible for managing the users of a work group and determining if they work in a collaborative or competitive way. It also maintains the awareness of the patients, showing the results of the interaction of a user to his peers or to a supervising therapist. In consequence, the component must exchange information between different instances of an exercise model and presentation, each instance running on different devices and by different users.

Therefore, two or more users can share an instance of a running exercise. The key internal mechanism to support exercise sharing is a middleware that provides a high level API to facilitate information exchanging between non co-located peer users (button clicks from other users, cursor movements, text inputs, etc.), distributing interaction events, and separating low-level communication management from the exercises logic.

The middleware used is BlueRose [22], and it includes publish-subscribe communication model to make more effective the distributed events among different VIRTRAEL instances. Different ways of collaborating are provided by this component: different users concurrently interact with the same exercise to complete it; turn-based collaboration; some users interact with the exercises, while others visualize their interactions; unstructured communication (chats).

The platform enables to maintain the awareness of each user's activities. Due to the limitations in the speed of the communications between on Internet, and in order to make a more effective system, we have also limited the kind of awareness, and the frequency and kind of events to be published.

3.4 Evaluation

A pilot study was carried out to evaluate the usability and end-user benefits of VIRTRAEL (formerly called PESCO). The usability analysis was based on heuristics and satisfaction tests on over 36 older people. Moreover, a comparative analysis of the traditional (paper-based) exercises in comparison with the exercises provided by VIRTRAEL was carried out in a sample of 44 seniors [23], including individuals with and without mild cognitive impairment. The results showed that the experimental group (i.e., the group using VIRTRAEL) presented significant improvements in visual attention and planning, while the control groups (i.e., the group using traditional exercises) did not. Regarding usability, the study concluded that VIRTRAEL was considered as usable by end users, but it suggested some room for improvement.

Considering those results, VIRTRAEL is currently being tested with a sample of more than 150 users to check that its usability has been improved in the latest versions, and to demonstrate the effectiveness of the stimulation over a longer period. Moreover, adaptation rules are being modified according to end-users' feedback.

The evaluations have been/are being performed by specialists in neuropsychology collaborating with our research team. Finally, users are invited to continue using VIRTRAEL after the evaluations have finished.

4 Conclusions and Future Works

VIRTRAEL is an open source and completely free web platform to support cognitive evaluation and intervention in elder people. It has been designed by an interdisciplinary team composed by psychologists, doctors and software engineers.

One of our objectives is to provide a rapid and effective tool to adapt the difficulty level of the exercises to the end users' needs. To achieve that goal, at runtime, we retrieve a data collection from the exercise performed by each user at any given moment. That collection is used by a rule-based system to infer a set of changes to apply to the exercises while the users are performing them. We use of a rule-based system in order to enable an easier run-time adaptation and efficient inference mechanisms.

Another objective is to provide a portable platform with a responsive design that self-adapts to the characteristics of the user device/s (e.g., screen size). That objective has been satisfied by promoting a modular design and an implementation based on Web standards.

Regarding the possibility of performing collaborative training, we have designed VIRTRAEL as a CSCL to organize groups and present awareness, using also a middleware to facilitate communication, cooperation and coordination.

Moreover, the middleware that is being used by VIRTRAEL (BlueRose) has also been used in multiple projects developed by our research team. Particularly, we have applied it to control home-automation systems, synchronize tasks on mobile ad-hoc networks and notify events on health caring systems and Law Enforcement software.

Now, we are finalizing the second study to evaluate VIRTRAEL, and we expect to obtain the results of its analysis soon.

Acknowledgements This project has been funded with support from the Regional Excellence Project TIC-6600, in Spain. We want to thanks them and also the institutions that collaborate in the study, and the participating seniors.

References

1. Boulton-Lewis, G.M., Buys, L., Lovie-Kitchin, J.: Learning and active aging. Educational Gerontology **32**(4), 271–282 (2006)
2. Talassi, E., Guerreschi, M., Feriani, M., Fedi, V., Bianchetti, A., Trabucchi, M.: Effectiveness of a cognitive rehabilitation program in mild dementia (MD) and mild cognitive impairment (MCI): a case control study. Archives of Gerontology and Geriatrics **44**, 391–399 (2007)
3. Ijsselsteijn, W., Nap, H.H., de Kort, Y., Poels, K.: Digital game design for elderly users. In: Proceedings of the 2007 Conference on Future Play, pp. 17–22. ACM, November 2007
4. Melenhorst, A.S.: Adopting communication technology in later life: The decisive role of benefits (Doctoral dissertation, Technische Universiteit Eindhoven) (2002)
5. Valenzuela, M., Sachdev, P.: Can cognitive exercise prevent the onset of dementia? Systematic review of randomized clinical trials with longitudinal follow-up. The American Journal of Geriatric Psychiatry **17**(3), 179–187 (2009)
6. Cameirão, M.S., Bermúdez, I.B.S., Duarte Oller, E, Verschure, P.F.: The rehabilitation gaming system: a review. Stud. Health Technol. Inform. 145(6) (2009)
7. Ines, D.L., Abdelkader, G.: Mixed reality serious games: the therapist perspective. In: 2011 IEEE 1st International Conference on Serious Games and Applications for Health (SeGAH), pp. 1–10. IEEE, November 2011
8. Bailey, J., Kingston, P., Alford, S., Taylor, L., Tolhurst, E.: An evaluation of Cognitive Stimulation Therapy sessions for people with dementia and a concomitant support group for their careers. Dementia, 1471301215626851 (2016)
9. Wilson, P.: Computer Supported Cooperative Work: An introduction. Springer Science & Business Media (1991)
10. Melby-Lervag, M., Hulme, C.: Is working memory training effective? A Meta-Analytic Review. Developmental Psychology **49**(2), 270–291 (2013)
11. Stahl, G., Koschmann, T., Suthers, D.: Computer-supported collaborative learning: An historical perspective. Cambridge Handbook of the Learning Sciences **2006**, 409–426 (2006)

12. Collazos, C.A., Guerrero, L.A., Pino, J.A., Renzi, S., Klobas, J., Ortega, M., Bravo, C.: Evaluating collaborative learning processes using system-based measurement. Educational Technology & Society **10**(3), 257–274 (2007)
13. Zhao, R., Zhang, C.: A Framework for collaborative learning system based on knowledge management. In: First International Workshop on Education Technology and Computer Science, 2009, ETCS 2009, vol. 1, pp. 733–736. IEEE, March 2009
14. Liu, S., Joy, M., Griffiths, N.: GAOOLE: a gaia design of agent-based online collaborative learning environment. In: Proceedings of the 8th European Conference on eLearning (ECEL 2009), pp. 339–350 (2009)
15. Hayes-Roth, F.: Rule-based systems. Communications of the ACM **28**(9), 921–932 (1985)
16. Bailey, J., Poulovassilis, A., Wood, P.T.: Analysis and optimisation of event-condition-action rules on XML. Computer Networks **39**(3), 239–259 (2002)
17. Goh, A., Koh, Y.K., Domazet, D.S.: ECA rule-based support for workflows. Artificial Intelligence in Engineering **15**(1), 37–46 (2001)
18. Paschke, A., Kozlenkov, A.: Rule-based event processing and reaction rules. In: Rule Interchange and Applications, pp. 53–66. Springer Berlin Heidelberg (2009)
19. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission, 21, 79 (2004)
20. Anicic, D., Fodor, P., Rudolph, S., Stühmer, R., Stojanovic, N., Studer, R.: A rule-based language for complex event processing and reasoning. In: Web Reasoning and Rule Systems, pp. 42–57. Springer Berlin Heidelberg (2010)
21. Rute-Pérez, S., Santiago-Ramajo, S., Hurtado, M.V., Rodríguez-Fórtiz, M.J., Caracuel, A.: Challenges in software applications for the cognitive evaluation and stimulation of the elderly. Journal of Neuroengineering and Rehabilitation **11**(1), 1 (2014)
22. Rodríguez-Fórtiz, M.J., Revelles-Moreno, J., Rute-Pérez, S., Rodríguez-Domínguez, C., Rodríguez-Almendros, M.L., Cano-Olivares, P., Hurtado-Torres, M.V.: Serious games for the cognitive stimulation of elderly people. In: 4th International Conference on Serious Games and Applications for Health (SeGAH), Orlando, Florida (2016)
23. Duque, R., Rodríguez, M.L., Hurtado, M.V., Bravo, C., Rodríguez-Domínguez, C.: Integration of collaboration and interaction analysis mechanisms in a concern-based architecture for groupware systems. Science of Computer Programming **77**(1), 29–45 (2012)

Performance Evaluation of Neural Networks for Animal Behaviors Classification: Horse Gaits Case Study

E. Cerezuela-Escudero, A. Rios-Navarro, Juan P. Dominguez-Morales, R. Tapiador-Morales, D. Gutierrez-Galan, C. Martín-Cañal and A. Linares-Barranco

Abstract The study and monitoring of wildlife has always been a subject of great interest. Studying the behavior of wildlife animals is a very complex task due to the difficulties to track them and classify their behaviors through the collected sensory information. Novel technology allows designing low cost systems that facilitate these tasks. There are currently some commercial solutions to this problem; however, it is not possible to obtain a highly accurate classification due to the lack of gathered information. In this work, we propose an animal behavior recognition, classification and monitoring system based on a smart collar device provided with inertial sensors and a feed-forward neural network or Multi-Layer Perceptron (MLP) to classify the possible animal behavior based on the collected sensory information. Experimental results over horse gaits case study show that the recognition system achieves an accuracy of up to 95.6%.

Keywords Multi-Layer Perceptron · Feed-forward neural network · Pattern recognition · Inertial sensors · Sensor fusion

1 Introduction

Behavior monitoring of wildlife animals is a hard technological task [1] due to several factors that need to be solved. (1) The development a lightweight and long batteries life (thus, low power consumption) devices to attach to the animal; (2) The design and implementation of a wireless network to collect the information

E. Cerezuela-Escudero · A. Rios-Navarro · J.P. Dominguez-Morales(✉) ·
R. Tapiador-Morales D. Gutierrez-Galan · C. Martín-Cañal · A. Linares-Barranco
Robotic and Technology of Computers Lab, ETS Informática - Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
e-mail: {ecerezuela,arios,jpdominguez,ricardo,dgutierrez,cmartin,alinares}@atc.us.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_41

from those devices; and finally, (3) download from the animals devices all captured information. Some commercial devices can track animals using global position systems (GPS) and obtain some of their vital signs through sensors. The information given consists only of raw and unprocessed data that require high bandwidth communications or high capacity memory cards and long life batteries, which usually are very heavy. Furthermore, these solutions are not able to recognize animal patterns from the obtained data.

Algorithms that look for particular behavioral patterns based on the input data usually conduct this kind of recognition or classification. Some of these algorithms are Neural Networks (NN), Support Vector Machines (SVM) or even complex statistical methods, which can detect specific behaviors such as sleeping, running, copulating, etc. Generally, the computational costs of these algorithms are highly enough to require specific platforms capable of parallelizing computations for this classification.

In this paper, a particular NN architecture is implemented for a behavioral classification of wildlife animals restricted to horse gaits, which are the ways a horse can move. The NN is designed and trained using a software tool and then all of its parameters are obtained and used on an embedded NN version implemented to run on a low-power microcontroller. There are some hardware platforms like SpiNNaker [2] that allows the easy development of spiking neural networks; however, the size and power consumption of this board are extremely high for this kind of task and targets.

MINERVA is a research project whose main aim is to study and classify wildlife behavior inside Doñana National Park [3]. To achieve this goal, a hierarchical wireless sensor network capable of transmitting and storing this information has to be set and tested inside this park. Moreover, in order to increment the value of the final product, an embedded system with energy harvesting techniques that will be able to digest sensor fusion data from inertial sensors [4], combined with other sensors (temperature, heart rhythm, GPS) has been developed in order to classify animal behavior in real time. This project has the additional aim of developing an infrastructure for collecting this information and make it accessible through the internet. The pattern recognition of the sensed data is performed in real time by the microcontroller using a low-power implementation of a NN that classifies three different horse gaits [12] (motionless, walking and trotting). This information is transmitted using a wireless multisensory network distributed on collars placed on some animals. This multisensory network reads data from the sensors and send them to a network of motes, which acts as a router and retransmits these packets to a base station. This base station receives the information through the network and uploads it to a remote server database. Researchers can access this data using a web-based user interface and track the animal activity at any time without the necessity of being in Doñana National Park.

The NN implementations presented can classify three different horse gaits. MATLAB (a mathematical software), with a toolbox for NN, has been used in order to design the architecture. The NN has been trained with data collected in Doñana with a prototype of the collar configured to capture and store raw data with a parameterized period of time. The NN training has been performed in the same software using 70% of the collected data. The remaining 30% has been used for the NN testing.

The paper is structured as follows: section 2 presents the collar device. Then, section 3 describes different fusion filters applied to the sensor data. Section 4 presents the NN architecture to classify three different behaviors. Section 5 describes the testing scenario and the results obtained. Finally, section 6 presents the conclusions.

2 Collars: Information Collection by Multiple Sensors

The aim of this collar is to gather information from the animal on which it is placed by using several sensors. Then, it will classify its animal behavior using this data as an input for a feed-forward NN implemented on the collar microcontroller. All detected patterns are locally stored in the collar memory. Finally, the collar will send the recognized gaits to a base station through XBee communications that will upload it to a database stored in a data server on the internet. The collar is provided with an inertial measurement unit (IMU), which consists of a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. This unit is used in addition to a GPS, which gives the position and time with respect to satellites. The IMU used in this work is MinIMU-9V2 [5], whose sensors have a resolution of 12 bits. The feed-forward NN architecture and training process is described in section IV.

The collar prototype, see Fig. 1, has an XBee module (XBee PRO S2B [6]) that can transmit data through a wireless network. XBee is the brand name for Digi International for a family of form factor compatible radio modules. XBee modules are integrated solutions based on ZigBee, which is an open global standard of the IEEE 802.15.4 MAC/PHY [7]. This device family allows to implement a mesh network of motes (or routers) where collars (or device) send information, and other elements (coordinators) of the network redirect these packets to a web server. The main objective is to transmit sensed information to the nearest router of the network, so that it can reach the coordinator and upload this information to the database. In such a case the signal cannot reach a valid point to transmit, i.e. the animal is out of the network coverage, the collar carries an SD card where the information is stored; so the animal behavioral information can be accessed later or offline, so it avoids data losing.

The periodical measurements of each sensor are carried out using a low power microcontroller (STM32L152 [8]) with a real-time operating system (RTOS) which is powered using a four AAA battery pack (1.5V, 1155 mAh each). Due to the fact that capturing an animal is very expensive, the process of obtaining data from each collar when it runs out of battery is organized as a task, allowing the microcontroller to switch to sleep mode if there are no router in the network coverage capable of receiving this collar's information. This increases batteries life. Moreover, the collar does not spend the whole time transmitting the information in a continuous manner. A periodic time is established for reducing radio transmissions and thus, reducing power consumption.

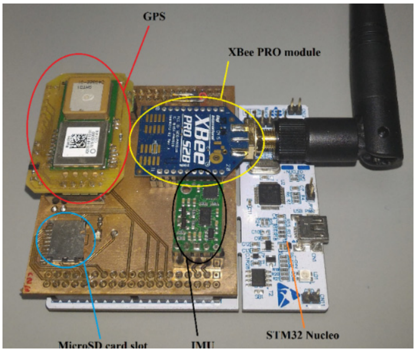


Fig. 1 Collar device prototype

3 Sensors Data Processing

Information from the sensors is enough to develop a NN able to recognize patterns from animals. Additionally, it is possible to increase the accuracy if more data are used during training. Sensor fusion obtains new combined information from sensors that can simplify and increase the accuracy of the NN architecture. This new information consists of three variables called pitch, roll and yaw, which are the angles of rotation of each axis of a three-dimensional space (pitch represents y-axis rotation, roll represents x-axis rotation and yaw the variation from z-axis). The purpose of this fusion is to generate new data for the NN and then compare results using data from sensors and from fusion algorithms.

An accelerometer is a sensor that measures the gravity acceleration at 3D each axis (x,y,z). It is possible to obtain pitch and roll calculation by applying mathematical formulas based on the combination of data collected by the accelerometer; however, if the accelerometer is not precise enough, then, small variations produce high signal-to-noise ratio (SNR) values. On the other hand, it is difficult to calculate the yaw value without combining the magnetometer tilt compensation and the gyroscope variation.

This paper uses two fusion algorithms: (1) FreeIMU [9] and (2) Kalman Filter [10]. FreeIMU, Fig. 2, is based on a quaternion representation. A quaternion is a complex number that represents the object orientation by four fields. The three first fields represent the orientation in each axis, while the last field represents the rotation of the object. The algorithm fuses the accelerometer and the magnetometer using the gradient descent algorithm when the fusion is finished. The gyroscope data are added to compensate the possible drift. Finally, in order to get pitch, roll and yaw, the values of the quaternion are combined. The second algorithm is the Kalman Filter, Fig. 2, which is a complex sensor fusion algorithm commonly used in control systems [10]. The principal advantage of this algorithm is that, by using initials estimators and dynamic parameters, the algorithm auto adjusts the output in time.

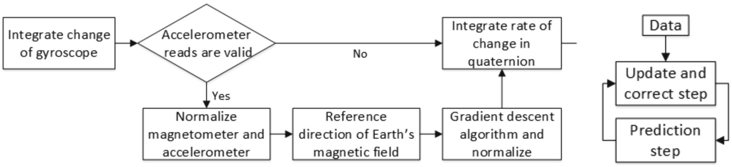

Integrate change of gyroscope
Accelerometer reads are valid
No
Yes
Integrate rate of change in quaternion
Normalize magnetometer and accelerometer
Reference direction of Earth's magnetic field
Gradient descent algorithm and normalize
Data
Update and correct step
Prediction step

Fig. 2 FreeIMU filter (left) and Kalman filter (right)

The prediction process generates an estimation of the next value and calculates the estimated error covariance. The second step updates the current value from the sensors and the error covariance using the different parameters mentioned before and the estimations calculated in the previous step. When each value is updated, it is sent to the prediction step. This feedback continues updating the error estimation and value predictions, so the algorithm converges over iterations. By using this algorithm it is possible to filter noise from each sensor generating pitch, roll and yaw. The main difference between these algorithms is the computational cost: Kalman filter is more accurate but requires a higher computational effort; on the other hand, FreeIMU does not have any feedback so it does not change filter parameters over time, adjusting the output to the data from the sensors. The idea of implementing both algorithms is to compare the results of the classification obtained in the NN and, with this output, decide which one is more adequate for an animal behavior classification.

4 Neural Network

4.1 Neural Network Architecture

This section presents the architecture of the used Multi-Layer Perceptron (MLP) NN. MLP is the most commonly used with the backpropagation algorithm: the multilayer feed-forward network. An elementary neuron with *R* inputs is shown in Fig. 3. Each input *p* is weighted with an appropriate *w*. The sum of the weighted inputs and the *bias* forms the input to the transfer function *f* [11]. Neurons can use any differentiable transfer function *f* to generate their output.

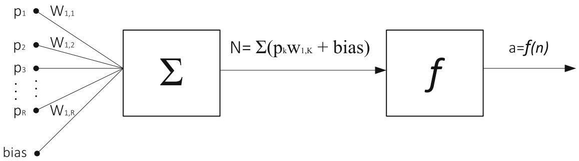


Fig. 3 Multi-layer Perceptron neuron

The topological structure of the MLP NN used consists of a two-layer feed-forward network, with a sigmoid transfer function in the hidden layer and softmax transfer function in the output layer. During this research, the optimal number of hidden units was found by running different performance tests, where a new MLP was created, trained and tested using a varying number of neurons in the hidden layer. The best performance was obtained using 10 neurons in the hidden layer. The number of neurons in the output layer is equal to the number of behaviors to be classified. In this manuscript, three different patterns are presented: motionless, walking and trotting. The results are presented for different input data: the raw data acquired from the IMU sensors (x, y and z for each sensor) and the output from the fusion algorithms presented before (roll, pitch and yaw). Therefore, the number of inputs in the NN directly depends on which data are used at the performance test. The network was trained using a scaled conjugate gradient backpropagation algorithm [13].

4.2 Neural Network Input and Target Data

The NN input data was the IMU sensors data collected from the collar while a horse performed three kinds of behaviors: motionless, walking and trotting. The sensors described in section 2 gathered data every 0.03 seconds, during a specific time period of 1 second (this is configurable), obtaining 2000 samples for each behavior (sampling frequency of 33 Hz).

The first performance test used the instantaneous raw sensor data and, consequently, the NN had *nine* inputs (x, y and z for each 3-axis sensor of the IMU). To evaluate the utility of each sensor on the recognition phase, the NN was tested with different combinations of sensors and using different numbers of neurons in the hidden layer.

On the other hand, for the second and third performance tests, Kalman and FreeIMU algorithms were used. In these two cases, the number of inputs in the NN were three (pitch, roll and yaw).

Finally, as forth performance test, to evaluate the utility of the frequency spectrum information for animal pattern recognition, the Discrete Fourier Transform of nine sensor data was calculated using a Fast Fourier Transform (FFT). The data used for training and testing the NN were the single-sided spectrum. Thus, the number of NN inputs were also nine and corresponds, for each frequency step, to the nine signals (3-axis of 3 sensors of the IMU).

5 Experimental Results and Analysis

This section presents the experimental results of the classification system with the MLP neural network varying the number of neurons in the hidden layer and the input data between raw and different filtered sensor data.

5.1 Classification System Results Using Sensor Raw Data (Unprocessed)

The classifier system was trained and evaluated using 30,000 samples (10,000 samples of each kind of behavior) obtained from the accelerometer, gyroscope and magnetometer. The samples were randomly divided into three sets: 70% for training, 15% for validation and 15% for testing. Table 1 shows the classification results for the raw sensor data testing set, using different numbers of neurons in the hidden layer.

Table 1 Proportions of true class accuracies using MLP and unprocessed sensor data

Neurons in Hidden Layer	Classes	Sensors used				
		Accelerom.	Gyroscope	Magnet.	Acc. Gyr. Mag.	Acc. Gyr.
10	Trotting	82.4%	78.2%	74.2%	84.0%	82.9%
	Motionless	82.8%	66.0%	43.6%	85.3%	80.7%
	Walking	67.6%	21.0%	53.7%	73.6%	71.4%
	Average	**77.6%**	**55.1%**	**57.2%**	**81.0%**	**78.3%**
20	Trotting	82.9%	80.4%	72.5%	86.0%	83.1%
	Motionless	83.0%	71.0%	46.9%	88.3%	86.8%
	Walking	69.8%	17.5%	57.7%	75.1%	70.9%
	Average	**78.6%**	**56.3%**	**59.0%**	**83.1%**	**80.3%**
30	Trotting	82.0%	77.2%	73.2%	88.4%	85.7%
	Motionless	84.9%	76.3%	49.3%	90.7	85.8%
	Walking	68.6%	22.2%	59.4%	78.0%	74.1%
	Average	**78.5%**	**58.6%**	**60.6%**	**85.7%**	**81.9%**

These results show that the accelerometer is the sensor with better information about the horse movement, while the gyroscope and magnetometer improve the pattern definition. The classifier system has an accuracy of 85.7% with 30 neurons in the hidden layer.

5.2 Classification System Results Using Kalman and FreeIMU Filters

The classifier system was trained and evaluated using 30,000 samples of pitch, roll and yaw. These samples were obtained when applying Kalman and FreeIMU filters to accelerometer, gyroscope and magnetometer raw data in real-time when the microcontroller collar captured this data. The samples were randomly divided into three sets: 70% for training, 15% for validation and 15% for testing. Table 2 shows the classification results for the testing set, using different numbers of neurons in the hidden layer and the applied filter.

Table 2 Proportions of true class accuracies using MLP with filtered sensor data

Neurons in Hidden Layer	Classes	Applied filter	
		Kalman	FreeIMU
10	Trotting	100%	70.5%
	Motionless	93%	51.0%
	Walking	93.9%	57.0%
	Average	**95.6%**	**59.5%**
20	Trotting	99.9%	71.4%
	Motionless	93.6%	57.9%
	Walking	93.7%	66.2%
	Average	**95.7%**	**65.2%**

From Table 2, the recognition performance of our classification system using Kalman filter is 95.6% regardless of the number of neurons in the hidden layer. Therefore, the best performance was obtained with at least 10 neurons in the hidden layer.

5.3 Classification System Results Using FFT Filtered Sensor Data

In order to calculate the FFT of the IMU sensor data, we divided the samples in sets of 256 and 512 samples. The classification system was trained and evaluated using the FFT data calculated for both cases. The samples were randomly divided into the same groups as in 5.2. Table 3 shows the classification results for the testing set, using different numbers of neurons in the hidden layer.

Table 3 Proportions of true class accuracies using MLP and FFT

Neurons in Hidden Layer	**Classes**	**Number of samples for FFT**	
		256	**512**
20	**Trotting**	77.8%	79.9%
	Motionless	59.7%	60.5%
	Walking	47.9%	45.3%
	Average	**61.8%**	**61.9%**
30	**Trotting**	80.5	80.2%
	Motionless	62.7	61.4%
	Walking	51.4	48.9%
	Average	**64.8%**	**63.5%**

These experimental results show that the best accuracy of the recognition system using accelerometer, gyroscope and magnetometer sensors and a MLP neural network was obtained by processing the sensor data with the Kalman fusion algorithm. Two approaches can be considered: to use a Cortex-M4 family microcontroller with a FPU using Kalman filter data as input, sacrificing battery life; or to use a low-power microcontroller without a FPU to save on battery consumption (using raw data as input).

6 Conclusion

In this work, we propose a system to recognize animal behaviors based on artificial-intelligent devices with inertial sensors, based on a NN implementation to classify the possible horse gaits from the collected sensors information. To evaluate the classification system accuracy, four performance tests with different sensor data processing have been performed. The sensors fusion algorithms used were Kalman and FreeIMU. In each test, a MLP NN was created, trained and tested using a varying number of neurons in the hidden layer. The best average accuracy value is 95.6% and it is obtained using 10 neurons in the hidden layer and the Kalman Filter. FreeIMU fusion algorithm and FFT do not bring any improvement to the accuracy of the recognition system. In the case of raw sensor data, the MLP NN needs 30 neurons in the hidden layer to attain 85.7% success. Future work will increase the number of behaviors and animals, and study historic aggrupation of data over time for the classification. The use of FANN open-source library for NN implementation on microcontrollers is under evaluation.

Acknowledgement This work was supported by the excellence project from Andalusia Council MINERVA (P12-TIC-1300).

References

1. Dominguez-Morales, M., et al.: Technical viability study for behavioral monitoring of wildlife animals in Doñana. In: Proc. Int. Conf. Data Commun. Netw. Int. Conf. Opt. Commun. Syst., pp. 98–101 (2011)
2. Painkras, E., et al.: SpiNNaker: A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation. IEEE J. Solid-State Circuits **48**, 1943–1953 (2013). doi:10.1109/JSSC.2013.2259038
3. Doñana National Park: http://whc.unesco.org/en/list/685
4. Tapiador-Morales, et al.: System based on inertial sensors for behavioral monitoring of wildlife. Int. Conf. Comput. Inf. Telecommun. Syst. (2015)
5. MinIMU-9 v2. https://www.pololu.com/product/1268
6. XBEE Pro SB2. http://ftp1.digi.com/support/documentation/90000976_W.pdf
7. IEEE 802.15.4 - IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) (2011)
8. STM32L152RET6. http://www.st.com/web/catalog/mmc/PF259539
9. Madgwick, S.O.H.: An efficient orientation filter for inertial and inertial/magnetic sensor arrays 2010. doi:10.1109/ICORR.2011.5975346
10. Chen, S.Y.: Kalman filter for robot vision: A survey. IEEE Trans. Ind. Electron., 4409–4420 (2012)
11. Haykin, S.: Neural Network: A Comprehensive Foundation, 2nd edn. Prentice Hall (1998)
12. Harris, S.E.: Horse Gaits. Balance and Movement. Howell Book House, New York (1993). ISBN 0-87605-955-8
13. Moller, M.F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks **6**, 525–533 (1993)

Computational Interpretation of Comic Scenes

Miki Ueno

Abstract Understanding intellectual products such as comics and picture books is one of the important topics in the field of artificial intelligence. Hence, stepwise analysis of a comic story, i.e., features of a part of the image, information features, features relating to continuous scene etc., by human and by a combination of several classifiers was pursued. As the first step in this direction, several classifiers for comics are constructed in this study by utilizing a convolutional neural network, and the results of classification by a human annotator and by a computational method are compared.

Keywords Computational model of comics · Comic engineering · Deep learning

1 Introduction

Image recognition by deep learning has seen rapid development in recent years [1]. Previously, I had focused on modeling the pictures in and analysis features of comics [2]. However, numerous challenging tasks must be addressed in order to understand comics and animation, which are composed of pictures and natural languages. The aforesaid tasks are broadly classified into the following two types.

1. **Recognize deformed pictures.** Classification of deformed pictures is difficult [3], while a convolutional neural network (CNN) [4] [5] remarks the potential to achieve the same.
2. **Recognize each scene and the whole story.** Recognize the complex meaning of the story in the following three steps and construct a suitable layered classifier.

M. Ueno(✉)
Information and Media Center, Toyohashi University of Technology, 1-1 Hibarigaoka,
Tempaku-cho, Toyohashi, Aichi 441-8580, Japan
e-mail: ueno@imc.tut.ac.jp

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_42

A part of each scene. The name of the object and facial expressions.

Scene. Social relationships between the characters and the emotions related to each character

Inbetweening scenes. Interpretation of the story by generating intermediate frames between two images for sequential transition of the first frame to the second.

The previously used image recognition method involves machine learning based on appropriate manual vectors utilizing SIFT, HOG and Haar-like features. Then, a part of each scene is recognized, for example face detection, object recognition, etc. On the other hand, CNN, which is one of the methods of deep learning, especially for images, is applied to the image and feature vectors are automatically constructed, so that scene recognition is possible. There are a few studies on the feature vectors for comics [6] because of the rapid advances in deep learning. Thus, it is difficult to design the problems to be solved and datasets to be prepared; adequate discussion of the feature vectors is hence needed.

In this study, as the first step toward understanding comics by using computers, I constructed several classifiers for comics and compared the result classification by hand and by a computational method. The rest of the paper is structured as follows. Section 2 describes the features of comics. Section 3 shows the preliminary experiment carried out to classify images by hand, while Section 4 shows the experiment for the computational method. Section 5 describes the detailed comparison of the two experiments. Finally, section 6 concludes this research and gives brief insights into a future study.

2 Four-Scene Comics

Numerous genres and structures of comics exist across the globe. In this study, four-scene comics, which are structured with four continuous scenes, are considered. The length of four-scene comics is limited so as to ensure clear interpretation of the contents. Figure 1 shows the general structure of each page of the four-scene comics.

Story four-scene comics is one of the styles used in popular Japanese comics. The notable feature of this type of comics is that the characters are common among various short stories, and continuous small stories result in a whole story in the book. Therefore, it is easy to classify two stories by considering the whole series of sequential images. Although the fourth scene of a small story plays an important role to interpret stories, it may be difficult to classify two stories focused on the fourth scene that is selected randomly, because some of characters may be identical between stories; features of characters are similar, and new characters may have appeared in the middle of a story, in the case of works by the same artist.

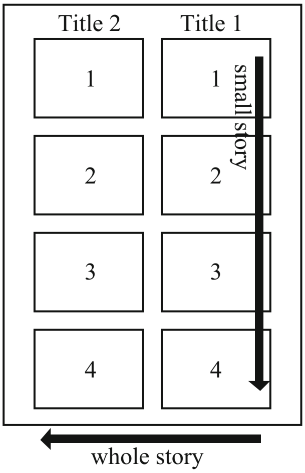
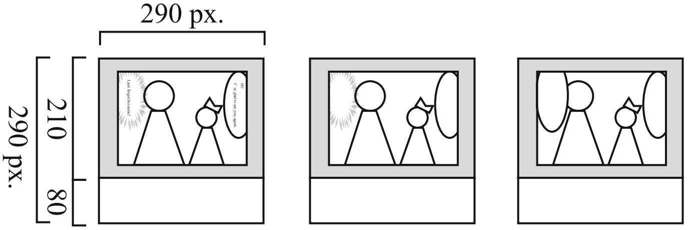


Fig. 1 The structure of four scene comics



(a) Original pictures

(b) Without words

(c) Same shaped balloon

Fig. 2 Each example of three types dataset

3 Preliminary Experiment

In this preliminary experiment, two *story four-scene comics* by same artist are used in order to consider interpretation of comics by human. In order to prevent interpret stories by the title of a small story and inbetweening meanings, only the fourth scene of each small story is given to the annotator.

Dataset:

Two classes, "*Konpeito 1*" [7] and "*Ringo no ki no shitakko de*" [8] by the same artist are used, which have the number of 182 and 178 images respectively.

Table 1 The mean accuracy rate by the human annotator

Size of image	Mean accuracy rate
32 × 32 px.	0.68
64 × 64 px.	0.83
128 × 128 px.	0.93

Six small stories of from the beginning of "*Konpeito 1* were not included in this dataset because the numbers of images in the dataset should be limited to 360 which can be divided by ten. The fourth scene of each small story with 290 × 210 pixels are resized to 290 × 290 pixels added 80 blank pixels to the bottom of the image. Subsequently, they are reduced to 32 × 32 pixels. All the images are in grey scale format.

Comics contain complex information based on both pictures and natural languages. I assumed that picture information was more important for understanding comics. To confirm this, I created three types of datasets by reducing information about natural languages as follows.

- Original pictures
- Without words inside balloons
- Same shaped balloons without words; shapes of all balloons are replaced into same ones manually.

Figure 2 shows each example of these three types of dataset. A preliminary experiment with "same shaped balloons" dataset type is carried out as follows.

1. Give 90% of the dataset with the name of the story to the annotator. The size of the image is 32 × 32 pixels.
2. Annotator labels 10% of the dataset as test data without the name of the story. The size of the image is 32 × 32 pixels.

The accuracy rate is calculated as : the number of the correct label of the book title is divided by the number of test data. As a result, the mean accuracy rate obtained after repeating the experiment three times. It is difficult for human to recognize what objects appear in image with 32 × 32 px. Thus, the accuracy rate of three types of the size in the same dataset are compared. Table 1 shows the mean accuracy rate for three sizes of images in the same datasets. As larger the size, the higher the accuracy rate. I found that characters appearing is the most important information for the human annotator.

4 Experiment

The experiment involves classifying two works by the same artist among three types of datasets by Chainer [9], which is a flexible framework of neural network, written

Table 2 Parameters of neural network

The number of output units	2
Batch size	20
The number of epoch	100
Dropout ratio	0.5
Activation function	ReLU function
Loss function	Softmax cross entropy
Optimizer	Adam. default parameter

Table 3 CNN layer parameters

	CNN1	CNN2
Filter size	5 × 5	5 × 5
Padding size	0	0
Stride	1	1

Table 4 Pooling layer parameters

	POOLING1	POOLING2
Filter size	2 × 2	3 × 3
Padding size	0	0
Stride	2	3

Table 5 The mean accuracy rate by deep learning

Types of datasets	Mean accuracy rate
Original pictures	0.72
Without words	0.83
Same shaped balloon	0.84

in Python. Table 2 shows the parameter of the neural network. Table 3, Table 4 shows the CNN layer parameters, the pooling layer parameters respectively.

The network architecture and the data are described below.

Architecture:
Input - CNN1 - ReLU - MAX POOLING1 - CNN2 - ReLU - MAX POOLING2 - LINEAR1 - ReLU- Dropout [10] - LINEAR2 - Output

Data:
The same dataset is used in Section 3. Each of the three types of datasets is randomly divided into two groups as follows.

Training data. 90 % of the number of the dataset
Test data. 10 % of the number of the dataset

Table 5 shows the result of the mean accuracy rate of 10 times of 100 epochs of learning.

5 Discussion

Comparison of the results presented in Section 3 and 4 indicates that the accuracy in the computational method is higher than that with the human annotator under the condition of the 32 × 32 pixels. Thus, it can be said that the CNN accurately obtains features of scenes of comics.

The human could learn the features of the two types of books focused on a certain character. In the books prepared for the experiment, numerous characters appeared in one scene, while there were a few scenery images and abstract images. Therefore, the human annotator found that the female protagonist of each story appeared frequently in the scenes. However, the size of image is so small that it is difficult to recognize the characters that appear. In addition, the annotator sometimes cannot identify the characters appearing in each book given that the characters are similar due to the same artist. However, the accuracy rate even for 32 × 32 pixels is quite good. It indicates that human classified two books even if they cannot obtain what objects appear. Thus, other information might contributes to classify.

On the other hand, the result might indicate that the computational method learned based on the arrangement of objects such as characters and balloons. The size of the scene in four scene comics is the same, and it is smaller than that in other types of comics. The balloon object is generally located at the end of the frame. Thus, the location of the other objects is limited.

Considering the effect by reducing natural words, the mean accuracy rate for same shaped balloon dataset is the highest, while that of original pictures dataset is the lowest. From this result, it can be said that pictorial information is sufficient to classify these two works by deep learning, because the size of the image is too small to read the character words. Namely, information about the character word is regarded as noise. In the future research, information about pictures and natural languages will be considered in detail.

6 Conclusion

In this study, as the first step toward understanding comics by using computers, three types of datasets are prepared and classifiers are constructed by using a CNN. Comparing the result of classification of two books by the human annotator and by computational method, I found that the CNN is efficient to be applied to grey scaled unphotographic complicated images such as scenes of comics. Furthermore, I indicated the differences of efficient features to interpret comics between humans and computers, and I discussed the effect of reducing information of scenes of comics; i.e., size and words. Features of human annotator recognition and that of computational method may not be the same. However, in order to interpret story of comics, I believe that only simple image recognition is not sufficient but it requires combination of several classifiers for various information of comics referring to human recognition. Thus, I continue analyzing the differences between information focused by human annotator and filters learned by deeplearning.

Future studies would involve the following:

- Consideration of suitable parameters and a layered architecture.
- Detailed analysis for detecting features between human annotators and the computational method.
- Change in the size of the images and multimodal information composed of images and natural languages.

Acknowledgment I wish to thank Mr. Suenaga for preparing the dataset.

References

1. Le, Q.V.: Building high-level features using large scale unsupervised learning. In: Acoustics, Speech and Signal Processing (ICASSP), pp. 8595–8598 (2013)
2. Ueno, M., Mori, N., Matsumoto, K.: 2-Scene comic creating system based on the distribution of picture state transition. In: Advances in Intelligent Systems and Computing, vol. 290, pp. 459–467 (2014)
3. Eitz, M., Hays, J., Alexa, M.: How Do Humans Sketch Objects? ACM Trans. Graph. (Proc. SIGGRAPH) **31**(4), 44:1–44:10 (2012)
4. Fukushima, K., Miyake, S.: Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognition **15**(6), 455–469 (1982)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
6. Tanaka, T., Toyama, F., Miyamichi, J., Shoji, K.: Detection and Classification of Speech Balloons in Comic Images. The journal of the Institute of Image Information and Television Engineers **64**(12), 1933–1939 (2010)
7. Fujino, H.: Konpeito ! 1 (Confetti ! 1). Houbunsha (2007)
8. Fujino, H.: Ringo no ki no shitakko de (Under the apple tree). Houbunsha (2005)
9. Chainer. http://chainer.org/
10. Hinton, G.E., et al.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)

Which Is the Most Appropriate Response? Combining Decision-Support Systems and Conversational Interfaces

David Griol and José Manuel Molina

Abstract In this paper we propose combining decision support systems with spoken dialog systems to facilitate training call-center human operators. In our proposal, the system responses are learned automatically from a dialog corpus by means of a statistical approach based on evolving classifiers. This permits inferring knowledge automatically, that is, the system may infer decisions in complex settings where it is not easy to establish clear hand-crafted rules. Also, the training corpus can be provided from human-human recordings so that the experience of highly qualified human operators can be distilled into the system and offered implicitly to the operators being trained with it. Our proposal has been evaluated with a practical spoken dialog system providing railway information, which follows our proposed approach to integrate a decision support system for the selection of the next system action.

Keywords Decision support systems · Spoken dialog systems · Call-centers · Evolving classifiers · Statistical methodologies

1 Introduction

Decision support systems (DSS) can be defined as computer-based applications that collect, organize and automatically analyze data to facilitate quality decision-making for management, operations and planning [4, 9]. They are especially valuable in situations in which the amount of available information makes very difficult to take decisions for which precision and optimality are very important.

Decision support systems can also aid human cognitive reasoning to integrate several information sources, providing intelligent access to relevant knowledge, and

D. Griol(✉) · J.M. Molina
Computer Science Department, Carlos III University of Madrid,
Avda. de la Universidad, 30, 28911 Leganés, Spain
e-mail: {david.griol,josemanuel.molina}@uc3m.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_43

aiding the process of structuring decisions. They can also support the selection among well-defined alternatives and build decision processes on formal approaches, operations research, statistics, and decision theory. These systems can also employ artificial intelligence methodologies to address heuristically problems that are not tractable by formal techniques [1, 2]. Current applications of DSS include medical diagnosis, business and management, geographic information systems, production control, automation in industrial applications, logistics, education, defense systems, etc[1].

The most important advantages of DSS include to improve performance and effectiveness of the user, allow for faster decision-making, reduce the time taken to solve problems, improve collaboration and communication within groups, provide more evidence in support of a decision, provide different perspectives to a situation, and help automate various types of systems [1, 9].

These important features have promoted the use of decision support systems to train humans to work in contexts where a vast amount of factors must be considered in order to perform tasks successfully. In this paper we propose a combination of decision support systems with spoken dialog systems [5, 7]. These systems can be defined as interactive computer programs that allow the communication with the user in natural language, engaging the user in a dialog that aims to be similar to that between humans [7].

Dialog systems have been elicited negative opinions by some sectors that view them as a replacement of human operators; however, we discuss how they can be used as a natural interface to a decision support system that actually facilitates training call-center human operators. In our proposal, the responses of the system are learned automatically from a dialog corpus by means of a statistical approach based on evolving classifiers. This permits inferring knowledge automatically, that is, the system may infer decisions in complex settings where it is not easy to establish clear hand-crafted rules. Also, the corpus can be obtained from human-human recordings so that the experience of highly qualified human operators can be distilled into the system and offered implicitly to the operators being trained with it. Our proposal has been evaluated with a real dialog system providing railway information, which follows our proposed approach to integrate a decision support system based on a set of fuzzy rules.

2 Combining Decision Support Systems and Spoken Dialog Systems

Figure 1 summarizes the five main tasks usually integrated in a spoken dialog system: Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialog Management (DM), Natural Language Generation (NLG), and Text-To-Speech

[1] A detailed bibliography review about Decision Support Systems can be found at http://www.cif.ulbsibiu.ro/mariusc/bibliographySSD.htm

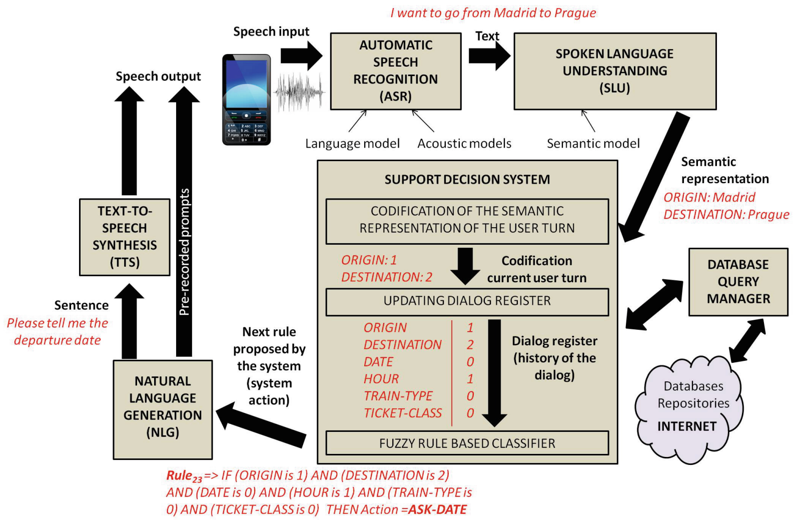

I want to go from Madrid to Prague
Speech input
AUTOMATIC SPEECH RECOGNITION (ASR)
Text
SPOKEN LANGUAGE UNDERSTANDING (SLU)
Speech output
Language model
Acoustic models
Semantic model
Semantic representation
ORIGIN: Madrid
DESTINATION: Prague
SUPPORT DECISION SYSTEM
CODIFICATION OF THE SEMANTIC REPRESENTATION OF THE USER TURN
TEXT-TO-SPEECH SYNTHESIS (TTS)
Pre-recorded prompts
ORIGIN: 1
DESTINATION: 2
Codification current user turn
UPDATING DIALOG REGISTER
DATABASE QUERY MANAGER
Sentence
Please tell me the departure date
Next rule proposed by the system (system action)
ORIGIN 1
DESTINATION 2
DATE 0
HOUR 1
TRAIN-TYPE 0
TICKET-CLASS 0
Dialog register (history of the dialog)
NATURAL LANGUAGE GENERATION (NLG)
Databases Repositories INTERNET
FUZZY RULE BASED CLASSIFIER
Rule23 => IF (ORIGIN is 1) AND (DESTINATION is 2) AND (DATE is 0) AND (HOUR is 1) AND (TRAIN-TYPE is 0) AND (TICKET-CLASS is 0) THEN Action =ASK-DATE

Fig. 1 Modular architecture of a spoken dialog system integrating our proposal to select the next system response

Synthesis (TTS). These tasks are typically implemented in different modules of the system's architecture.

The goal of speech recognition is to obtain the sequence of words uttered by a speaker. Once the dialog system has recognized what the user uttered, it is necessary to understand what he said. Spoken language processing generally involves morphological, lexical, syntactical, semantic, and pragmatical knowledge.

The main objective of the dialog management module is to select the next system action. In our proposal, we integrate a decision support system in order to complete this task. As can be observed, the DSS receives as input the semantic interpretation of a text string recognized by the ASR module. This information also includes confidence scores generated by the ASR and SLU modules.

Our proposal is focused on slot-filling dialog systems, for which dialog systems use a structure comprised of one slot per piece of information that the system can gather from the user. This data structure, which we call *Dialog Register* (DR), keeps the information provided by the user (e.g., slots) throughout the previous history of the dialog. The system can capture several data at once and the information can be provided in any order (more than one slot can be filled per dialog turn and in any order), thus supporting mixed-initiative dialogs.

As described in Figure 1, the DSS must consider the values for the slots provided by the user throughout the previous history of the dialog to decide the next system action. For the DSS to take this decision, we have assumed that the exact values of the attributes are not significant. They are important for accessing databases and for constructing the output sentences of the system. However, the only information necessary to predict the next action by the system is the presence or absence of concepts and attributes. Therefore, the codification we use for each concept and attribute provided by the SLU module is in terms of three values, {0, 1, 2}, according to the following criteria: i) (0): The value for the slot has not been provided; (1) The value is known with a confidence score that is higher than a given threshold; (2): The value of the slot has a confidence score that is lower than the given threshold.

We propose to determine the next system action by means of a classification process, in which the DSS employs the *eClass0* (evolving Classifier) for the definition of the classification function. These classifiers are trained from a labeled corpus of training dialogs, which can be provided by means of human-human dialogs. The classifier is trained by means of the following steps:

1. Classify each new sample in a group represented by a prototype (i.e., data sample that groups several samples which represent a specific system action). To do this, the sample is compared with all the prototypes previously created.
2. Calculate the potential of the new data sample to be a prototype. This value represents a function of the accumulated distance between a sample and all the other $k-1$ samples in the data space. Based on the potential of the new data sample to become a prototype, it could form a new prototype or replace an existing one.
3. Update all the prototypes considering the new data sample. All the existing prototypes are updated considering the new data sample. A new prototype is created if its value is higher than any other existing prototype. Existing prototypes could also be removed.

After training the classifier, a set of fuzzy rules is generated describing the values of the observed features for the classification of each class (i.e., system action):

$$Rule_i = IF(Feature_1\ is\ P_1)\ AND \ldots AND(Feature_n\ is\ P_n)$$
$$THEN\ Class\ =\ c_i$$

where i represents the number of rule; n is the number of input features (observations corresponding to the different slots defined for the semantic representation of the user's utterances); the vector $Feature$ stores the observed features, and the vector P stores the values of the features of one of the prototypes (coded in terms of three possible values, {0, 1, 2}) of the corresponding class $c_i \in$ {set of different classes}. Each class is then associated to a specific system action (response).

The following steps are carried out by the developed dialog managers after each user turn:

1. The values of the different slots provided by the SLU module for the current user turn is coded in terms of the previously described three possible values. Confidence scores also provided by this module are used to determine data reliability (e.g., in Figure 1 the value of the associated confidence scores are used to code “Madrid” as a 1 and “Prague” as a 2).
2. The previous Dialog Register is updated with the new values for the slots determined in the previous step. In the example that Figure 1 shows, the Dialog

Register contains the slots ORIGIN, DESTINATION, DATE, HOUR, TRAIN-TYPE, and TICKET-CLASS. The previous DR containing just a 1 value for the HOUR slot is updated with the ORIGIN and DESTINATION values.

3. The classifier determines the fuzzy rule to be applied (i.e., next system action that will be translated into a sentence in natural language by the NLG module). The cosine distance is used to measure the similarity between the new sample to be classified and the rest of prototypes. For instance, the rule corresponding to the action "Ask for the date" is selected in Figure 1.

3 Practical Application: A Spoken Dialog System Providing Railway Information

We have applied our proposal to develop a mixed-initiative spoken dialog system to provide railway information system using spontaneous speech in Spanish. The different modules of the system were developed using an initial corpus of 900 transcribed dialogs (10.8 hours) provided by human operators and related to real conversations with users asking about timetables, fares, duration of the trips, types of trains, and services [3]. The corpus consists of 6,280 user turns, with an average number of 7.7 words per turn.

The system integrates the CMU Sphinx-II system speech recognition module[2]. As in many other conversational agents, the semantic representation chosen for dialog acts of the SLU module is based on the concept of frame [6]. This way, one or more concepts represent the intention of the utterance, and a sequence of attribute-value pairs contains the information about the values given by the user. For the task, we defined eight concepts and ten attributes. The eight concepts are divided into two groups:

1. *Task-dependent concepts*: they represent the concepts the user can ask for (*Timetables*, *Fares*, *Train-Type*, *Trip-Time*, and *Services*).
2. *Task-independent concepts*: they represent typical interactions in a dialog (*Acceptance*, *Rejection*, and *Not-Understood*).

The attributes are: *Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Class*, *Departure-Hour*, *Arrival-Hour*, *Train-Type*, *Order-Number*, and *Services*.

A total of 51 system responses were defined for the task (classified into confirmations of concepts and attributes, questions to require data from the user, and answers obtained after a query to the database).

Using the previously described codification for the concepts and attributes, when a dialog starts (in the greeting turn) all the values are initialized to "0". The information provided by the users in each dialog turn is employed to update the previous values and obtain the current ones, as Figure 2 shows.

[2] cmusphinx.sourceforge.net

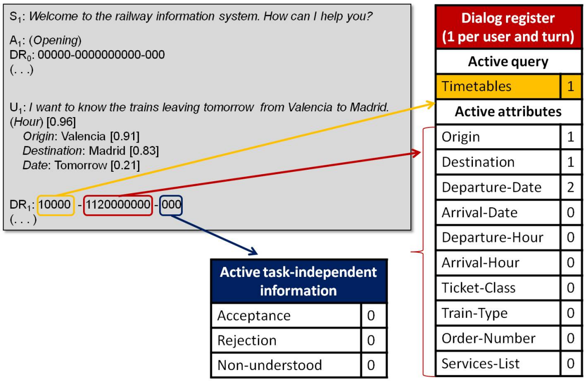


Fig. 2 Excerpt of a dialog with its correspondent representation of the task-dependent and active task-independent information for one of the dialog turns

This figure shows the semantic interpretation and confidence scores (in brackets) for a user's utterance provided by the SLU module. In this case, the confidence score assigned to the attribute *Date* is very low. Thus, a "2" value is added in the corresponding position for this attribute. The concept (*Hour*) and the attribute *Destination* are recognized with a high confidence score, adding a "1" value in the corresponding positions.

The set of features for the support decision system includes the codification of the different concepts and attributes that can be provided by the user and the task-independent information provided in the last user turn (none in this case). A total of 49 rules for the task were obtained with *eClass0*. Figure 3 shows the structure of these rules. Using them, the DSS would select the class $'SystemResponse23'$, which corresponds to a system confirmation of the departure date. This process is repeated to predict the next system response after each user turn.

$FRB - RailwayTask(eClass0):$

$IF\ (Timetables \text{ is } 1)\ AND\ (Fares \text{ is } 0)\ AND \cdots AND\ (Not - Understood \text{ is } 0)$
$THEN\ Class =' Ask - Date'$

$IF\ (Timetables \text{ is } 2)\ AND\ (Fares \text{ is } 0)\ AND \cdots AND\ (Not - Understood \text{ is } 1)$
$THEN\ Class =' Confirm - Timetables'$

$IF\ (Timetables \text{ is } 0)\ AND\ (Fares \text{ is } 1)\ AND \cdots AND\ (Not - Understood \text{ is } 0)$
$THEN\ Class =' Provide - Fares'$

$\ldots$
$IF\ (Timetables \text{ is } 1)\ AND\ (Fares \text{ is } 2)\ AND \cdots AND\ (Not - Understood \text{ is } 1)$
$THEN\ Class =' Close - Dialog'$

Fig. 3 Set of rules for the DSS obtained with *eClass0* for the railway task

4 Evaluation

A 5-fold cross-validation process was used to carry out the evaluation of our proposal. The dialog corpus described in the previous section was randomly split into five subsets of 1,232 samples (20% of the corpus). Our experiment consisted of five trials. Each trial used a different subset taken from the five subsets as the test set, and the remaining 80% of the corpus was used as the training set. A validation subset (20%) was extracted from each training set.

We propose three measures to evaluate the decisions (i.e., system actions) provided by the DSS, which are calculated by comparing the response automatically generated by applying this set for each input in the test partition with regard to the reference answer annotated in the corpus. This way, the evaluation is carried out turn by turn. These three measures are:

- *Exact*: the percentage of responses that are the same that the reference response in the training corpus;
- *Coherence*: the percentage of responses that are coherent with the current state of the dialog although they are not exactly the reference response in the training corpus.
- *Error*: the percentage of responses that would cause the failure of the dialog;

The measure *%exact* is automatically calculated, evaluating whether the response provided by the DSS is the same that the reference answer in the training corpus. On the other hand, the measures *%coherent* and *%error* were manually evaluated by an expert in the task. The expert evaluated whether the response selected by the DSS allows the correct continuation of the dialog for the current situation or whether the answer causes the failure of the dialog (e.g., the dialog system suddenly ends the interaction with the user, a query to the database is generated without the required information, etc).

We have assessed the behavior of our proposal comparing it with different definitions of the classification function used to determine the next system response. In this work, we have used three approaches for the definition of the classification function: a multilayer perceptron (MLP), a multinomial naive Bayes classifier, and finite-state classifiers. We also defined three types of finite-state classifiers: bigram models, trigram models, and Morphic Generator Grammatical Inference (MGGI) models [8].

Table 1 shows the results obtained. As it can be observed, the Fuzzy-rule-based classifier provides satisfactory results in terms of the percentage of correct responses selected (*Matching* and *Coherence* measures) and responses that could cause the failure of the dialog (*Error* measure). With regard the rest of classifiers, the MLP classifier is the one providing the closest results to our proposal. The table also shows that among the finite-state model classifiers, the bigram and trigram classifiers are worse than the MGGI classifier, this is because they cannot capture long-term dependencies. The renaming function defined for the MGGI classifier seems to generate a model with too many states for the size of the training corpus, therefore, this classifier could be underestimated.

Table 1 Results of the evaluation of the different classification functions

Dialog manager	*Exact*	*Coherence*	*Error*
Fuzzy-rule-based (FRB) classifier	76.7%	89.2%	5.6%
MLP classifier	76.8%	88.8%	5.8%
Multinomial classifier	63.4%	76.7%	10.6%
Bigram classifier	28.8%	37.3%	42.2%
Trigram classifier	31.7%	42.1%	44.1%
MGGI classifier	46.6%	67.2%	24.8%

5 Conclusions and Future Work

In this paper, we have presented a proposal to combine decision support systems and spoken dialog systems to facilitate training call-center human operators. In our proposal, the system responses are selected by the DSS using a statistical approach based on evolving classifiers. The training corpus can be provided from human-human recordings so that the experience of highly qualified human operators can be offered implicitly to the operators being trained with the system.

We have described a practical application of our proposal to develop and evaluate a spoken dialog system providing railway information. A codification of the information sources has been proposed to facilitate the correct operation of the *eClass0* classification function. The results of the evaluation shows the correct operation of the proposal with regard other definitions of the classification function. Future work will be oriented to deploy and evaluate our proposal in more practical spoken dialog systems. We also want to evaluate the influence of the main features of the training corpus in the quality of the dialog responses provided by the system.

Acknowledgements This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485).

References

1. Baumeister, J., Striffler, A.: Knowledge-driven systems for episodic decision support. Knowledge-Based Systems **88** (2015)
2. Bhargava, H., Power, D., Sun, D.: Decision support systems and web technologies: A status report. Decision Support Systems **43**(4), 1085–1095 (2007)
3. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication **50**(8–9), 666–682 (2008)
4. Marakas, G.M.: Decision Support Systems. Prentice Hall (2002)
5. McTear, M.F.: Spoken Dialogue Technology: Towards the Conversational User Interface. Springer (2004)
6. Minsky, M.: The psychology of computer vision, chap. In: A Framework for Representing Knowledge, pp. 211–277. McGraw-Hill (1975)
7. Pieraccini, R.: The Voice in the Machine: Building Computers that Understand Speech. MIT Press (2012)
8. Segarra, E., Hurtado, L.: Construction of language models using morfic generator grammatical inference MGGI methodology. In: Proc. Eurospeech, pp. 2695–2698 (1997)
9. Turban, E., Aronson, J., Liang, T.: Decision Support Systems. Pearson (2005)

Betfunding: A Distributed Bounty-Based Crowdfunding Platform over Ethereum

Viktor Jacynycz, Adrian Calvo, Samer Hassan and Antonio A. Sánchez-Ruiz

Abstract Blockchain, the technology behind Bitcoin, is a permissionless distributed database which allows distributed storage and computation over a large network of nodes. This technology has been applied recently to many other fields besides e-currencies (*Bitcoin 2.0* projects). In this paper we present Betfunding: a blockchain-based decentralized crowdfunding platform. On the contrary of regular crowdfunding platforms, our system does not require a central and reliable organization. In Betfutding users bet whether the project will or will not be implemented in a given time frame, increasing the bounty and incentive for potential developers to carry it out.

Keywords Bitcoin · Blockchain · Bounty · Crowdfunding · Cryptocurrencies · Distributed software · Ethereum · P2P · Smart contract

1 Introduction

Crowdfunding is a relatively novel and popular form of collective funding in which the costs related to the development of new projects is supported by small donations or investments made by groups of people interested in those projects. These investors are usually rewarded for their early support with free products and different types of recognition when the project is finished.

Nowadays, crowdfunding is seen as a serious alternative to standard financing using banks in different areas like the development of new gadgets, computer games or even to support political campaigns. This boom was in part brought about by the situation of deepening economic crisis, which limited the issuance of loans by

V. Jacynycz · A. Calvo · S. Hassan · A.A. Sánchez-Ruiz(✉)
Universidad Complutense de Madrid, Madrid, Spain
e-mail: vsjacynycz@ucm.es, adrianclv@gmail.com, {samer,antsanch}@fdi.ucm.es

S. Hassan
Berkman Center for Internet and Society, Harvard University, Cambridge, USA

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_44

banks to new projects, and the mass adoption of the Internet and social networks, facilitating the spread of ideas and alternative financing [5] search.

In the usual approach to crowdfunding, a person or entity ready to carry out a project (the *developer*) looks for economical support from small donors or investors (the *funders*). What we propose in this work is just the opposite, a platform in which the users interested in a project look for developers to carry it out. In order to incentive the developers to work on the project we propose a bet system in which the users bet against the project they want to be done and the developers that are going to work in the project bet in favor of it.

Our main contribution is the development of *Betfunding*, a crowdfunding platform developed on top of Ethereum. Ethereum is a very novel technology that allows the creation of distributed applications that run in an arbitrary large and trust-less network of nodes. Ethereum is based on Blockchain, the technology behind Bitcoin, and facilitates the creation of Bitcoin 2.0 applications, solving most of the technical issues inherent to decentralized platforms, like the one we introduce in this work. The use of this technology provides significant advantages for our application, as it does not require a central organization to manage the projects and bets. In particular, users don't need to trust in a third party with their funds, and the network is resistant to node failures and attacks.

2 Literature Review: Decentralizing Crowdfunding

2.1 Centralization and Its Disruption: The Blockchain

Although there is a large diversity of online platforms, most of them entail centralized control over user data, server infrastructure, and social network interactions. This raises multiple issues like: (1) data is stored in centralized servers (e.g. Google, Facebook, Github) with serious implications for *user's privacy*; (2) the value collectively created by the community is appropriated and *capitalized* by the infrastructure owner; (3) users are enclosed into *walled gardens* and cannot interact beyond the platforms limits (propietary vs. free/libre open source software); (4) users are subject to the *laws* of the country in which the infrastructure operator is physically based (usually USA). These issues have raised concerns from multiple actors, including the European Commission [3, 4, 9], the World Wide Web Consortium W3C [14] or the Free Software Foundation [7, 11].

As a response, different platform architectures emerged like P2P networks. Although this approach was already popular for file-sharing (e.g. eMule, BitTorrent) it was the advent of *Bitcoin* [8] what further pushed it forward. The underlying technology of Bitcoin, the decentralized cryptographic ledger or *blockchain*, provided a paradigm shift for the implementation of distributed systems.

2.2 Bitcoin 2.0 Projects

After Bitcoin's success, a wide diversity of similar digital currencies (cryptocurrencies) appeared, e.g. Litecoin, Ripple or Faircoin[1]. Soon, blockchain-based applications different than currencies emerged, marking the emergence of the *Bitcoin 2.0* projects, i.e., blockchain-based non-currency applications[2]. Thus, e.g. P2P messaging (Bitmessage), domain name management (Namecoin), Dropbox-like cloud storage (Storj), Twitter-like microblogging (Twister). Today, major players like IBM and Samsung are joining the trend[1].

One of the latest *Bitcoin 2.0* projects, and probably the most popular in the blockchain community, is *Ethereum* [2]: a infrastructure with its own programming language for the development of distributed applications over a blockchain. Although other alternatives exist[3], such as Eris *Thelonius* or Counterparty, Ethereum is currently the only running complete solution for *Smart Contracts*.

The Smart Contract protocol proposed by Nick Szabo [12] allows the possibility of self-executing contractual clauses, providing more security than traditional contracts with fewer costs [13]. Ethereum's smart contracts offer the possibility of developing decentralized autonomous applications such as cryptocurrencies, financing platforms, or social networks, to name a few.

2.3 Crowdfunding

Crowdfunding is a form of alternative finance, in which a project is funded by raising monetary contributions from a large number of people. Typically, users or clients assume the cost of the project creation before its implementation. If the project is successful, users are commonly rewarded with perks (e.g. merchandising, premium services, or an instance of the product developed) in exchange of their donations. Crowdfunding popularity has grown exponentially since 2009, with the advent of Kickstarter [6], facilitating the spread of ideas and alternative financing [5]. New forms of crowdfunding have been arising, going further the common Kickstarter model[4]: Goteo for open projects which return to the commons, Gratipay for weekly payments for software developers, or Patreon for creators receiving payments when they produce a piece.

There have been a few experimental attempts to build decentralized crowdfunding on the blockchain[5]. The now defunct Swarm attempted to provide investment-based crowdfunding. Koinify, also defunct, aimed to focus on funding blockchain projects.

[1] Litecoin: http://litecoin.org, Ripple: https://ripple.com/, Faircoin: http://fair-coin.org

[2] Bitmessage: https://bitmessage.org/, Namecoin: https://namecoin.inf, Storj: http://storj.io, Twiter: http://twister.net.co

[3] Thelonius: http://erisindustries.com, Counterparty: http://counterparty.io

[4] Goteo: http://goteo.org, Gratipay: https://gratipay.com, Patreon: http://patreon.com

[5] Swarm: http://swarm.co, Koinify: http://koinify.com

Lighthouse, in Beta stage, is a crowdfunding platform for the Bitcoin cryptocurrency that allows the creation of crowdfunding campaigns for free [10]. Lighthouse can be used in a decentralized way, although its use is much simpler using a centralized server (something that could be avoided if it used Ethereum instead of the Bitcoin blockchain).

3 Betfunding Platform

Betfunding is a distributed crowdfunding platform based on bets. There are 3 different actors involved in the platform:

- *Funders*: users that are interested in the development of a project but do not have the knowledge or the resources to develop it. They can propose new projects or contribute to the bounty of an existing project to make it more attractive.
- *Developers*: users with the knowledge to develop or participate in the development of some of the projects that have been proposed in the platform. Their main motivation is to earn money in exchange of their work.
- *Judges*: they act as a trusted third party designed by the proposer of the project and their mission is to evaluate if the implementation made by the developers fulfills the project specifications. Human judges (see below) always receive a small reward from the project's bounty, regardless the verdict.

Note that, in contrast to traditional crowdfunding platforms, in Betfuding the promoter of the project is not the same team that will develop it. When a funder proposes a new project, he has to provide a contract, i.e., an accurate description of the specifications that will be used to validate the implementation provided by the developers. This description depends on the project but it will always include, among other parameters, the deadline for completion. The funder will also designate a judge that will be in charge of the validation process.

The judge can be either a trusted third human party or a smart contract stored in the blockchain. Note that when we talk about human judges we refer to either individuals or even organization with its own decision mechanisms. Human judges are more versatile since they can take into account parameters from the specifications that are difficult to formalize. Smart contracts, on the other hand, cannot be influenced in any way (*the code is law*) but the project requirements usually have to be much simpler to check.

Each project has 2 different wallets associated in the blockchain: one for the funders and another for the developers. The funders bet against the project they want to be developed transferring money to the funders' wallet. This wallet contains the bounty for the developers if they complete the project and is validated by the judge. The other wallet contains the bets in favor of the project completion, i.e., deposits made by the developers to show their commitment to the project. Once the project is finished there may be two different scenarios:

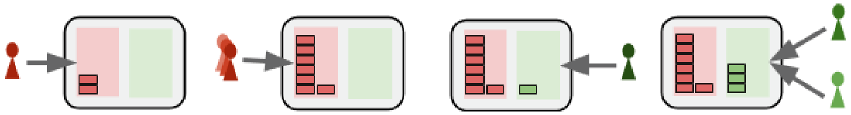


Fig. 1 On the left, bets against the development of the project (funders). On the right side, the contribution by the developers to the deposit.

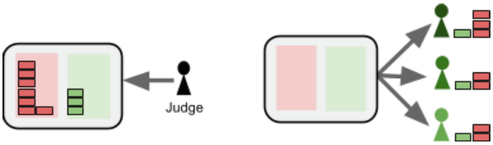


Fig. 2 When the judge verifies the development of the project, the contract gives the money to the developers.

- If the developers do not provide a valid implementation (according to the judge) before the deadline, they will lose their deposits that will be transferred to the funders proportionally to their contributions to the bounty. This way, funders are compensated for the lost time and they will have more money to look for new developers.
- If the project is completed successfully, developers will get their deposits back in addition to a proportional share of the bounty.

The whole process is illustrated in Figures 1 and 2. Let's suppose that Alice is a videogame designer with a fantastic idea about a new mobile game but she does not have the technical knowledge required to implement it. Alice creates a new project in Betfunding, attaches a detailed game design document (GDD) and designates Peter, a renowned game developer, as the independent judge. She also sends some money to the funders' wallet. Then, some other users discover Alice's game proposal and think is very interesting so they bet that the project would never be completed, becoming funders as well and increasing the bounty.

Bob loves programming and he is quite confident to complete Alice's game in time so he talks to some friends and all of them bet in favor of the project and start working on it. Eve is a speculator. She is not interested in games really, but she thinks that this project cannot be done on time so it is a good opportunity to make some money. Eve bets that the game will not be completed. Bob and his friends are even more motivated to finish the game now because Eve has increased the bounty. They work really hard and complete the game, Peter validates it against the GDD, and both the judge and the developers are rewarded for their work.

4 Software Architecture

Betfunding has been implemented using the blockchain-based Ethereum platform described earlier [2]. Thus, following Figure 3, we can observe two main layers: a smart contract, and a standard web interface.

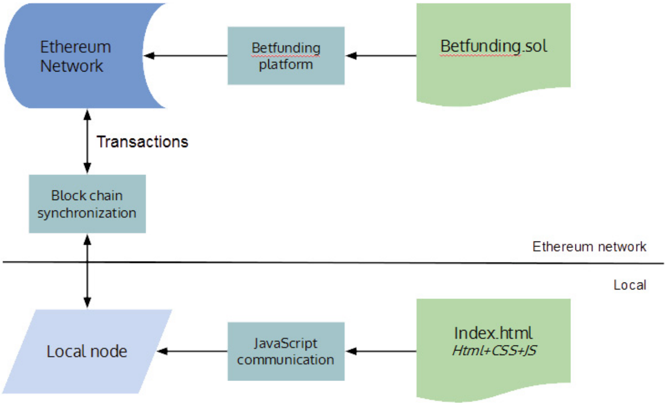


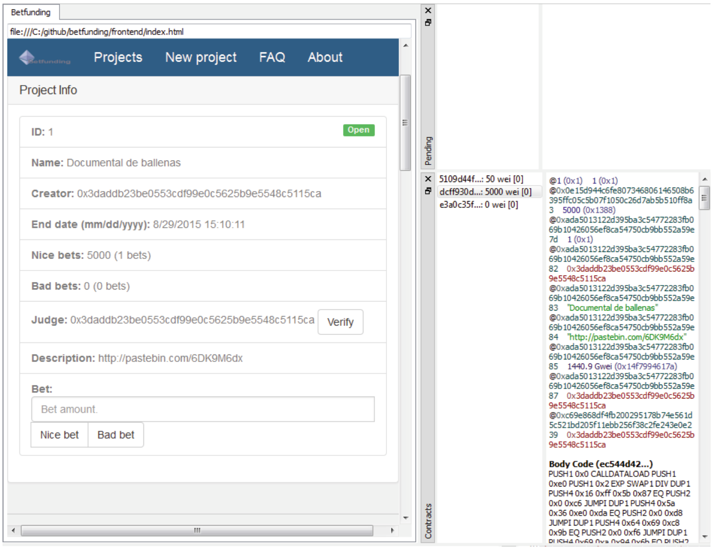
Fig. 3 Structure of the application on the Ethereum network. The application is composed of the contract *Betfunding.sol* and the web interface *Index.html*.

The logic of the application is embedded in the smart contract (*betfunding.sol* in Figure 3), implemented using the *Solidity* programming language and deployed over the Ethereum network. As such, any Ethereum user can run and interact with this smart contract. The contract is fully autonomous in order to minimize dependencies. This implies that, once it is deployed over the blockchain, its creator (or anyone) cannot modify its behaviour or access the stored funds.

The contract is the responsible of storing the information of a project and the records of the bets. The storage and distribution of the money is also done by the contract without external intervention. The money is stored and transferred through the blockchain in the form of *ether*, the native cryptocurrency of Ethereum that works similar to Bitcoin. Thus, users must bet transferring the desired ether amounts to the smart contract, and it will redistribute the ether when the specified project expiration date/time is reached. The evaluation of the success or failure depends on the described judge chosen by each project (and thus Betfunding's only external dependency).

Any project proposers (any Ethereum identity) can transact with the smart contract to create a new project crowdfunding, specifying some simple parameters: name, project description link, expiration date/time and chosen judge. The state of the smart contract will keep track of the bets and store the funds transferred. Projects will remain active until the expiration date/time is reached, when money will be redistributed and the project will be disabled.

In order to facilitate user interaction with the smart contract, a web interface has been developed, using HTML5 and JavaScript (see Fig. 4). In order to use it, web users must have an Ethereum node in their computer. The web interface communicates (using JavaScript functions) with the user's local node, which stores the blockchain with a record of all contracts and transactions. Note such Ethereum node is part of the distributed network of nodes that form the Ethereum network. Therefore, it hosts a copy of the Ethereum blockchain, including a copy of all smart contracts and their state, including Betfunding. The synchronization of any local interaction with



Project Info

ID: 1

Name: Documental de ballenas

Creator: 0x3daddb23be0553cdf99e0c5625b9e5548c5115ca

End date (mm/dd/yyyy): 8/29/2015 15:10:11

Nice bets: 5000 (1 bets)

Bad bets: 0 (0 bets)

Judge: 0x3daddb23be0553cdf99e0c5625b9e5548c5115ca

Description: http://pastebin.com/6DK9M6dx

Bet: Bet amount

```
5109d44f...: 50 wei [0]
dcff930d...: 5000 wei [0]
e3a0c35f...: 0 wei [0]
```

```
@1 (0x1)    1 (0x1)
@0x0e15d944c6fe807346806146508b6
395ffc05c5b07f1050c26d7ab5b510ff8a
3    5000 (0x1388)
@0xada5013122d395ba3c54772283fb0
69b10426056ef8ca54750cb9bb552a59e
7d    1 (0x1)
@0xada5013122d395ba3c54772283fb0
69b10426056ef8ca54750cb9bb552a59e
82    0x3daddb23be0553cdf99e0c5625b
9e5548c5115ca
@0xada5013122d395ba3c54772283fb0
69b10426056ef8ca54750cb9bb552a59e
83    "Documental de ballenas"
@0xada5013122d395ba3c54772283fb0
69b10426056ef8ca54750cb9bb552a59e
84    "http://pastebin.com/6DK9M6dx"
@0xada5013122d395ba3c54772283fb0
69b10426056ef8ca54750cb9bb552a59e
85    1440.9 Gwei (0x14f7994617a)
@0xada5013122d395ba3c54772283fb0
69b10426056ef8ca54750cb9bb552a59e
87    0x3daddb23be0553cdf99e0c5625b
9e5548c5115ca
@0xc69e868df4fb200295178b74e561d
5c521bd205f11ebb256f38c2fe243e0e2
39    0x3daddb23be0553cdf99e0c5625b
9e5548c5115ca

Body Code (ec544d42...)
PUSH1 0x0 CALLDATALOAD PUSH1
0xe0 PUSH1 0x2 EXP SWAP1 DIV DUP1
PUSH4 0x16 0xff 0x5b 0x87 EQ PUSH2
0x0 0xc6 JUMPI DUP1 PUSH4 0x5a
0x36 0xe0 0xda EQ PUSH2 0x0 0xd8
JUMPI DUP1 PUSH4 0x64 0x69 0xc8
0x9b EQ PUSH2 0x0 0xf6 JUMPI DUP1
```

Fig. 4 A Betfunding sample project, shown in the web interface (left) and in the blockchain (right).

Betfunding over the network is performed through the mining process (similarly to Bitcoin). Thus, the web interface is not stored in a server but runs in the user's computer. Moreover, anyone may develop an alternative web interface and use it with the same smart contract.

5 Conclusions

Crowdfunding is currently provided by centralized web platforms with a business model based on commissions, and where the project proposer is the person/entity ready to carry out the project. Using the blockchain-based Ethereum platform, we have implemented a novel decentralized crowdfunding platform, Betfunding. Betfunding is fully decentralized, and no third-party may tamper with its code, appropriate its funds, or charge commissions to its users. Besides, the proposers of a project are not necessarily the ones that will implement it, but the ones willing to fund it. The more funders, the higher the *bounty* and incentive for potential developers to carry out the project.

This application is currently in a functional Beta stage. It is free software, released under a GPL license, and its code can be found in Github[6], ready to be deployed and tested. Betfunding encourages the bottom-up creation of projects through a new and hopefully attractive manner. In particular, it can be exceptionally useful to promote

[6] https://github.com/EthereumUCM/Betfunding

the creation of small projects which have a large community of supporters, and which require a low-medium investment like indie games or specialized apps for mobile devices.

Numerous improvements may be performed on Betfunding. The judge may be certainly improved, for instance depending on trusted projects importing real-world info into the blockchain. Another possibility would be to replace it by a decentralized oracle implemented with SchellingCoin[7], i.e. allowing users to vote on the outcome, rewarding those that are close to the median, complemented with a reputation system. Other improvements may involve the use of Ethereum functionality under development, such as using its storage protocol (Swarm) for storing the project descriptions, or its communication protocol (Whisper) for user coordination. Another improvement would be to use Bitcoin or other cryptocurrencies and not just ether.

We strongly believe blockchain technology is not yet another hype but is here to stay. It has the potential to disrupt a diversity of fields, while empowering the users in a more decentralized Internet. An important way would be to have a trusted way to coordinate efforts and accumulate funds, for projects that, although clearly needed, traditional funding sources have not worked. Beyond seeing Betfunding work, the real question would be: what yet-to-be conceived projects will be proposed and, against all stakes, be funded?

Acknowledgments This work was supported by Spanish Ministry of Economy and Competitiveness under grant TIN2014-55006-R and the Framework programme FP7-ICT-2013-10 of the European Commission through project P2Pvalue (grant no.: 610961). We especially thank Primavera De Filippi (CERSA/CNRS, Berkman Center) for coauthoring the original idea that made this work possible.

References

1. Barker, C.: Is blockchain the key to the Internet of Things? IBM and Samsung think it might just be (2015)
2. Buterin, V.: Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform (2014)
3. Force, E.S.N.T.: Europen Commission Safer Social Networking Principles for the EU. European Commission, Luxembourg (2009)
4. Hogben, G.: Security issues and recommendations for online social networks. ENISA position paper **1**, 1–36 (2007)
5. James, T.G.: Far from the maddening crowd: Does the jobs act provide meaningful redress to small investors for securities fraud in connection with crowdfunding offerings. BCL Rev. **54**, 1767 (2013)
6. Kuppuswamy, V., Bayus, B.L.: Crowdfunding Creative Ideas: The Dynamics of Project Backers in Kickstarter. UNC Kenan-Flagler Research Paper (2013-2015)
7. Moglen, E.: Freedom in the cloud: software freedom, privacy and security for web 2.0 and cloud computing. In: ISOC Meeting, New York Branch, vol. 5 (2010)
8. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)

[7] https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/

9. Schubert, L., Jeffery, K.G., Neidecker-Lutz, B.: The Future of Cloud Computing:Opportunities for European Cloud Computing Beyond 2010:–expert Group Report. European Commission, Information Society and Media (2010)
10. Shubber, K.: The coin rush. New Scientist **225**(3006), 35–39 (2015)
11. Stallman, R.: Who does that server really serve. Boston Review **35**(2) (2010)
12. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)
13. Szabo, N.: The Idea of Smart Contracts. Nick Szabos Papers and Concise Tutorials (1997)
14. Yeung, C.M.A., Liccardi, I., Lu, K., Seneviratne, O., Berners-Lee, T.: Decentralization: the future of online social networking. In: W3C Workshop on the Future of Social Networking Position Papers, vol. 2, pp. 2–7 (2009)

Investigation of the Effects of Imputation Methods for Gene Regulatory Networks Modelling Using Dynamic Bayesian Networks

Sin Yi Lim, Mohd Saberi Mohamad, Lian En Chai, Safaai Deris, Weng Howe Chan, Sigeru Omatu, Juan Manuel Corchado, Muhammad Farhan Sjaugi, Muhammad Mahfuz Zainuddin, Gopinathaan Rajamohan, Zuwairie Ibrahim and Zulkifli Md. Yusof

S.Y. Lim · M.S. Mohamad(✉) · L.E. Chai · W.H. Chan · M.M. Zainuddin · G. Rajamohan
Artificial Intelligence and Bioinformatics Research Group, Faculty of Computing,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
e-mail: {sinyi.5792,mahfuzgear1,gopinathaan24}@gmail.com, saberi@utm.my,
{lechai2,whchan2}@live.utm.my

S. Deris
Faculty of Creative Technology and Heritage, Universiti Malaysia Kelantan,
Locked Bag 01, Bachok 16300 Kota Bharu, Kelantan, Malaysia
e-mail: safaai@umk.edu.my

S. Omatu
Department of Electronics, Information and Communication Engineering,
Osaka Institute of Technology, Osaka 535-8585, Japan
e-mail: omatu@rsh.oit.ac.jp

J.M. Corchado
Biomedical Research Institute of Salamanca/BISITE Research Group,
University of Salamanca, Salamanca, Spain
e-mail: corchado@usal.es

M.F. Sjaugi
Center for Bioinformatics, School of Data Sciences, Perdana University,
43400 Serdang, Selangor, Malaysia
e-mail: farhan@perdanauniversity.edu.my

Z. Ibrahim
Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang,
26600 Pekan, Pahang, Malaysia
e-mail: zuwairie@ump.edu.my

Z.Md. Yusof
Faculty of Manufacturing Engineering, Universiti Malaysia Pahang,
26600 Pekan, Pahang, Malaysia
e-mail: zmdyusof@ump.edu.my

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_45

Abstract DNA microarray technology plays an important role in advancing the analysis of gene expression and gene functions. However, gene expression data often contain missing values, which cause problems as most of the analysis methods of gene expression data require a complete matrix. Several missing value imputation methods have been developed to overcome the problems. In this paper, effects of the missing value imputation methods in modeling of gene regulatory network are investigated. Three missing value imputation methods are used, which are k-Nearest Neighbor (kNN), Iterated Local Least Squares (ILLsimpute), and Fixed Rank Approximation Algorithm (FRAA). Dataset used in this paper is *E. coli*. The results suggest that the performance of each missing value imputation method is influenced by the percentage and distribution of the missing values in the dataset, which subsequently affect the modeling of gene regulatory network using Dynamic Bayesian network.

Keywords Bioinformatics · Artificial intelligence · Gene regulatory network · Missing values · Gene expression · Dynamic Bayesian Network · Gene expression data · Imputation methods

1 Introduction

DNA microarray technology plays an important role in the biological research. This powerful technology has been used to study variety of biological processes and has indirectly increased the speed in analyzing gene expression and gene functions as well. Microarray experiments have produced large amount of gene expression data, which represent the expression level of a particular gene and it has been widely used for scientific study of the gene expression process in organisms. However, gene expression data often contains missing values [1-2], caused by hybridization errors on microarray chips and insufficient resolution. This kind of dataset usually affects the gene expression data analysis as most of the algorithms require a complete matrix of gene expression values. In this paper, several imputation methods have been developed in order to overcome the problems caused by the missing values such as k-nearest neighbor (kNN), Iterated Local Least Squares Imputation (ILLsimpute), and Fixed Rank Approximation Algorithm (FRAA). kNN imputes the missing values by selecting the genes that contain the similar profile of expression to the gene of interest [3]. The ILLsimpute is an imputation method that applies an iterated way using local least squares to increase the accuracy of impute missing values [4]. It consists of two parts which are the estimation of similarity threshold using known expression values in gene expression data and the threshold is applied in Local Least Square imputation method for several iterations in order to obtain the final estimated values for the missing entries. FRAA is a global method proposed by Friedland et al. [5] that finds the optimal values for the missing values of the gene expression matrix using Eigen genes. The main element of the FRAA is the singular value decomposition.

Despite the fact that the effects of imputation methods on the modelling of gene regulatory using Dynamic Bayesian Network (DBN) are not thoroughly investigated. In this paper, three imputation methods have been used to impute missing values in *E. coli* dataset and their effects on the modelled gene regulatory networks are investigated and compared with previous studies. This paper is organized as follows: In section 2, kNN, ILLsimpute, and FRAA imputation methods as well as Dynamic Bayesian Network are briefly explained. Next, section 3 describes the dataset used, experimental setup, and experimental results. Section 4 consists of the conclusion which summarizes this paper and discussion regarding future developments.

2 Methods

2.1 k-Nearest Neighbor for Missing Value Imputation

As mentioned by Troyanskaya et al. (2001) [3], k-Nearest Neighbor imputation method imputes missing values by selecting the genes that contain the similar profile of expression to the gene of interest. Assuming that gene A consists of one missing value in experiment 1, kNN would find K other genes, that consists of a value present in experiment 1 with most similar expression profile to gene A in experiments 2 to N (where N refers to the total number of experiments). A weighted average of values in experiment 1 from K closest genes is then used to estimate missing value of gene A. In the weighted average, each gene is weighted by its similarity of expression to the gene A. According to Oba et al. (2003) [6], kNN imputation method computed first choose K genes that have expression vectors are similar to *yi*. After that, the missing value by the average of the corresponding entries in the chosen K expression vectors is estimated. The similarity measure $s_i\ (y_j)$ between two expression vectors y_i and y_j is determined using the reciprocal of the Euclidian distance calculated over observed components in y_j. Some heuristics is required when there are other missing values in y_j or /and y_i. The measurement below is mentioned by [3]:

$$1si\ (yj) = h\in oi\cap\ oj\ (y_{ih} - y_{jh})2 \qquad (1)$$

$$O_i = \{h|the\ h - th\ component\ of\ y_i is\ observed\}$$

The missing entry yih is measured using the average weighted of the similarity:

$$\widehat{y_{ih}} = \frac{\sum_{j\in I_{Kih}} s_i(y_j)y_{jh}}{\sum_{j\in I_{Kih}} s_i\,(y_j)} \qquad (2)$$

where I_{Kih} is referred as the index set of k-nearest neighbor genes of i-th gene, and if yjh is missing the j-th gene is excluded from I_{Kih}. There are no theoretical criteria for kNN imputation method to select the best k-value, so k-value has to determine empirically. Based on previous research [7], kNN imputation method performs well when dealing with small values of k. However, this method performs poorly when k is too large or small. Hence, it is negatively affected by a badly chosen k.

2.2 Iterated Local Least Squares for Missing Value Imputation

ILLsimpute is a novel iterated method using local least squares to estimate the missing entices [4]. Iterated Local Least Squares method consists of two main steps. In the first step, the known expression values are used to estimate the similarity threshold. In the subsequent step, the final estimated values for missing entries in gene expression matrix is obtained by applying the threshold in LLSimpute method for several iterations.

In ILLsimpute, the number of coherent genes is not fixed and distance threshold is used to define coherent gene. Distance threshold is used to remove the genes which are not similar. Coherent genes within the distance threshold to the target gene are selected. Threshold is set to $mean \ \times ratio$, where mean is the average distance of all genes to the target gene while ratio is a constant to be determined. In the first iteration, Iterated Local Least Squares uses a pre-determined ratio to select the coherent genes for every target gene and then estimate the missing values by running the LLSimpute method. The method iteratively evolves by applying the imputed results obtained from the last iteration to re-select the coherent genes for every target gene to re-estimate the missing values until a pre-specific number of iteration is reached. Liew et al. (2011) [8] states that Iterated Local Least Squares has shown better performance compared to basic LLSimpute, BPCA and KNNimpute because of these modifications.

2.3 Fixed Rank Approximation Algorithm for Missing Value Imputation

Fixed Rank Approximation Algorithm was proposed by Friedland et al. (2006) [5] to estimate the missing entries using Eigen genes. FRAA is a global method which finds the optimal values for the missing values of the gene expression data. Assuming that G represents the gene expression matrix with missing values, the effective rank of G can be estimated by computing the effective rank of the sub matrix, corresponding to all genes with uncorrupted entries. We consider l to be the estimation for the effective rank of the complete gene expression matrix. We let x represent the set for all possible completions of the corrupted gene matrix. In Fixed Rank Approximation Algorithm method, the missing values are refined by searching the minimum to the following optimization problem [9]:

$$min\,\mathrm{x} \in x \sum_{i=l+1}^{m} \sigma_i(X)^2 \ = \ \sum_{i=l+1}^{m} \sigma_i(G^*)^2 \quad ,where\ G^* \in x \tag{3}$$

Based on equation 3, G^* is the completion of the gene expression data which consists of missing values. Basically, Fixed Rank Approximation Algorithm uses iterative procedure to solve the optimization problem that was stated in problem (2.8). Let $G_p \in x$ represents the p^{th} approximation to a solution of problem (2.8). Then, let $A_p := G_p^T G_p$ and search an orthonormal set of eigenvectors for $A_p, v_{p,1}, \ldots, v_{p,m}$. Next, G_{p+1} is a solution to the minimum of a convex non-negative quadratic function that is shown below in equation 2.9 [9].

$$\min_{X \in x} \sum_{q=l+1}^{m} \left(X v_{p,q}\right)^T \left(X v_{p,q}\right) \tag{4}$$

2.4 Dynamic Bayesian Network for Gene Regulatory Network Modelling

Dynamic Bayesian Network is the extension of Bayesian Network which is able to model the gene regulatory networks with cyclic regulations from time series data (Kim et al., 2004). According to Li et al. (2007) [10], Dynamic Bayesian Network is indicated by (B_0, B_1) pair that defines the joint probability distribution over all potential time series of variables $X = \{X_1, X_2, \ldots, X_n\}$ refers discretized-valued random variables in the network. Values of variable X_i are represented as $x_i(1 \leq i \leq n)$. X_i consists of initial state and transition Bayesian Network which is represented by $B_0 = (G_0, \theta_0)$ and $B_1 = (G_1, \theta_1)$respectively. The joint distribution of the variables in X (0) is indicated by B_0, and B_1 refers to the transition probabilities $\Pr\{X(t+1) | X(t)\}$ for all t. When time slice is 0, parents of X_i (0) are specified in the prior network B_0, which mean $Pa\left(X_i (0) \subseteq X(0)\right)$. In slice t+1, the parents of $X_i(t+1)$ are nodes in slices t, $Pa\left(X_i (t+1) \subseteq X(t)\right)$ the connections discovered when it is in between consecutive slices. The joint distribution of random variables over a finite list is shown below:

$$\Pr\{x(0), x(1), \ldots, x(T)\} \tag{5}$$

=1𝑛Pr𝑥𝑖0|(𝑋𝑖(0))× 𝑡=0𝑇−1𝑗=1𝑛Pr𝑥𝑗 𝑡+1𝑝𝑎 (𝑋𝑗 (𝑡+1))}

Besides that, DBN is a two-slice temporal Bayes net (2TBN) and the example of the structure is shown in Figure 1.

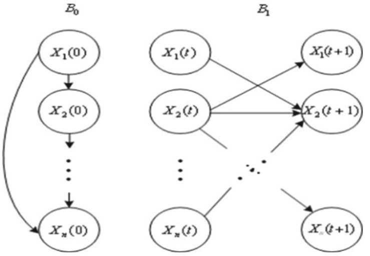


Fig. 1 The basic building block of Dynamic Bayesian Network [8]

3 Experiments

3.1 Dataset and Experimental Setup

In this research, an *Escherichia coli* dataset was used. The dataset was obtained from the previous research work of Spellman et al. (1998) [11]. It consisted of 4 experiments, 8 operons, and 55 time points. The *E. coli* dataset is raw gene expression data and contains many missing values. There is an approximate 11% of missing values in the data set, which are represented by blank spaces.

Three computational approaches were implemented in order to impute all the missing values in this dataset. Imputation methods that were applied in this research were kNN, ILLsimpute, and FRAA imputation methods. These three methods were applied in a MATLAB environment for imputing missing values in the dataset. After the imputation, the subsequent step was the discretization of the gene expression data. Discretization is the process where the noises of the dataset are removed. The gene expression data were discretized into several levels such as -1.0, 0, and +1 which indicates down-regulation, normal regulation, and up-regulation. The three states indicate whether the expression value is lower than or similar to the threshold. In this research, the threshold for down-regulation and up-regulation are determined based on the baseline cut-off of the gene expression values. Discretization can be carried out in a MATLAB environment by setting a certain range of values which enables the conversion of continuous data into discrete data. The threshold that was used for *E. coli* dataset was < 3.7 for down-regulation and > 5.7 for up-regulation. Lastly, GlobalMIT toolbox was used to learn the globally optimal DBN structure from the complete matrix of *E. coli* dataset and model a graph that represented the DBN using Mutual Information Test (MIT). Figure 2 shows the overall steps involved in the experiment setup.

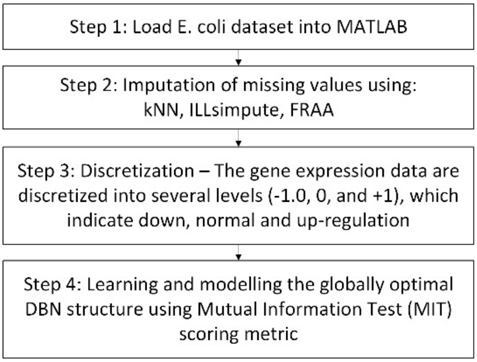

Step 1: Load E. coli dataset into MATLAB
Step 2: Imputation of missing values using: kNN, ILLsimpute, FRAA
Step 3: Discretization – The gene expression data are discretized into several levels (-1.0, 0, and +1), which indicate down, normal and up-regulation
Step 4: Learning and modelling the globally optimal DBN structure using Mutual Information Test (MIT) scoring metric

Fig. 2 Flow of experiment setup.

3.2 Experimental Results and Discussion

Gene regulatory networks modeled using *E. coli* dataset were compared to a well-known publication, Bruno (2003) [12], as the benchmark. This network was involved in the DNA repairing process after DNA damage had occurred. The whole system consisted of approximate 30 genes which were regulated at the transcriptional level. All the genes of the network are normally repressed by a master transcription factor LexA which binds to the promoter region of the genes when there is no DNA damages. RecA acts as a sensor of DNA damage which binds to single-stranded DNA and activates destruction of LexA. The decrease of LexA levels, results in the de-repression of S.O.S. genes.

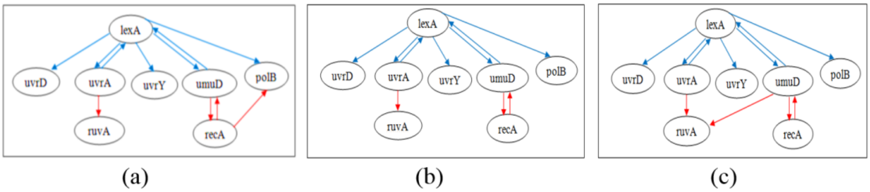


Fig. 3 Gene regulatory network modeled by imputation method for *S.O.S. DNA Repair* data using (a) kNN (b) ILLsimpute (c) FRAA

When the damage has been fixed, the levels of activated RecA decreases, and LexA gathers and represses genes of S.O.S. and cells back to their original state. Figure 3(a), 3(b), and 3(c) shows the *E. coli* gene network modeled in this research with different imputation methods. The directed edges with red color represent the novel potential interactions that were formed in this research. For kNN and FRAA methods, the modeled network consists of 8 nodes and 11 directed edges which are comprised of 4 new potential edges. On the other hand, the modeled network for ILLsimpute method consists of 8 gene nodes and 10 directed edges which are comprised of 3 new potential edges.

Table 1 shows the sensitivity for kNN, ILLsimpute, and FRAA methods have the same percentage, 55.56%. This indicates 5 out of 9 edges that were formed in (Dejori, 2002) also formed in the resultant networks of this research. The specificity and accuracy of kNN and FRAA shows the same percentages which are 80.95% and 73.33 %. Whereas, ILLsimpute has higher specificity and accuracy compare to the kNN and FRAA methods which are 85.71% and 76.67% respectively. This indicates ILLsimpute generates lower less error rates and is a more accurate gene regulatory network compared to kNN and FRAA methods. In this sub-network ILLsimpute methods outperform than kNN and FRAA in term of specificity and accuracy. The computational time required for kNN, ILLsimpute, and FRAA imputation methods to impute the missing values in *E. coli* dataset are discussed and showed in Table 1. The *E. coli* dataset, required a lesser computational time of 1 minute to impute the missing values for the three imputation methods. The results show that the kNN outperforms the other two methods in dataset. This because the kNN method perform well when the number of genes is not to large or small [7].

Table 1 The sensitivity, specificity, and accuracy of *E.*coli gene network modeled by using DBN with different imputation methods

	Sensitivity	Specificity	Accuracy	Computational Time (HH:MM:SS)
kNN	55.56%	80.95%	73.33%	00:00:02
ILLsimpute	55.56%	85.71%	76.67%	00:00:03
FRAA	55.56%	80.95%	73.33%	00:00:13

In this research, effects of the imputation methods on gene regulatory network modeling using DBN with the *E. coli* S.O.S DNA repair network expression dataset are studied. The ILLsimpute outperforms both kNN and FRAA when dealing with the *E. coli* dataset. Specificity and accuracy of ILLsimpute shows significant improvement over the other two methods with p-values of 1.02E-254 and 1.73E-257 respectively based on two-tailed t-test. In term of computational time, kNN requires the least time for missing values imputation. Generally, the three imputation methods require less computation time in small dataset like the *E. coli* dataset to impute the missing values.

4 Conclusion

In this paper, three imputation methods, kNN, ILLsimpute, and FRAA were used to impute the missing values. The effects of the imputation methods in modeling gene regulatory network using DBN were investigated. The results show that the performance of the imputation methods is influenced by the percentage of missing values and distribution of missing values in the dataset, which subsequently affects the modeling of gene regulatory network using DBN. In addition, the resultant networks show that the DBN has capability to discover more potential edges or interactions between genes such as cyclic relationships. In future work, the modeling of gene regulatory using DBN must also consider the effects of imputation methods. For instance, implementation of other imputation methods and application to other gene expression data such as *Arabidopsis thaliana*.

Acknowledgements We would like to thank Universiti Teknologi Malaysia for funding this research through the Research University Grant (Grant number: Q.J130000.2528.12H12). This research also supported by the Malaysian Ministry of Education through the Fundamental Research Grant Schemes (Grant numbers: R.J130000.7828.4F720, RDU140114, and RDU140132).

References

1. Kim, H., Golub, G.H., Park, H.: Missing Value Estimation for DNA Microarray Gene Expression Data: Local Least Squares Imputation. Bioinformatics **21**, 187–198 (2005)
2. Kim, S., Imoto, S., Miyano, S.: Dynamic Bayesian Network and Nonparametric Regression for Nonlinear Modeling of Gene Networks from Time Series Gene Expression Data. Biosystems **75**, 57–65 (2004)

3. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing Value Estimation Methods for DNA Microarrays. Bioinformatics **17**, 520–525 (2001)
4. Cai, Z., Heydari, M., Lin, G.: Microarray missing value imputation by iterated local least squares. In: APBC, pp. 159–168 (2006)
5. Friedland, S., Niknejad, A., Kaveh, M., Zare, H.: An algorithm for missing value estimation for DNA microarray data. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1092–1095. IEEE Press, New York (2006)
6. Oba, S., Sato, M.A., Takemasa, I., Monden, M., Matsubara, K.I., Ishii, S.: A Bayesian Missing Value Estimation Method for Gene Expression Profile Data. Bioinformatics **19**, 2088–2096 (2003)
7. Yoon, D., Lee, E.K., Park, T.: Robust Imputation Method for Missing Values In Microarray Data. BMC Bioinformatics **8**, S6 (2007)
8. Liew, A.W.C., Law, N.F., Yan, H.: Missing Value Imputation for Gene Expression Data: Computational Techniques to Recover Missing Data from Available Information. Briefings in Bioinformatics **12**, 498–513 (2011)
9. Tamura, M.: Missing Value Expectation of Matrix Data by Fixed Rank Approximation Algorithm. Doctoral Dissertation, University of Illinois (2006)
10. Li, P., Zhang, C., Perkins, E.J., Gong, P., Deng, Y.: Comparison of Probabilistic Boolean Network and Dynamic Bayesian Network Approaches for Inferring Gene Regulatory Networks. BMC Bioinformatics **8**, S13 (2007)
11. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botsein, D., Futcher, B.: Comprehensive Identification of Cell Cycle-Regulated Genes of The Yeast Saccharomyces Cerevisiae by Microarray Hybridization. Molecular Biology of The Cell **9**, 3273–3297 (1998)
12. Gene Networks Inference using Dynamic Bayesian Networks: Bruno, E.P., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., d'Alche–Buc, F. Bioinformatics **19**, ii138–ii148 (2003)

Textile Engineering and Case Based Reasoning

J. Bullón Pérez, A. González Arrieta, A. Hernández Encinas and A. Queiruga Dios

Abstract Textile Engineering relies increasingly on the use of computer models that seek to predict the properties and performance of certain textile structures. Those models have been using different computational tools to represent fabrics in a suitable computing environment and also to predict its final properties. Among others, the mathematical models to simulate the behavior of the studied textile structures (yarns, fabrics, kniting and nonwoven). The analysis of textile designs or structures through the Finite Element Method (FEM) has largely facilitated the prediction of their behavior of the textile structure under mechanical loads. For classification problems Artificial Neural Networks (ANNs) have proved to be a very effective tool for a quick and accurate solution. The Case-Based Reasoning (CBR) method is proposed, to complement the results of the those systems where the finite element simulation, mathematical modeling and neural networks can not be applied.

Keywords Textile engineering · Artificial Intelligence · Case Based Reasoning

1 Introduction

Historically the main use of the textile fabrics has been limited mainly to clothing and domestic applications. The technical uses were less important. However in the last

J. Bullón Pérez(✉)
Department of Chemical and Textile Engineering, University of Salamanca, Salamanca, Spain
e-mail: perbu@usal.es

A. González Arrieta
Department of Computer Science and Control, University of Salamanca, Salamanca, Spain
e-mail: angelica@usal.es

A. Hernández Encinas · A. Queiruga Dios
Department of Applied Mathematics, University of Salamanca, Salamanca, Spain
e-mail: {ascen,queirugadios}@usal.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_46

decades the use of the textile structures has started to spread over other sectors like construction, medicine, vehicles, aeronautics, etc. The increased interest in technical applications have improved the fabric design and engineering procedures, given that the final products must be characterized by certain mechanical, electrical, etc., properties. The performance of the fabrics should be predictable from the design phase. The design of a fabric is based on the materials selection as well as on the definition of its structural parameters, so that the requirements of the end use are fulfilled.

These changes in the field of the textile structures application caused a change from the esthetic design to the totally technical design, where the fabric appearance and the particular properties affecting its final performance are taken in account. However, the textile structures are highly complex. A textile fabric consists of yarns; yarns in turn consist of fibres. Thus the mechanical performance of the fabrics is characterized by the structural geometrical complexity and non-linearity, as well as by the non-linearities of the materials themselves. This double non-linear bahaviour of the textile fabrics increases the difficulty in the fabric design and engineering processes. The complex structure and the difficulties introduced by the raw materials do not allow the use of precise analytical models for the technical design of the fabrics.

Fabric engineering activities are increasingly based on computational models that aim the prediction of the properties and the performance of the fabrics under consideration. Different computational tools have been used in order to represent the fabrics in a computational environment and to predict their final properties. Among others, FEM analysis has mainly supported the prediction of the behaviour of the complex textile structures under mechanical loads ([4]). In the case of classification problems ANNs have been proved to be a very efficient tool for a fast and precise solution (see [10]).

2 Overview of the Textile Industry

Currently, the textile industry is dealing with different problems and challenges: product quality and customer service competitiveness, a short-term production and delivery period, the need of a quickly and flexible response due to the high market demand, which is becoming ever more demanding and diversify.

In textiles and clothing industries, a large number of variables are involved. Because of the high degree of variability in raw materials, multistage processing and a lack of precise control of process parameters, the relation between such variables and the product properties is relied on the human (expert) knowledge but it is not possible for human being to remember all the details of the process-related data over the years.

Furthermore, product performance failures are unacceptable because there is a clear risk of loss of functionality, and as a consequence, sometimes there is danger for human life. Up to now, the typical development practice for multifunctional textiles is to adjust the processing parameters and modifying preliminary products by trial and error, producing samples until the desired quality. Design and functionality parameters can be safely achieved during production. This procedure accumulates problems along the manufacturing value chain. For instance, a deviation in the specifications of a yarn can produce a fabric that does not match the required performance or functionality.

There are several research papers focus on the efficiency of the tools of the Computer Aided Design (CAD) in the textile industry (see [1],[3], and [6], [8] for example). This efficiency can be determined by:

- The decrease of the production time of a same collection regarding the traditional method.
- The reduction of physical prototype tests.
- The reliability (the capacity of decision grew).
- The flexibility (the capacity rectification becomes agile).
- The creativity: Accentuating the ability to manifest the creativity of designers, related to the concept of unevenness ([9]), deviation of ideas in the process of creation.
- The organization process: Organization reinforcements of the creating products process that allow the transmission or correct information in each state of creation (suitable formats and presentations). Quality control from the beginning of the whole process.

More necessary and important than simulations is to connect this simulation with the production systems because:

- It is considered that the computer simulation systems do not allow to make reliable decisions about what would be the final product.
- It is still necessary for sale the creation of a sample, since simulations are not accepted in the current culture of the market.
- The application of CAD tools does not prevent knitting prototypes, but it reduces their number, and discard those that are not satisfactory.
- The simulation of fabrics, raw material, colorful, structures are not precise enough.

The latest trends in database development and the use of a CAD system are based on the advice of a system of Artificial Intelligence (AI). And more specifically in the use of CBR ([11]) is concerned with the analysis or solution of problems based on previous cases.

Compared to other systems, the CBR provides a more natural way to continue the work. It is based on the use of the experience to understand and to solve new problems, interacting with the user (expert) to determine their objective. The feedback from the expert will help to advise and make proposals and to satisfy the new demands.

But due to the complex structure of the textile and the difficulties commodity added, it is not possible to use accurate models for design analysis. Therefore these activities textile engineering increasingly rely on the use of computational tools, whose approximate models predicting the properties and behavior of textile structures.

3 CBR and the Textile Industry

Humans solve many problems by reasoning with previous cases: Lawyers use in their arguments the verdict of previous cases. Doctors look for clusters of symptoms that identify patients with a set of previous cases. Engineers make many of their ideas from previous solutions already built successfully. Experts programmers reuse more or less abstract schemes of previously obtained solutions.

Therefore, the idea is to use some method of AI that allows the approach of these parameters of the process, learn from the past of such processes. Creating a database that will grow over time as more products are manufactured. Each industry produces different products and each product can have different associated processes.

The CBR is a methodology that uses different technologies for building expert systems, and in the acquisition of knowledge:

- Can propose solutions in domains that are not fully understood.
- It is possible to evaluate solutions when there is no algorithmic method.
- It is easier to acquire new cases, to discover new rules and generalizations.
- It is faster to reuse a solution to get the solution from scratch.
- The cases help a reasoner to concentrate on the important aspects of a problem, to identify the defining characteristics
- Cases can also provide "negative information", warning of possible failures (exceptions).
- Maintaining the knowledge base: users can add new cases without expert help.

CBR solves a problem by adapting solutions given previously to similar problems ([15]). The CBR memory stores a number of problems with their solutions. The solution of a new problem is obtained recovering cases (or problems) stored in the memory like the CBR, as these problems are studied together with their solutions. Therefore a case encompasses a given problem and the solution to this problem.

In textiles the problem starts when a user wants to create a new textile product and he/she asks for a suitable solution satisfying as much as possible his/her requirements on the process parameters. Hence, the idea is to use some AI method to approximate these process parameters, learning from the past executions of this kind of processes.

All in all, an AI method which supports the following properties is needed:

- Incremental: The algorithm should adapt itself along time with the new data.
- Non-dependent of any parameter, since each query can contain different parameters and the relevance of these parameters can change depending on the user requirements.
- Being able to predict more than one parameter.

These features point to a flexible methodology like CBR. The fact that CBR is a lazy learning approach allows that the algorithm will not be influenced by any parameter, and can be adapted to any new situation giving more importance to some parameters than others. Besides, the implementation of CBR can be enough open to allow the prediction of more than one parameter ([16]).

Moreover, textile end users are not interested in a general model which generalize all the data, but a particular solution proposed for the particular query that they are requesting.

Another point to take in to account is that our system must support more than one attribute belonging to the solved problem, and this classification between description and solution can vary on demand. In CBR literature, some hybrid solutions have been found to face up with the reuse step, but most of them are oriented to solve only one solution attribute.

Most of the proposals use techniques from supervised learning and have only one attribute to predict ([7]). For example, a system to predict oceanographic temperatures ([5]). This system retrieves the most similar cases and retrains a radial basis network with them to create a new solution.

Another CBR approach that handles more than one attribute as solution is the case completion CBR ([2]), but the attributes are predefined to belong to the description or the solution part. It handles more than one attribute to solve, but in this case, they solve them step by step. As they are dealing with domains that have predefined and well-known attribute dependencies, the order in the steps is derived from those dependencies.

In CBR recommended literature, the compromise-driven retrieval ([12],[13]) can be found. This concept allows to extend the query specification including rules for numerical attributes: *Less-is-Better and More-is-Better.* A typical CBR system consists of four sequential steps that are invoked whenever it is necessary to solve a problem ([11]). Figure 1 shows the CBR applied to Textile Industry. A textile process involves transforming the fibers by mechanical action into a final product, that is, equipment and materials: raw material inlet and outlet tissue or product. These elements can be described by parameters or attributes. Consequently, the process can be described as an attribute-list-value.

The development of a rapid configuration system for textile production machinery based on the physical behaviour simulation of precision textile structures system ([14]) was born to be a tool for helping to adjust the parameters of a process. This system starts when a user wants to create a new product. The user should specify some of the parameters that he/she wants to obtain for the required product. These parameters can belong to the raw material, to any machine in the process or to the final product. So, with an AI method it is possible to approximate the rest of the parameters learning from the past executions of this kind of process.

In this system for textile production machinery more than one parameter must be predicted and depending on the query, these parameters are not always the same. Moreover, these could change any time that there is a new query. Besides, the database will grow as more process executions are finished.

In addition, some of these parameters have extra information that can be used to know its value in a determinate situation like the parameters that can be defined by a predefined empirical formula. So these parameters are not going to be predicted, because it is more reliable the exact value. Another kind of information that guides the predicting process is the knowledge of the relations among parameters.

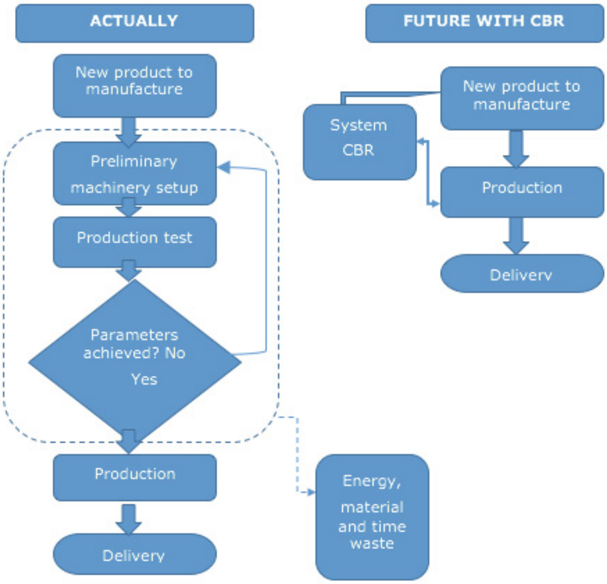

ACTUALLY
New product to manufacture
Preliminary machinery setup
Production test
Parameters achieved? No
Yes
Production
Delivery
Energy, material and time waste
FUTURE WITH CBR
System CBR
New product to manufacture
Production
Deliverv

Fig. 1 CBR applied to Textile Industry

The solution adopted in this study was the Case Based Reasoning. CBR is a problem solving paradigm that uses the human reasoning model as a base: humans use past experiences to solve new situations or problems.

This solving paradigm is based in collecting a lot of relevant cases that are the past experiences of the system. The CBR consists on:

- To gather the problem description by the user.
- To measure the similarity of the current problem to previous problems stored in a case base (or memory) with their known solutions, retrieving one or more similar cases.
- To adapt (reuse) the solution of one or more of the retrieved cases, possibly after adapting it to account for differences in problem descriptions.
- To evaluate (revise) by the expert the solution proposed by the system.
- The problem description and its new solution can be retained (stored) as a new case, and the system has learnt to solve a new problem.

In this Project ([14]), the last two points has been discarded, due to the fact that the project cannot rely on this (as it cannot be assure) it will be done or at least, done with a minimum quality (see Figure 2).

The CBR software has 4 phases: Retrieve, Reuse, Revise Retain.

The Retrieve phase involves findind the most similar case/s to a given case. This task starts with a (partial) problem description and ends when best matching previous case has been found. It is usual to divide this task in two parts. The first one tries to

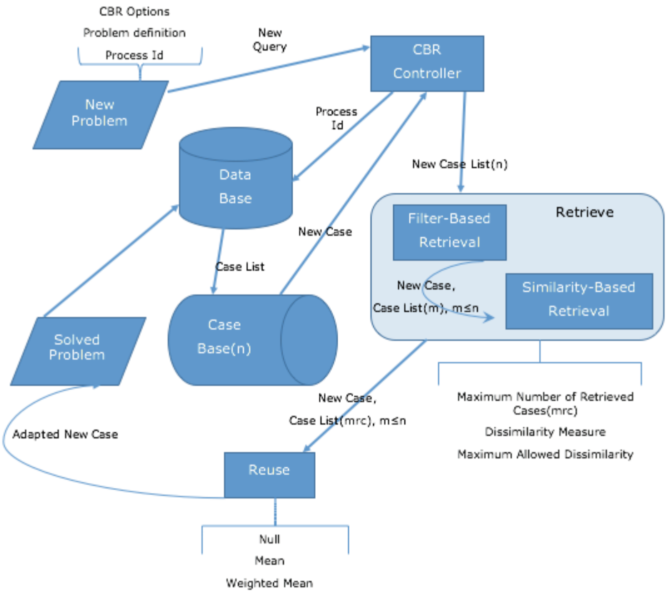


Fig. 2 The life cycle of CBR applied to Textile Industry

select similar cases and the second one chooses the best matches. The similarity is a function that gives a scalar distance between two arguments (case element and case problem). This value is also calculated also with a weight given for each parameter.

The Reuse phase is the most complex tasks in the CBR cycle. It is a phase based on the principle: *Similar problems have similar solutions*.

The Revise and the Retain phases have been discarded, as previously mentioned, due to the fact that the system cannot rely whether the expert has revised and evaluated the solution provided by the system.

4 Conclusions

The impact in the manufacturing processes in the textile Small and Medium-Sized Enterprises with no design software applications, and also the textile companies with CAD software applications to support the design processes, is:

- Detection of defects or other problems that may appear during the posterior manufacturing process.
- Integration of the simulation results and parameters into the textile machinery for the rapid configuration of the machine as well as the precision manufacturing of the virtual textile structures developed using this software.

- The conventional visualization and design tools don't bring added value since they are of little help to design multifunctional technical textiles. The company that follows this pattern is only able to offer conventional textile products. The capacity to design any kind of multifunctional textiles has a great impact in the capacity to offer added-value from the company and its products.
- The improvement in the speed and flexible development of new technical textile products. The simplification of the production process to obtain such products increases the rate of new products and at the same time successfully finding a niche market, and covering new society necessities, and giving added-value opportunities to the textile companies

The possibility of monitoring the properties of these products in the computerized model of the textile will accelerate enormously the configuration of the production machines. At the same time, the companies are capable of generating added value with the design and manufacturing of technical textiles, and they will be capable of moving towards the innovation-driven type of company.

Acknowledgments We would like to thank Ana Bullón Carbajo for her language advice.

References

1. Bingham, G.A., Hague, R.: Efficient three dimensional modelling of additive manufactured textiles. Rapid Prototyping Journal **19**(4), 269–281 (2013)
2. Burkhard, H.D.: Case completion and similarity in case-based reasoning. Comput. Sci. Inf. Sys. **1**(2), 27–55 (2004)
3. Chattopadhyay, R., Guha, A.: Artificial Neural Networks: Applications to Textiles. Textile Progress **35**(1), 1–46 (2004)
4. Chen, X.: Modelling and Predicting Textile Behaviour. Woodhead publishing Series in Textiles, Elsevier (2009)
5. Corchado, J., Lees, B.: A hybrid case-based model for forecasting. Appl. Artif. Intell. **15**(2), 105–127 (2001)
6. Dwivedi, A.: Role of Computer and Automation in Design and Manufacturing for Mechanical and Textile Industries: CAD/CAM. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2278–3075 (2013)
7. Fernández-Riverola, F., Corchado, J.M.: Sistemas híbridos neuro-simbólicos: Una revision. Inteligencia Artificial, Rev. Iberoam. de I. A. **4**(11), 12–26 (2000)
8. Fontana, M., Rizzi, C., Cugini, U.: 3D virtual apparel design for industrial applications. Computer-Aided Design **37**(6), 609–622 (2005)
9. Green, T.R.G.: Cognitive Dimensions of Notations. People and Computers V. Cambridge University Press, Cambridge (1989)
10. Guruprasad, R., Behera, B.K.: Soft Computing in Textiles. Ind. J. of Fibre and Text. Res. **35**, 75–84 (2010)
11. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, San Mateo (1993)
12. McSherry, D.: Similarity and compromise. In: Lect. Notes Comput. Sci., vol. 2689, pp. 1067–1067. Springer, Heidelberg (2003)
13. McSherry, D.: Completeness criteria for retrieval in recommender systems. In: Lect. Notes Comput. Sci., vol. 4106, pp. 9–29. Springer, Heidelberg (2006)

14. MODSIMtex: Development of a rapid configuration system for textile production machinery based on the physical behaviour simulation of precision textile structures, Project reference: 214181 Funded under: FP7-NMP. modsimtex.eu
15. Riesbeck, C.K., Schank, R.C.: Inside Case-Based Reasoning. Lawrence Erlbaum Ass, Hillsdale (1989)
16. Sevilla, B., Sánchez, M.: Case-based reasoning applied to textile industry processes. In: Lect. Notes Comput. Sci., vol. 7466, pp. 428–442 (2012)

Distributed Fair Rate Congestion Control for Vehicular Networks

Jamal Toutouh and Enrique Alba

Abstract Vehicular ad hoc networks (VANETs) are self-organizing communication networks, which principally consist of vehicles that broadcast beacons with relevant real time traffic information. These continuous exchanges of information allow the development of applications that drastically improve the road traffic safety and efficiency. Such services suffer from network congestion problems when the road traffic density increases. This may cause VANET malfunction, and thus, the increase of hazardous road situations. In this study, we present a family of fully distributed intelligent light-weight congestion control algorithms (executed by each node), i.e., the Distributed Intelligent Fair Rate Adaptation (DIFRA) family. These methods accurately estimate the channel load in a distributed manner and dynamically adapt the beacon rate of each node. Experimental analyses show the effectiveness of DIFRA methods in increasing the amount of data exchanged between the vehicles and the balance in the channel usage, while avoiding network congestion.

Keywords VANET · Intelligent · Broadcasting · Congestion control

1 Introduction

Vehicular traffic is becoming a major concern in modern cities. Several problems related to road safety, traffic efficiency, environment, etc. can be efficiently solved by applying innovative *intelligent transport systems* (ITS). The main idea behind these systems consist in sharing information about the traffic conditions with road users and authorities. A better informed vehicle/driver can take better driving decisions, positively influencing the global traffic (safety and efficiency).

J. Toutouh(✉) · E. Alba
Dept. de Lenguajes y Ciencias de la Computación, University of Málaga, Málaga, Spain
e-mail: {jamal,eat}@lcc.uma.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_47

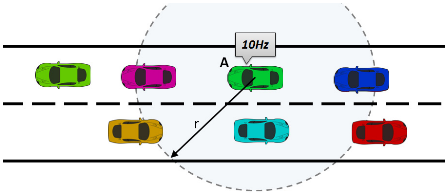


Fig. 1 Vehicle A is performing a beaconing operation at a frequency of 10 Hz.

Vehicular ad hoc networks (VANETs) emerge as a promising technology to define ITS. VANETs are distributed and self-organizing communication networks, which principally consist of vehicles equipped with on-board units that broadcast beacons by using *direct short range communications* (DSRC) [1].

Most VANET applications principally rely on beaconing (continuously broadcasting beacons) on the neighborhood defined by the communication range of the nodes (r). Beacons include vehicle kinematics (e.g., position and acceleration) and other relevant information for several applications or services. Fig. 1 shows an example of a vehicle broadcasting beacons.

One of the most challenging issues in the deployment of VANETs is the network congestion, that aggravates as the scale of the system grows. This is mainly because of the critical increase of the periodic beacons that cannot be finally transmitted through the limited channel. Network congestion increases the loss of packets and the communication delays. This may lead to excessive information inaccuracy and to an eventual failure of VANET applications.

Cooperative Collision Warning (CCV) provides an active VANET safety mechanism based on beaconing [4]. CCV is implemented by broadcasting static and dynamic vehicle parameters to the neighborhood. This information is used to compute the relative safety distance between neighbor vehicles. However, if a network congestion situation appeared, the CVV information could not be received on time and the probability of hazardous situations could increase.

Several strategies have been proposed to address such congestion problems in VANETs, keeping the communication capabilities of the nodes over a given QoS threshold. Most of them can be included in the following basic schemes [1, 6, 8]: i) adapting the transmission range of used communication channels, ii) adjusting the data rate generation of applications and services, and iii) hybrid methods by combining the two previous schemes.

In this study, we have defined *fair beacon rate (FBR) optimization problem* in order to adjust the current beacon rate, which considers two main goals: i) maintaining the VANET load under a given threshold to avoid network congestion and ii) balancing beacon rates (allowing the nodes in a given area exchange beacons with similar rates). This problem has been addressed by using the ***Distributed Intelligent Fair Rate Adaptation (DIFRA)*** family of methods, which are a set of intelligent methods based on light-weight computations. According to the experimental results, DIFRA exhibits a reliable, non-complex, and efficient congestion control.

The remainder of this paper is organized as follows. Section 2 formalizes the FBR optimization problem. Section 3 describes the DIFRA methods devised here. Section 4 presents and discusses the experimental analysis. Finally, Section 5 summarizes the main findings of our study.

2 Fair Beacon Rate (FBR) Optimization Problem

This section defines the FBR optimization problem. The information required to adapt the beacon rates is the current network load (channel occupancy). The evaluation of the channel occupancy in VANETs can be carried out by monitoring the length of the queues (in the MAC sublayer of IEEE 802.11p) in a given window of time [3]. The FBR problem then considers:

- The *maximum allowed channel occupancy* ($MaxQ$). $MaxQ$ in practice represents the maximum value of queues length, i.e., the number of beacons waiting in the reception queue that does not pose problems like congestion.
- A *threshold limit ratio* $\alpha \in [0, 1]$ over the $MaxQ$. If the queue length exceeds a given *effective capacity of the channel* ω, which is defined as $\omega = \alpha \times MaxQ$, the protocol considers that the current network load could lead to a congestion situation, provoking an unpredictable performance of the network [2].
- A *set of allowed beacon rate values* (integers) BR=$\{br^1, br^2, ..., br^k\}$ that contains all the possible beacon rate values ($br^i \in [br^{MIN}, br^{MAX}]$) that can be selected by the all nodes according to the application QoS requirements.
- Given an integer v that represents each vehicle that belongs the VANET, the $NN(v)$ function returns the set that contains all the 1-hop neighbor nodes.

The problem consists in finding $br_v \in BR$ for each node v to optimize two objectives: i) maximizing the number of the beacons traveling through the shared medium in terms of the ratio of the channel occupancy ($\eta(v)$ in Equation 1); and ii) minimizing the difference between the effective beacon rates in the neighborhood of v ($\sigma(v)$ in Equation 2), i.e, maximizing the balance/fairness.

$$MAX\ \eta(v) = MAX\ \frac{\left(\sum_{j}^{j \in NN(v)} br_j\right) + br_v}{MaxQ} \qquad \eta(v) \in [0, +\infty) \tag{1}$$

$$MIN\ \sigma(v) = MIN \frac{\left(\sum_{j}^{j \in NN(v)} (br_j - \overline{br_v})^2\right) + (br_v - \overline{br_v})^2}{|NN(v)|} \qquad \sigma(v) \in [0, +\infty) \tag{2}$$

The $\overline{br_v}$ value represents the average beacon data rates of all the vehicles in the neighborhood of v. As a constraint, the selected beacon rates computed by the algorithms br_v should not generate network congestion ($\eta(v) \leq \omega$).

Therefore, the fair beacon rate optimization problem consists in finding for each VANET node v its current br_v that optimizes both objectives. The computations are carried out in a distributed manner (by each node itself) taking into account $MaxQ$, α, and the beacon rates of all its neighbors $(NN(v))$.

3 Distributed Intelligent Greedy Dynamic Broadcasting

In this study, we define the DIFRA family of algorithms to dynamically compute accurate beacon rates to address the FBR optimization problem. These algorithms are fully distributed and executed periodically by each node of the VANET at a given frequency according to a given window time. DIFRA is composed by two different algorithms: the ***Self DIFRA***, in which the nodes compute FBR using isolated information from their own network monitoring, and the ***Swarm DIFRA***, which combines isolated information with information received from the neighbor nodes.

The DIFRA algorithms have three different software components:

- *Self Queue Monitoring Component (SQMC)*: It evaluates the number of neighbor nodes. SQMC is used by Self and Swarm DIFRA algorithms.
- *Swarm Information Exchange Component (SIEC)*: Swarm DIFRA utilizes it to evaluate the information encoded in the received beacons.
- *Beacon Rate Adaptation Component (BRAC)*: It adapts the beacon rate according to the current network status.

The following subsections detail the operation of DIFRA algorithms.

3.1 Self DIFRA Method

The main idea of Self DIFRA consists in sharing proportionally the channel by dividing its maximum effective capacity into all the nodes in the neighborhood (including the own node). Therefore, when VANETs apply Self DIFRA for network congestion, each node invokes its SQMC to compute $|NN(v)|$, i.e., the number of neighbor nodes (see Fig. 2). Then, BRAC computes the *tentative beacon rate* (tbr_v) by dividing the effective capacity of the channel (ω) by the number of the neighbor nodes plus one (see Equation 3). If tbr_v is higher than br^{MAX}, then br_v is equal to br^{MAX}. Otherwise, br_v is equal to tbr_v.

$$
\begin{aligned}
tbr_v &= \left\lfloor \frac{\omega}{|NN(v)|+1} \right\rfloor \\
br_v &= \begin{cases} tbr_v & \text{if } tbr_v \le br^{MAX} \\ br^{MAX} & \text{if } tbr_v > br^{MAX} \end{cases}
\end{aligned}
\tag{3}
$$

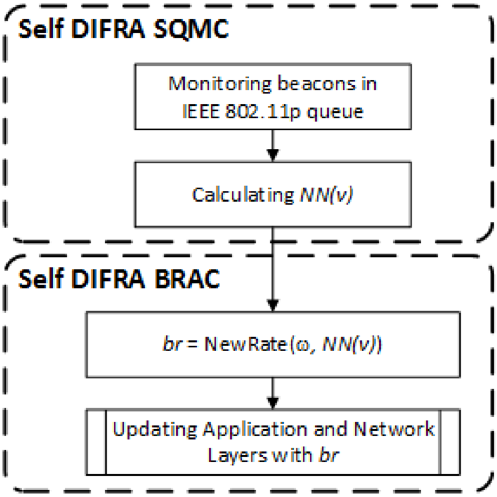

Self DIFRA SQMC
Monitoring beacons in IEEE 802.11p queue
Calculating NN(v)
Self DIFRA BRAC
br = NewRate(ω, NN(v))
Updating Application and Network Layers with br

Fig. 2 Complete flowchart of Self DIFRA algorithm.

3.2 Swarm DIFRA Method

As to Self DIFRA, this method is based in the idea that the channel should be proportionally divided according to the neighborhood size. However, this proposal combines self measured (monitored) information with shared congestion control data from the neighbor nodes. This control information is encoded as an integer value in the beacons to be broadcasted. Summarizing the process, each node in the VANET computes a *desirable beacon rate (DBR)* according to the neighborhood size (as shown in Equation 3) and exchanges this information to its neighbor nodes in order to request them to change their beacon rates. The DBR received through the exchanged beacons is stored in a temporal buffer (*BRBuffer*) in order to utilize it in the near future beacon rate computations.

The BRBuffer of each node is a vector of natural values with $|BR|$ components, $[x_1\ x_2\ ...\ x_k]$. Each one of the k components stores the number of petitions received by the node to change its beacon rate to i beacons per second. For example, if BRBuffer=[0 0 10 25 0 0 0 0 0 1], then it means that the node has received 10 requests to change the beacon rate to 3 Hz ($x_3 = 10$), 25 to change to 4 Hz ($x_4 = 25$), and 1 to change to 10 Hz ($x_{10} = 1$).

Fig. 3 summarizes the Swarm DIFRA operation. SQMC and SIEC are executed in parallel and BRAC is run periodically with a given frequency determined by a given *timeout* timer.

The Swarm DIFRA SQMC component behaves similarly to Self DIFRA. The difference is that the swarm method does not update the current beacon rate, in contrast, it increases the BRBuffer component of the DBR that is computed as br_v in Equation 3, i.e., $DRB = br_v$. Thus, this component updates the BRBuffer by increasing the DBR-th component ($x_{DBR} = x_{DBR} + 1$) and includes the DBR

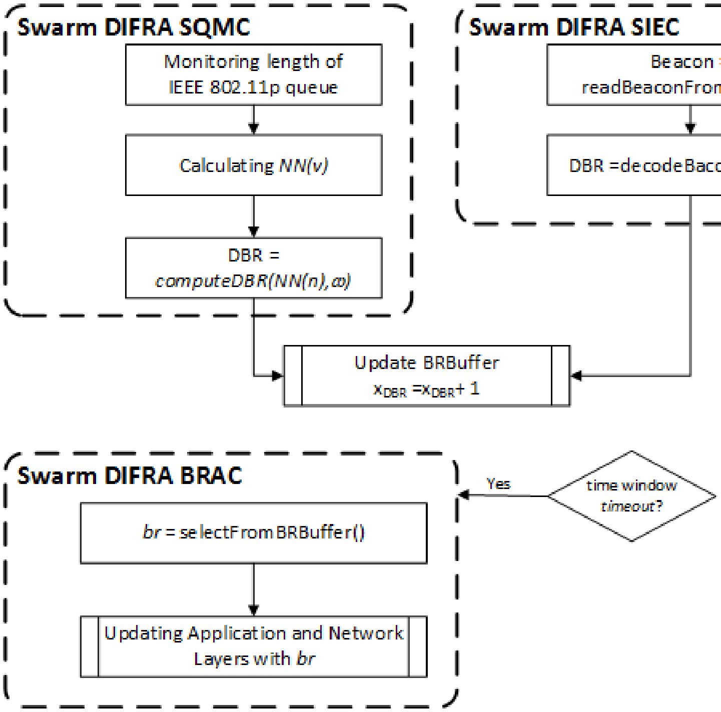

Swarm DIFRA SQMC
Monitoring length of IEEE 802.11p queue
Calculating NN(v)
DBR = computeDBR(NN(n),ω)
Swarm DIFRA SIEC
Beacon = readBeaconFromQueue()
DBR =decodeBacon(Beacon)
Update BRBuffer
x_DBR =x_DBR+ 1
Swarm DIFRA BRAC
br = selectFromBRBuffer()
Updating Application and Network Layers with br
Yes
time window timeout?

Fig. 3 Complete flowchart of the Swarm DIFRA algorithm.

value in the new beacons in order to inform the neighbor nodes. The SIEC decodes the beacons received to extract the DBR requested by the neighbors to update the BRBuffer. The buffer is modified by increasing the DBR-th component according to DBR received, i.e., $x_{DBR} = x_{DBR} + 1$.

After a given time window, BRAC is executed to compute the new beaconing frequency br^{t+1} according to the BRBuffer. We propose two different variants of Swarm DIFRA depending the what metric is evaluated to select the br^{t+1} to use: the **Swarm DIFRA-med**, which chooses the median value between the two most frequently requested values of the BRBuffer as new beacon rate, and the **Swarm DIFRA-mod**, which selects the most requested one (the mode). For example, if BRBuffer = [0 0 5 15 20 40 15 30 5 0], in Swarm DIFRA-med it holds that $br^{t+1} = 7$ and in Swarm DIFRA-mod it holds that $br^{t+1} = 6$.

4 Experimental Analysis

The experiments are carried out by using MATLAB. Each analyzed congestion control method is simulated 100 times over the same highway scenarios. The reason is the probabilistic wireless signal propagation model applied produces different communication results over the same scenario. We now define the highway VANET scenarios used in the experiments and discuss the numerical results.

4.1 Highway VANET Scenarios

The distributed beacon rate optimization is studied in a segment of a highway of two kilometers long and six lanes (three lanes in each direction). Four highway VANET scenarios are defined over this road by changing the mobility models to test the performance of the congestion control algorithms in different vehicular network distributions. Two low density scenarios (with 200 and 240 vehicles) and two high density scenarios (with 320 and 380 vehicles) have been defined.

In order to define realistic highway scenarios, the vehicles are assigned to a given lane randomly. The vehicles have higher probability to be assigned to the external lanes than to the internal ones. The speed per lane increases from the external to the internal ones. The distances between vehicles and the speeds are computed according to **the square law**, i.e., $speed^2 \simeq distance/100$.

Following the MATLAB simulation presented in Mir et al. (2015) [5]: 1) the VANET applications require exchanging beacons with a frequency that ranges from 1 Hz to 10 Hz; 2) the wireless devices utilized have a communication range (r) of 250 meters; and 3) it is considered that the maximum size of the queues of IEEE 802.11p ($MaxQ$) is 400 and threshold limit ratio (α) is 0.8.

4.2 Numerical Results and Discussion

This section discusses the results obtained by the proposed DIFRA congestion control methods in the defined scenarios. In addition, other two additional methods are included in the experiments as a baseline for the comparisons:

- ***Aloha time slot*** based method [7]: The nodes broadcast beacons at the beginning of a given time slot without analyzing the medium, i.e., regardless whether there are other nodes using the medium or not.
- ***CSMA first*** based method: When nodes have to transmit a given beacon they first analyze the medium. If no other node is using it then they broadcast the beacon, otherwise they drop the given beacon since new beacons will be generated after a short while. Thus, it reduces the likelihood of collisions.

These last two broadcasting methods do not adapt their beacon rates. Therefore, we have evaluated them with three different fix frequencies: 1, 5, and 10 Hz.

Table 1 summarizes the experimental results by showing the average and the normalized standard deviation values of the two optimized metrics, *channel occupancy* (see Equation 1) and *network balance* (see Equation 2), respectively.

Analyzing the occupancy, there are three groups of results: the ones that guarantee the proper operation of the network ($\eta(v) \leq \alpha$), the ones that may incur in a critical drop of QoS ($\alpha < \eta(v) \leq 1.0$), and the ones that exceed the channel capacity

Table 1 Experimental results in terms of channel occupancy and balance.

Algorithms	Channel occupancy				Balance			
	low density		high density		low density		high density	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Self DIFRA	0.84	0.71%	0.85	0.31%	0.29	37.20%	0.45	13.81%
Swarm DIFRA-med	0.73	14.02%	0.69	16.48%	0.08	74.32%	0.11	55.17%
Swarm DIFRA-mod	0.74	2.78%	**0.73**	3.70%	**0.03**	87.12%	**0.02**	100.93%
Aloha-1Hz	0.14	29.29%	0.22	8.63%	-	-	-	-
Aloha-5Hz	0.71	29.30%	1.08	8.64%	-	-	-	-
Aloha-10Hz	1.42	29.25%	2.16	8.65%	-	-	-	-
CSMA-1Hz	0.14	29.24%	0.22	8.64%	0.000	0.00%	0.000	0.00%
CSMA-5Hz	0.69	8.88%	0.76	0.21%	0.10	289.00%	0.66	15.02%
CSMA-10Hz	**0.78**	5.87%	0.87	2.17%	0.92	21.23%	1.23	5.54%

$(\eta(v) > 1.0)$. In Table 1 the second and third groups are shaded with light and dark gray, respectively.

As expected, the **aloha based methods** are the least competitive ones. They allow the nodes to communicate with very low beacon rates ($1\ Hz$). There is no deviation in their behavior, because they always broadcast packets, i.e., always $\sigma(v) = 0$. The **CSMA based methods** improve the performance of the aforementioned non-adaptive methods. Taking into account just the channel occupancy metric, they offer the best values in low density highway scenarios when CSMA-10Hz is used. However, they incur in a very high cost in terms of balance. This means that the VANET applications cannot property operate because many nodes do not broadcast their beacons.

Regarding to **Self DIFRA**, it has exceeded the α value defined in our experiments (0.8). Therefore, it provokes a drop in the QoS of the network. This is because the computations take into account just the information of their neighborhood (1-hop nodes), and ignore the rest of the nodes in the same cluster. In addition, the fairness between the nodes (balance) is the least competitive of all DIFRA methods, but in general better than non-adaptive methods.

Analyzing the **Swarm DIFRA** algorithms, they provided the best trade-off results in terms of occupancy and fairness. Taking into account the metric used to select the values from the BRBuffer (median or mode) we can observe a different behavior. The algorithm that uses the mode (the most repeated value in BRBuffer) is significantly the most competitive one. This leads to the vehicles broadcast beacons with high and similar beacon rates, which is the desirable behavior to allow the proper operation of VANET safety applications.

The aforementioned results are statistically confirmed by using Friedman and Wilcoxon statistical tests that resulted with *p-value*$<<$0.01. These non-parametric tests have been applied because the distribution of the results are not normally distributed according to Kolmogorov-Smirnov test.

5 Conclusions

This study has evaluated the FBR optimization problem to address the congestion control in VANETs. Thus, a set of intelligent greedy distributed broadcasting methods (DIFRA) has been devised to be applied in such a problem. These methods are based either on isolated information monitored by the node (Self DIFRA) or on combining the monitored information with the shared one received in the same beacons broadcasted by the neighbor nodes (Swarm DIFRA).

In the light of the results, we conclude that the Swarm DIFRA methods are the most competitive ones in terms of both evaluated metrics: channel occupancy and fairness (balance). Specifically, the Swarm DIFRA-mod method presented the best trade-off between these two metrics. In addition, Self DIFRA algorithms outperformed other baseline broadcasting methods.

The Swarm DIFRA methods have demonstrated to provide a reliable and non-complex congestion control method. They just require to include a natural value from 1 to 10 in the beacon as the control information for the broadcasting algorithm (i.e., four bits long).

The main lines of future research are mainly two: i) evaluating the proposed congestion control methods by using realistic urban VANET scenarios aiming at confirming their competitive performance; and ii) using as staring point the Swarm DIFRA algorithms devised here for developing new distributed broadcasting methods that utilize modern computational intelligence strategies (e.g., Neural Networks or Swarm Intelligence).

Acknowledgments This work was partially founded by the Spanish project MINECO TIN2014-57341-R (http://moveon.lcc.uma.es). University of Malaga, International Campus of Excellence Andalucía Tech.

References

1. Campolo, C., Molinaro, A., Scopigno, R. (eds.): Vehicular Ad Hoc Networks - Standards, Solutions, and Research. Springer (2015)
2. Fallah, Y.P., Huang, C., Sengupta, R., Krishnan, H.: Congestion control based on channel occupancy in vehicular broadcast networks. In: 2010 IEEE 72nd. Vehicular Technology Conference Fall (VTC 2010-Fall), pp. 1–5. IEEE (2010)
3. Han, C., Dianati, M., Tafazolli, R., Kernchen, R., Shen, X.: Analytical study of the ieee 802.11p mac sublayer in vehicular networks. IEEE Transactions on Intelligent Transportation Systems **13**(2), 873–886 (2012)
4. Hartenstein, H., Laberteaux, K.: VANET Vehicular Applications and Inter-Networking Technologies. Intelligent Transport Systems. John Wiley & Sons, Upper Saddle River (2009)
5. Mir, Z.H., Toutouh, J., Filali, F., Alba, E.: Qos-aware radio access technology (rat) selection in hybrid vehicular networks. In: Kassab, M., Berbineau, M., Vinel, A., Jonsson, M., Garcia, F., Soler, J. (eds.) Communication Technologies for Vehicles. Lecture Notes in Computer Science, vol. 9066, pp. 117–128. Springer International Publishing (2015)

6. Sattari, M.R.J., Noor, R.M., Keshavarz, H.: A taxonomy for congestion control algorithms in vehicular ad hoc networks. In: 2012 IEEE International Conference on Communication, Networks and Satellite (ComNetSat), pp. 44–49. IEEE (2012)
7. Tanenbaum, A.: Computer Networks. Prentice Hall Professional Technical Reference, 4th edn. (2002)
8. Zhang, W., Festag, A., Baldessari, R., Le, L.: Congestion control for safety messages in vanets: concepts and framework. In: 8th International Conference on ITS Telecommunications, ITST 2008, pp. 199–203. IEEE (2008)

Low Cost Software Prototyping of a Diagnosis Computer

Marisol García-Valls

Abstract Diagnostic systems are software and hardware-based equipment that interoperate with an external monitored system. They have typically been expensive equipment running algorithms to monitor physical properties in, e.g., vehicles, or civil infrastructure equipment. As computer hardware is increasingly powerful (whereas its cost and size is decreasing) and communication software become simpler to program and more run-time efficient, new scenarios are enabled that yield to lower cost monitoring solutions. This paper presents a low cost approach towards the development of a diagnostic system relying on a modular component-based approach and running on a resource limited embedded computer. Results on a prototype implementation are shown that validate the presented design.

Keywords Middleware · Diagnosis computer · Software architecture

1 Introduction

Diagnosis systems (e.g. vehicle diagnosis) interact with physical processes that sense and monitor in order to detect faulty operation They typically provide comprehensive functionality, a self-explanatory operating concept, and the capacity of handling increasing data volumes that must fuse to extract meaningful results to users. As the amount of monitored and sampled data may yield huge volumes, these could even be be analysed in a cloud infrastructure. In time sensitive domains, the threats to the predictable cloud computing technologies have to be carefully considered as explained in [9]. On the other side, the hardware equipment that supports the execution is of paramount importance; it has to integrate the suitable communication interface to

M. García-Valls(✉)
Universidad Carlos III de Madrid, Av. de la Universidad 30, 28911 Leganes, Spain
e-mail: mvalls@it.uc3m.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_48

the monitored object (e.g., vehicle), the needed computation power to provide timely execution of the diagnosis functions, and suitable verification operations [17].

From the early days of life of such systems where the cost of the hardware part was typically high, evolution towards more affordable systems has been favored by the fast progress of the computers technology. Nowadays, the market offers unexpensive solutions based on embedded processors ranging from a few dolars to a few hundred dolars; therefore, the actual cost of the hardware equipment is not a barrier to provide sophisticated and powerful diagnostic systems over plaftorms with a number of processing cores.

This paper presents a low cost alternative to designing a diagnosis system. A modular design is used based on portable code technology that may execute a number of state machines with the diagnostic functions. The design is prototyped on a low cost hardware processor, to show the feasibility of the approach.

The paper is structured as follows. Section 2 describes the related work. Section 3 presents the approach provided to the engineering of a diagnosis system. Section 4 validates the design with a prototype implementation. Section 5 concludes the work and draws future work.

2 Related Work

A number of diagnostic systems have appeared over the last decades that integrate intensive software usage in the monitoring and detection of operational faults (software and hardware) in all types of systems. Specially interesting are those related to vehicle diagnosis, where a number of contributions have appeared mainly as patents, e.g. [1], [3], or [2]. These, and other related inventions and works, do not expose cleanly the software design of the system. Despite the fact that some of them are intensive software systems, the necessary software platform and architecture is mostly neglected in these works. Fundamental elements such as the communication middleware architecture and its relation to the actual software pattern of the communication units is not well elaborated.

In the last few decades a number of middleware technologies have appeared for supporting remote operation and easing interoperability. Examples are traditional component based technologies such as Corba [5], its light-weight evolution Ice (Internet Communication Engine) [7]; object oriented middleware such as RMI (Remote Method Invoction) [4] that is a language dependant solution, and other message based technologies such as JMS (Java Messaging Service), AMQP (Advanced Message Queuing Protocol) [12]. In the last decade, publish-subscribe data centric middleware such as DDS (Data Distribution Systems for Real-Time applications) [6] have become de-facto standards in some domains.

Enhancements to these technologies have improved their benefits for specific contexts, such as to support dinamic execution [8]; for real-time reconfiguration of service-oriented architectures the iLAND middleware [10] provides time-bounded operation. Improved resource management to the operating systems has also enabled the higher efficiency in these solutions. Recently, the *Oma-Cy* architecture [16] has provided a reference design to implement middleware for cyber-physical systems, where the on-line verification part is of paramount importance to support dynamic behavior.

In summary, approaches to diagnostic systems have not sufficiently exposed the software design part, nor have they been designed to favor the portability and flexibility diagnostic computer and its counter part in the monitored system. This paper contributes to filling this gap by providing a simple but clean software design that supports the development of diagnotic systems focusing at the components that act as bridges to different underlying middleware solutions. Previous contributions on middleware bridges were presented in [13] for DDS based communications in remote surveillance systems, and [14] for distributed Ada applications in critical domains. However, these earlier approaches focuse strictly on the bridging of a middleware technology towards other generic middleware implementations.

3 System Engineering of a Diagnosis Computer

The system overview is shown in figure 1. It has an emulated part that replicates the vehicle logic for performing auto-tests upon request from the user.

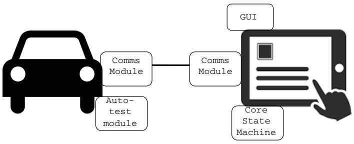


Fig. 1 Diagnosis system design based on modules

The system contains two main blocks: the *vehicle* software and the *diagnosis computer* (DC) software. DC modules are as follows. *Diagnosis Computer Core State Machine* (DSM) is the active entity that governs the execution of the diagnosis system. It determines the execution path of the different functions and units in the system. *Communications Module* (Comms) implements the communication protocol between the monitored vehicle and DC. The protocol identifies the specific data that is exchanged and the communication sequence (containing acknowledgements and startup). *Graphical User Interface* (GUI) is the component that displays the information to the user and requests user inputs to guide the operation . User inputs are fed to the state machine component, parameterizing its execution.

The vehicle components are listed as follows. *Auto-test module* (ATM) contains the algorithms that monitor parameters of the physical system. The ATM receives the user input through the communication module and runs the requested test. A variable number of algorithms can be executed, favoring maintanance and flexibility. *Communications Module* (VComms) implements the vehicle side of the communication protocol to exchange information with the user.

The *Core State Machine* component contains a set of classes that model the needed information about the vehicle and the tests to be run, i.e.,:

- the vehicle model (`VehModel`) with all the vehicle paramaters that can be monitored through the tests;
- the list of all available tests for the specific vehicle (`TestList`);
- the parameter values that are fed to tests (`TestParameters`), coming from the user specification upon the launching of the test execution;
- the set of results that each test outputs (`TestResultsData`).

Comms module includes classes to design and implement the communications protocol among the vehicle and the diagnosis computer, i.e.:

- the set of messages that can be exchanged (`MessageList`) between the diagnosis computer and the vehicle communication module;
- the set of commands (`CommandsList`) that the diagnosis computer can issue to the vehicle. Examples are to run a specific test, to relaunch a test, or to provide further information on a test, among others.
- the set of connection resources used for the communication `SocketPoints` that uses sockets with a transport protocol that can be selected. Security can be added for data encription by using SSL.

The graphical user interface module *GUI* performs the friendly display of the diagnosis computer operation. It has information about the specific display characteristics (`DisplayLayout`); it allows to easily change to a different display computer (e.g., a touchpad display of a different resolution, size, etc.) as the specific characteristics of the hardware display are hidden in this class. Also, GUI contains is designed to support different display modes such as for technical users or for the accounting staff to gather statistics on the specific vehicle failures.

An abreviated class diagram of the system is shown in figure 2. It presents some selected data on two important classes. Class `VehModel` presents the data model of the vehicle. Its attributes are the set of parameters that define the vehicle characteristics refering to the ECUs or electronic control units. These units control all the operation of the car, carrying out functions at all levels from the engine control, driving assistance, or the less critical passenger comfort. Each car function is divided into subsystems that are express in each of the attributes `PhyParn` (*physical parameters of subsystem* `n`) that model a given subsystem (subsystem `n` in this case).

The communications module's interface is presented in figure 2, precisely in the interface `CommsI`. It shows the basic operations for the communication:

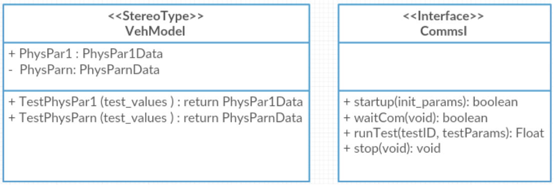

<<StereoType>>
VehModel
+ PhysPar1 : PhysPar1Data
- PhysParn: PhysParnData
+ TestPhysPar1 (test_values) : return PhysPar1Data
+ TestPhysParn (test_values) : return PhysParnData
<<Interface>>
CommsI
+ startup(init_params): boolean
+ waitCom(void): boolean
+ runTest(testID, testParams): Float
+ stop(void): void

Fig. 2 Simplified class diagram of the diagnosis computer side

- vehicle auto-test module start up (`startup`) and shut down (`stop`); by issueing this invocation, the vehicle module begins its operation and the values for the auto-test runs are initialized.
- the vehicle auto-test module has a mode to wait for communication from DC (`waitCom`); in this mode, the vehicle auto-test module is not running any test, but purely idle in attention to connection requests from DC.
- specific tests can be requested by DC by invoking the method `runTest` and specifying as parameter a given test with execution parameters.

The central module of the diagnosis computer, the *Core State Machine* module triggers the operation of the tests in the vehicle. Its operation is shown in figure 3. All indicated states are run in the diagnosis machine.

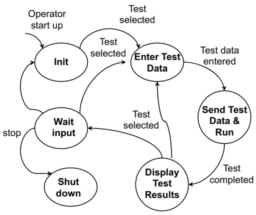


Fig. 3 Operational sequence of the Core State Machine module

The startup of the system (init button pressing) places the DC in the initial state (*Init*), indicating that no previous tests have been run since the system is active. From that state, an operator/user may select a specific test to be run; the system enters the *Enter Test Data* state and indicates the specific threshold values to check for the specific test. Upon completing the entering of the test data and pressing the *run test* option, the information is sent to the vehicle software and there the test is run; the DC then enters state *Send Test Data & Run*. Once the test is run and the test results are sent back by the vehicle software, the DC enters state *Display Test Results* where the information is displayed to the operator. After this display, the DC enters state *Wait input* for further operation. If a new test is to be performed, the sequence resumes by entering state *Enter Test Data*.

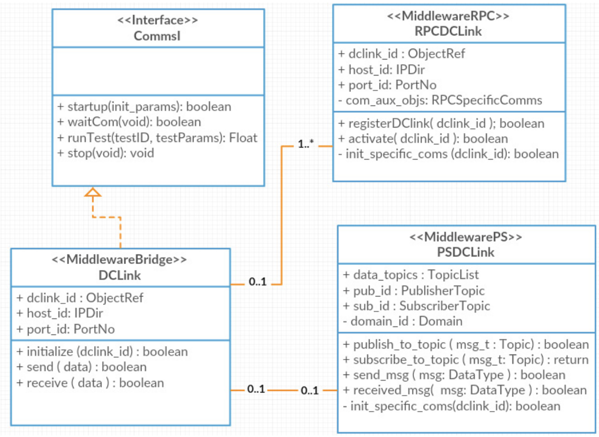

<<Interface>>
CommsI
+ startup(init_params): boolean
+ waitCom(void): boolean
+ runTest(testID, testParams): Float
+ stop(void): void
<<MiddlewareRPC>>
RPCDCLink
+ dclink_id : ObjectRef
+ host_id: IPDir
+ port_id: PortNo
- com_aux_objs: RPCSpecificComms
+ registerDClink(dclink_id); boolean
+ activate(dclink_id): boolean
- init_specific_coms (dclink_id): boolean
1..*
0..1
<<MiddlewareBridge>>
DCLink
+ dclink_id : ObjectRef
+ host_id: IPDir
+ port_id: PortNo
+ initialize (dclink_id) : boolean
+ send (data) : boolean
+ receive (data) : boolean
0..1
0..1
<<MiddlewarePS>>
PSDCLink
+ data_topics : TopicList
+ pub_id : PublisherTopic
+ sub_id : SubscriberTopic
- domain_id : Domain
+ publish_to_topic (msg_t : Topic) : boolean
+ subscribe_to_topic (msg_t: Topic) : return
+ send_msg (msg: DataType) : boolean
+ received_msg(msg: DataType) : boolean
- init_specific_coms(dclink_id): boolean

Fig. 4 Flexible bridge to use multiple middleware backbones for communication

Figure 4 shows the modular design of the communication module. Multiple middleware technologies can be used as the actual communication means, as it is abstracted by the *DCLink* class that acts as a general communication description that can latter be mapped to different middlewares. In figure 4, two options are shown: an RPC (*remote procedure call* paradigm) such as Java Remote Method Invocation [4], CORBA [5] or the Internet Communication Engine [7]; and a P/S (*publish-subscribe*) middleware such as DDS [6]. Lower level mechanisms can be easily adapted to this structure, e.g., socket based communications. It is also possible to use this bridging for more complex communication schemes that could support communication with multiple vehicle units using paradigms that support dynamic reconfiguration such as iLand middleware [10, 11].

The communication interface is `CommsI` that specifies the basic functionality of the DC interfacing module. This module is initiated via *startup* function that performs the initialization operations that vary according to the specific middleware implementation that is selected. For example, in Ice, two environment objects must be created (i.e., *communicator* and *adapter*) that create a remote object that is exported to the public domain and is visible for the vehicle software part. In the case of other technologies such as DDS, a *domain* has to be created, among other entities such as the *domain participants*, *writers*, *readers*, *publishers*, and *subscribers*. This specific per-technology communication structure is abstracted in the `DCLink` class.

Most available middleware technologies support the usage of multi-language and multi-platform; it is possible that both ends of the communication are implemented in different programming languages over different operating systems that is often enabled by using an interface definition language (IDL). An important consideration in such a case is that some programming environments require the addtion of a virtual machine, e.g., Java or C#. For efficiency reasons, the proposed model considers that the same middleware technology is used at both communication end points (vehicle software and DC software).

4 Validation

The diagnosis computer system prototype is implemented in two main parts: the monitored object is implemented and emulated in a desktop computer that simulates the execution of algorithms that monitor physical parameters in the engine subsystem; the diagnosis computer is realized in a bytecode processor that is an actual embedded computer with limited computation power such as aJile's aJ10x family of Java bytecode processor [18]. The desktop machines runs a Java based prototype on a Ubuntu 10.04 Linux distribution and a Java SE 7 for RMI-enabled connection. The *Comms* module uses a TCP/IP connection for message exchanges between DC and vehicle software.

The results shown present the overhead incurred by our prototype in a sequence of 500 tests of an average duration of 5*s* each. The prototype uses a Java environment with an underlying TCP socket implementation. For the test duration, the communications module shows that the overhead is influenced by the characteristics of the aJ100 embedded processor that is a resource limited environment; it is a 32 bit processor with direct execution with support for multiple JVMs, 48KB RAM memory. By using a more powerful device such as RaspberriPi series with ARM processors, the increase in response time could be highly relevant. Figure 5 shows the communication time between the DC and the vehicle. On the one side, it shows only the time taken by the interaction (i.e., the network time plus the time taken by the TCP/IP software stack to process the communication). On the other side, it also shows the overall time that includes the temporal cost of the tests that are 5*s*, therefore a consistent greater cost is shown. The design is highly portable and has been adapted to use a full fledged C++ environment with Ice version 3.1 both over TCP and UDP transports. The portability of the *CommsI* module is easily achived. The results were tested over both destop patforms with Intel dual core processors running at 2.6GHz with 1GB of memory. Overhead is reduced to less that 0.02%.

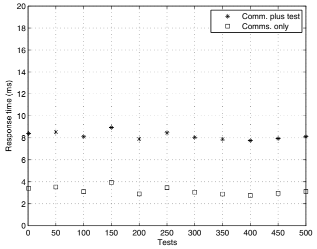


Fig. 5 Communication time with and without including the test times

5 Conclusion

The paper has presented the flexible design of diagnostic systems based on components and with a specific structure that eases the portability to different underlying middleware technologies. The proposed design has been prototyped in a resource constraint environment based on a Java embedded processor, showing a time overhead that is suitable for this type of domain and for the duration of the performed tests. The design has also shown to be very flexible as it was ported to a different underlying middleware that uses a different programming language and an IDL for the specification of the *CommsI* interface in a reduced time. This shows that the proposed approach is suitable for diagnosis systems as the general framework provides a simple and clean structure that is easily adaptable to run on multiple underlying communication backbones.

Acknowledgement This work has been partly funded by the project REM4VSS (TIN2011-28339) funded by the Spanish Ministry of Economy and Competitiveness.

References

1. Brunemann, G., Dollmeyer, T. A., Mathew, J.C.: System and method for transmission of application software to an embedded vehicle computer. US 6487717 B1. US patent (2002)
2. Lowrey, L.H., Banet, M.J., Lightner, B., Borrego, C., Myers, C., Williams, W.: Internet-based vehicle-diagnostic system. US 6611740 B2. US Patent (2003)
3. Parrillo, L.C.: Wireless motor vehicle diagnostic and software upgrade system. US 5442553 A. US Patent (1995)
4. Sun Microsystems: Java^{TM} Remote Method Invocation API (2016). http://docs.oracle.com/javase/7/docs/technotes/guides/rmi/
5. Object Management Group: The Common Object Request Broker. Architecture and Specification, Version 3.3, November 2012. http://www.omg.org/spec/CORBA/3.3
6. Object Management Group.: A Data Distribution Service for Real-time Systems Version 1.2. Real-Time Systems (2007)
7. ZeroC Inc.: The Internet Communications Engine (2003). http://www.zeroc.com/ice.html
8. Cano Romero, J., García-Valls, M.: Scheduling component replacement for timely execution in dynamic systems **44**(8), 889–910 (2014)
9. García-Valls, M., Cucinotta, T., Lu, C.: Challenges in real-time virtualization and predictable cloud computing. Journal of Systems Architecture **60**(2), 726–740 (2014)
10. García-Valls, M., et al.: iLAND An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems. IEEE Transactions on Industrial Informatics **9**(1), 228–236 (2013)
11. García-Valls, M., et al.: A Dual-Band Priority Assignment Algorithm for QoS Resource Management. Future Generation Computer Systems **28**(6), 902–912 (2012)
12. ISO/IEC Information Technology Task Force (ITTF). OASIS AMQP1.0 – Advanced Message Queuing Protocol (AMQP), v1.0. ISO/IEC 19464:2014 (2014)
13. Rodríguez-López, I., García-Valls, M.: Architecting a common bridge abstraction over different middleware paradigms. In: Ada-Europe 2011, pp. 132–146 (2011)
14. García-Valls, M., Ibánez-Vázquez, F.: Integrating middleware for timely reconfiguration of distributed soft real-time systems with ada DSA. In: Ada-Europe 2012, pp. 35–48 (2012)
15. García-Valls, M., et al.: An architecture of a quality of service resource manager middleware for flexible multimedia embedded systems. In: LNCS, vol. 2596 (2003)

16. García Valls, M., Baldoni, R.: Adaptive middleware design for CPS: considerations on the OS, resource managers, and the network run-time. In: Proceedings of the 14th Workshop on Adaptive and Reflective Middleware (ARM@Middleware) (2015)
17. García-Valls, M., et al.: Time-sensitive adaptation in CPS through run-time configuration generation and verification. In: COMPSAC (2014)
18. aJile Systems, Inc.: Low power real-time network direct execution SOC for the Java ME platform. White paper (2016). (http://www.ajile.com)

Facial Expression Recognition System for User Preference Extraction

Naoya Yamaguchi, Maria Navarro Caceres, Fernando De la Prieta and Kenji Matsui

Abstract This paper describes our preliminary study of facial expression recognition in order to extract user response information. We used Kinect to get real time facial expressions of the user to extract 6 facial expression categories (neutral, happiness, disgust, surprise, sadness, angry). As for the recognition process, we applied a multi-layer-perceptron to classify the face expressions. A total of 1,912 facial expression data sets were collected from 16 subjects. We performed holdout test using 80% of training data and 20% of test data. The recognition rate without "sadness" feature was around 90%, and the rate using every categories was around 80%. The positive results obtained shows this system as a proper one to measure user preferences in a visual test.

Keywords Face recognition · Kinect · Neural net · Categorical classification

1 Introduction

In our life, facial expression is an important role in communication. Therefore, observing listener's facial expression by computer may have a potential to solve the problems, and we might be able to obtain accurate user feedbacks automatically.

Face detection has been studied by many researchers, and latest powerful approaches are based on 3D images. However, face detection in video frames has not been extensively studied. Vinnetha et al. [5] described facial expression recognition using Kinect 3D features. A. Youssef et al. [6], described their attempt to recognize facial expressions using a 3D Kinect sensor. They constructed a

N. Yamaguchi · K. Matsui
Graduate School of Engineering, Osaka Institute of Technology, Osaka, Japan

M.N. Caceres · F. De la Prieta(✉)
Department of Computer Science, University of Salamanca, Salamanca, Spain
e-mail: Salamanca.fer@usal.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_49

training data set containing time dimension. For individuals who did not participate in training the classifiers, the best accuracy levels were 38.8% (SVM) and 34.0% (k-NN). However, in case of closed test, the best accuracy levels that they obtained raised to 78.6% (SVM) and 81.8% (k-NN).

Puica et al. [4] showed emotion recognition from facial expressions using Kinect sensor with the Face Tracking SDK. The accuracy of emotion recognition with data outside the training set was off 80%.

However, the results of the systems described showed that the expression of sadness and disgust were more difficult than the others to recognize [5,6]. In order to solve this limitation, a facial expression detection is here performed by applying a trained neural net. The Kinect has been the tool to recognize the faces in video frames, using a multilayer perceptron to classify the different expressions detected in the users' faces. The positive results obtained showed the accuracy of this system to detect sad or disgust expressions.

Likewise, to evaluate some systems based on art, like music or visual art, an empirical test is usually performed. This test consists of a list of questions about several audio files or digital images where they give a punctuation of the quality of the art shown. However, in such situations the results obtained can differ from what listeners really feel when the sounds are played. Also, detailed subjective evaluation requires a lot of effort and time. Thus, a facial recognition application could be helpful to validate this kind of results more efficiently.

Therefore, the contribution of this work is twofold. On the one hand, detect several sentiments expressed by the users is aimed using a neural net. On the other hand, an application is proposed to carry out some test for other subjects, such as musical listenings or work of art evaluations.

This article is structured as follows. Section 2 contains the overall description of the system. Section 3 describes the experiments performed for the facial recognition. Section 4 explains the preliminary results obtained and Section 5 presents the conclusions and future work.

2 System Description

An automatic face expression recognition system falls into the following steps: face detection and location in a practical scene, facial feature analysis, and facial expression pattern matching. The application requires the user to be seated in front of the Kinect Sensor. Then, the device detects the human body and estimates the position of his head, drawing a face on this position. Once the face is isolated, 17 features are extracted to be the input of the multilayer perceptron. This classifier properly trained gives the facial expression of the user. This overall process is shown in Fig. 1.

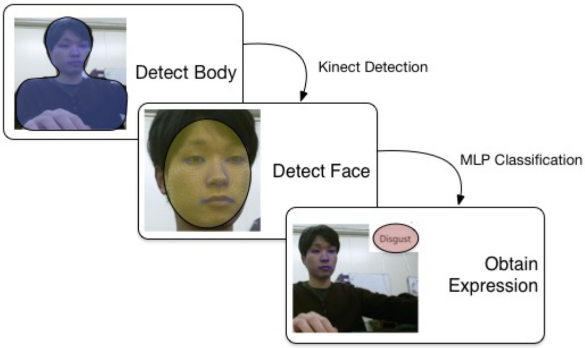


Fig. 1 Overall system process to detect facial expressions.

Section 2.1 describes some kinect details and the features extracted from the face. Section 2.2 explains the multilayer perceptron design and the training set.

2.1 *The Architecture of Face Expression Recognition*

The main tool used to retrieve the frames is the kinect [7]. This tool has been successfully applied in different research contexts. Khoshelham et al. [3] uses the kinect in map indoor data. Chang et al. [1] applies the device to improve skills in young adults with motor disabilities.

It also exists a variety of versions whithin the kinect devices. In this study we utilized KinectV2 Sensor for the face tracking and detection as well as feature extraction. KinectV2 Sensor was released in 2014 to be connected to windows devices, thus it permits to work with a computer. It has a RGB camera, depth sensor, and a microphone array (See Fig. 2). Together with Kinect, Microsoft provides a SDK as a software development tool kit and a software development kit HD Face API for face tracking. That makes it possible to track the user's face in the real time

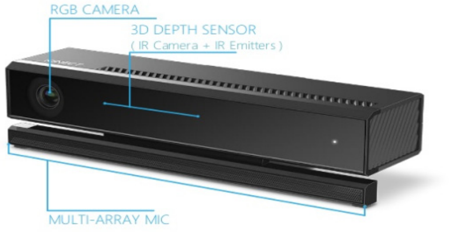


Fig. 2 Main external features of the kinect device

The first step consists of detecting the users' head from the frames obtained by the kinect camera. To do so, it has been used a method called "Time of Flight (TOF)", which relies upon an image sensor to measure the time of laser light pulses to travel to a target surface. Based on that depth information, the face position is calculated and tracked in each frame emitted by the kinect. Then, the system starts to extract the information from the face expressions. For this purpose, the Face API provides various useful properties for the face expression recognition. In this study, we used FaceShapeAnimation Enum. This Enum property has Shape Units (SU) and Animation Units (AU). SU represent shape features of the detected 3D face model, while AU represents motion features of the face. After examining the behavior of those features, AU values were used in this study. In the API property, there are 17 AUs as shown in Fig. 3: 1) JawOpen, 2) LipPucker, 3) JawSlide, 4) LipStretcherRight, 5) LipStretcherLeft, 6) LipCornerPullerRight, 7) LipCornerPullerLeft, 8) LipCornerDepressorLeft, 9) LipCornerDepressorRight, 10) LeftCheekPuff, 11) RightCheekPuff, 12) LeftEyeClosed, 13) RightEyeCloed, 14) RightEyebrowLowerer, 15) LeftEyebrowLowerer, 16) LowerLipDepressorLeft, 17) LowerLipDepressorRight.

Most of the AUs are expressed as a numeric weight varying between 0 and 1, three of them, JawSlide, RightEyebrowLowerer, and LeftEyebrowLowerer, vary between -1 and +1. Kinect updates those AU values every frame.

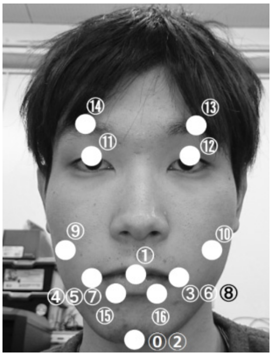


Fig. 3 The locations of the 17 Animation Units.

These data extracted from the kinect are used as input for the MLP module to classify the different facial expressions.

2.2 The Architecture of Face Expression Recognition

The architecture of our Face Expression Recognition system is based on multilayer-perceptron implemented by Weka and applied the default configurations.

Figure 4 shows the basic architecture of multilayer-perceptron classifier. As we can see, the inputs are the AU values obtained with the kinect. The outputs are the different type of faces we target in this article. Due to the difficulty of recognizing sad faces, we develop one first system without sadness recognition, and then we added this feature in a more complex design. The number of hidden layers are also changed in order to study the classification accuracy. This system is able to recognize six different categories of facial expressions, namely, neutral, happy, disgust, surprise, sad and angry, equivalent to closed-eyes.

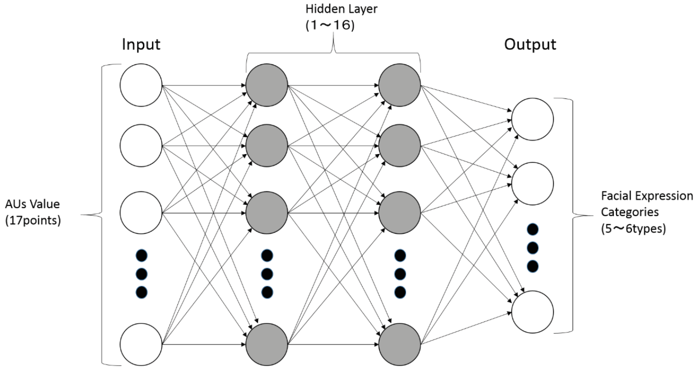


Fig. 4 Basic architecture of multilayer-perceptron based classifier

Figure 5 shows the flow chart of the MLP design. The training data are obtained from the kinect information collected. In this study, 80% of the data were used for the training, and 20% of the data were used for recognition validation. The MLP is trained with the corresponding data. The evaluation of the trained network is achieved by using the validation data. This data contains the AU values used as input for the MLP. Then, the results are compared with the expected ones and a measure of accuracy is obtained. This facial recognition accuracy depends on the number of hidden layers. Thus, we designed several iterations to change dynamically the number of hidden layers and select the best option according to the accuracy measure.

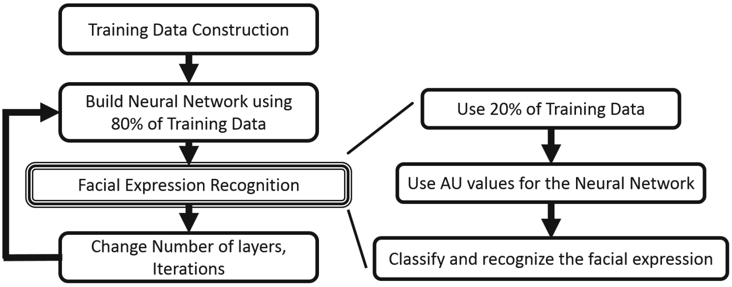

Training Data Construction
Build Neural Network using
80% of Training Data
Facial Expression Recognition
Change Number of layers,
Iterations
Use 20% of Training Data
Use AU values for the Neural Network
Classify and recognize the facial expression

Fig. 5 Flow chart of facial expression recognition system.

3 Experimental System Overview

To train our network and evaluate our design, a preliminary experiment is performed. 16 subjects were participated in this study. They were asked to place their face 1.5 meter from the KinectV2 sensor. We captured for each individual 120 images, 20 for each facial expression (happy, sad, disgust, angry, neutral and surprise), taken in twenty seconds. Thus, a total of 1,920 images are used to train and validate the system, with a set of 17 AU values for each image. We measure the accuracy by using the number of recognized faces against the number of analysed faces, according to the number of hidden layers and the number of training iterations of the neural network.

The number of nodes *N* on each layer has been stablished following the next equation:

$$N = \frac{a+c}{2}$$

Where *a* means the number of attributes measure (in this case, 17) and *c* are the number of classes to classify (in the present study, 5 or 6 classes). We obtain then a total of 11 nodes for each hidden layer.

In order to separate the dataset in training and validation data, the images were randomized and 1536 (80%) were selected to train the system, while 384 (20%) were used to validate it.

To measure the accuracy, the following equation is applied [2]:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$

Where TP means True Positive values, TN True Negative rates, FP False Positive values and FN False Negative Values according to [2].

4 Results and Discussion

The results are plotted in Fig. 6 and 7. Horizontal axes represent the number of hidden layers applied in each execution. Vertical axes represent the recognition rate (accuracy) in percentages. Color lines represent the number of iterations used. Blue line indicates 500 iterations for each number of hidden layers. Likewise, orange lines corresponds to 1000 iterations, whereas grey lines mean 1500 iterations are applied to the system.

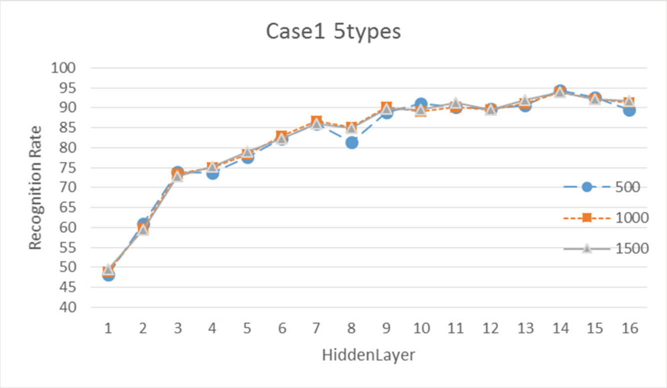


Fig. 6 Recognition results in the case of 5 facial expression categories. Vertical axis represents the recognition rate in percentages. The horizontal axis represents the number of hidden layers. The color lines corresponds to different number of iterations for each number of hidden layers.

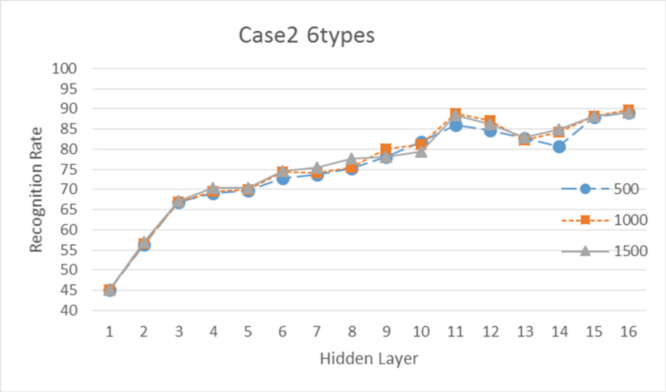


Fig. 7 Recognition results in the case of 6 facial expression categories. Vertical axis represents the recognition rate in percentages. The horizontal axis represents the number of hidden layers. The color lines corresponds to different number of iterations for each number of hidden layers.

During the system test operation, we noticed that the sadness facial expression seems confusable compare with other categories. Therefore, we decided to test both cases, i.e. 5 categories and 6 categories. Figure 6 shows the recognition results in the case of 5 facial expression categories, excluding “Sadness”. Figure 7 shows the ones in the case of 6 facial expression categories. The results showed that the classification accuracy is higher when the sad faces are excluded, although both cases have good recognition rates (above the 85% or recognition). As we can see, the classifier gives nice results when the number of hidden layers is above eleven. In the case of six faces classification, 11 layers is the best result obtained. In the first case, the best one achieved has 14 hidden layers. Thus, we can state that the classifier using between 11 and 13 layers seems good performance in any case.

It is to note that the results show quite similar results independently of the number of training iterations. We only obtain some small deviations (more than 5% in the recognition rate) in the case of fourteen hidden layers with six categories and in the case of eight hidden layers with five categories. Thus, we can conclude that the number of training iterations is not sensitive in terms of the recognition accuracy.

Although our preliminary face expression classification experiment is a sort of initial step, we were able to confirm that the proposed system setting has a potential to get user feedbacks from the facial expressions. The use of facial recognition in videoframes can be very useful to capture immediate reactions along the time for specific events, such as visualization of a work of art, listening of musical pieces. For the users, this kind of recognition can be easier, as they can make natural movements, and not staying very quiet in front of a camera, which can also bias our final results.

KinectV2 sensor with HD face API seems very powerful tool to make such facial expression recognition system in a short development period. In this study, we did not use the facial motion, however, facial motion information seems very important to be able to detect the detailed expression information.

5 Conclusions

This paper described our preliminary study of facial expression recognition in order to extract user response information. Kinect V2 HD face API is applied to solve our problem, and animation units (AU) were used to detect facial expressions of the user to extract 6 categories (neutral, happiness, disgust, surprise, sadness, angry). To recognize the different faces, a machine learning using neural network MLP is designed and trained.

A total of 1,920 facial expression data sets were collected from 16 subjects. We performed a test using 80% of training data and 20% of validation data extracted from this images taking to the 16 individuals.

The recognition rate without “sadness” feature was around 90%, and the rate using every categories was around 80%. This lead us to conclude as a good system to recognize different facial expressions accurately.

As our next step, we plan to collect the facial expression data from the real audiences to be able to investigate the effective facial expression recognition process. Also, facial motion information needs to be tested to make the system more reliable.

References

1. Chang, Y.J., Chen, S.F., Huang, J.D.: A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in developmental disabilities **32**(6), 2566–2570 (2011)

2. Gupta, N., Rawal, A., Narasimhan, V.L., Shiwani, S.: Accuracy, sensitivity and specificity measurement of various classification techniques on healthcare data. IOSR J. Comput. Eng. (IOSR-JCE) **11**(5), 70–73 (2013)
3. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors **12**(2), 1437–1454 (2012)
4. Puica, M., Florea, A.M.: Towards a computational model of emotions for enhanced agent performance. Phd Thesis. Romania: University of Bucharest (2013)
5. Vineetha, G.R., Sreeji, C., Lentin, J.: Face Expression Detection Using Microsoft Kinect with the Help of Artificial Neural Network. Trends in Innovative Computing (2012)
6. Youssef, A.E., Aly, S.F., Ibrahim, A.S., Abbott, A.L.: Auto-Optimized Multimodal Expression Recognition Framework Using 3D Kinect Data for ASD Therapeutic Aid. International Journal of Modeling and Optimization **3**(2), 112 (2013)
7. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE MultiMedia **19**(2), 4–10 (2012)

Improved Metrics Handling in SonarQube for Software Quality Monitoring

Javier García-Munoz, Marisol García-Valls and Julio Escribano-Barreno

Abstract SonarQube platform is an open source initiative with the goal of assessing the quality of the software projects. Currently, contributors have focused on providing source code analysis functions essentially for Java-based projects. However, in certain domains Java is not the predominant language such as critical software projects in transport, avionics, or medical systems. This paper presents how to provide plugins to enhance the current capabilities of the platform in order to include coding rules analysis results from external tools; the platform is then enriched with the results of external tools.

Keywords Software quality · Source code analysis · Coding rules analysis

1 Introduction

Software projects need information about almost every aspect of the development phase, like achievement of objectives, monitoring and control of activities, project costs and technical quality [2]. Metrics are essential in all engineering disciplines and, in particular, for software development (see [6]), providing a fundamental insight into the development process to assess maintainability, reliability, and even development progress. Metrics provide reproducible indicators useful to estimate the quality, performance, management, and cost within a project. They bring in benefits like the possibility of analysing the data to understand, improve, and predict future behaviours for undertaking corrective actions on time. They have increased

J. García-Munoz · M. García-Valls(✉) · J. Escribano-Barreno
Universidad Carlos III de Madrid,Madrid, Spain
e-mail: {jgarciam,mvalls}@it.uc3m.es

J. Escribano-Barreno
Indra Sistemas, S.A., Alcobendas, Spain
jebarreno@indra.es

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_50

their relevance due to their applied use and the contrasted benefits to define baseline quality indicators for serveral purposes [18]. For example, the SEI Capability Mature Model Integration (CMMI) [5] for development, relies on the usage of metrics (see [10]), and is used for evaluating the maturity of the development process of the organizations. Software quality models are also being exported to different domains such as multimedia [13]; cloud computing and virtualization technologies [11]; distributed systems based on middleware [12, 14, 16]; and on-line verification [3, 15].

The present paper describes how to enrich the metrics management and presentation in the SonarQube [26] framework, providing additional functionality to the platform in order to integrate its own analysis results with the ones from external coding-rules analysis tools in a single presentation space.

The paper is structured as follows. Section 2 describes the related work on norms and practices for technical quality and metrics. Section 3 describes the baseline framework offered by SonarQube Section 4 describes the addition of rules. Section 5 validates the design providing a snapshop of a real-world project. Section 6 draws some conclusions and future work.

2 Related Work

The metrics and quality information with respect to a software project of complexity may easily require different analysis techniques that generate results and data from different sources, possibly also collecting data in different ways [8]. For some specific projects more than the source is analysed, such as dependencies among packages such as [21]. Information about the software is, in the end collected in different ways by means of different tools. Some examples are Understand ([25]) that is a comercial tool for static code analysis, supporting multi-language; LDRA ([19]) another comercial tool for static code analysis, also with multi-language support; PC-Lint ([22] a comercial solution for static code analysis for the C/C++ languages; Splint ([27]) that enables static code analysis for C programs, with GNU licence; PMD ([23]) is a static code analyser for Java, JavaScript, XML and XSL; and SonarQube ([26][4]) that is an open source quality management platform, developed under LGPL v3 license.

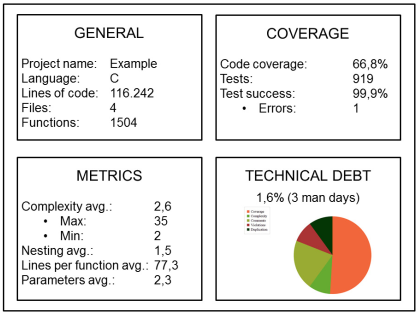
As not a single tool is capable of providing all required data, it is typically required to set up a tool chain for code analysis setting also the procedures and principles for information collection and interpretation in the form of a collaborative environment. As a result, the SonarQube platform has appeared as a platform to support advanced analysis; however, SonarQube appears in a similar way to a blank sheet of paper [9], so that the required techniques and methods to implement the needed functionality have to be designed and integrated in it.

3 Enhancing Quality Frameworks

A software quality framework tool provides different views over multiple aspects and characteristics of the source code that yield and overall quality essence. Basically, a quality framework supports analysis of the source code files. A powerful framework should also be able to analyse the results from external tools, providing a suitable enhancement mechanism for this. The result is that by integration of the own analysis algorithms and those of external tools comprehensive views over the project is provided to engineers.

Web interfacing is an essential element of today's frameworks as it enables remote work and distributed cooperation, preserving the coherence over the displayed data. The data is mostly centralised in a data base with which the software quality framework connects for storing/retreiving information for the remote users/engineers.

An example of the results output by a software quality framework are shown in figure 1. It shows general information about the project, together with some metrics, coverage test information and an estimation of the time to correct the current situation.



GENERAL

Project name:	Example
Language:	C
Lines of code:	116.242
Files:	4
Functions:	1504

COVERAGE

Code coverage:	66,8%
Tests:	919
Test success:	99,9%
• Errors:	1

METRICS

Complexity avg.:	2,6
• Max:	35
• Min:	2
Nesting avg.:	1,5
Lines per function avg.:	77,3
Parameters avg.:	2,3

TECHNICAL DEBT

1,6% (3 man days)

Fig. 1 Example of analysis results as displayed by SonarQube

The underlying logic of SonarQube is based on a *source code analyzer* component that performs basic analysis activities (such as counting lines of code) and an *application server* that graphically displays the data that results from the analysis in a browser front-end. The following elements are key to the internal function of the SonarQube framework:

- *Widgets* are the components that configure the graphical display of data resulting from the source code analysis, i.e., it enables the customization of the analysis results presentation to the user. A widget supports the specification of the visual format and display locations of the presented data. Some examples of widgets are shown in Figure 1. Each widget yields one of the square boxes that are shown. Each box contains a number of data items whose display location and characteristics is indicated in the widget code. For each new analysed project, SonarQube creates a *project dashboard* for selecting, adding, or removing the available widgets.

- *Sensors* are components that access the specific source code to be checked and that support the implementation of analysis functions over the source code to extract the metrics.
- *Decorators* are the components that support the programming of additional processing over the initial metrics provided by sensors in order to derive more complex metrics.

The framework requires to integrate a key component that controls the sequence of steps to be performed in order to launch the source code analysis. SonarQube provides the component Sonar Runner, but other tools can be employed for this matter. In order to execute a SonarQube analysis, it is then needed to initially launch *Sonar Runner* that determines the sequence of invocations of the functions provided by the *Sensors* and *Decorators*. Once the process is completed, the *runner* stores in a database all the collected data.

4 Enhanced Coding Rules Analysis

This section presents the design and implementation of the enhancements to integrate into SonarQube additional analysis of coding rules. These are integrated with SonarQube's own analysis algorithms. A specific rule analyser plugin is designed that adds new coding rules and then analyses the documents generated by other tools to present the source code parts where there is some rule violation.

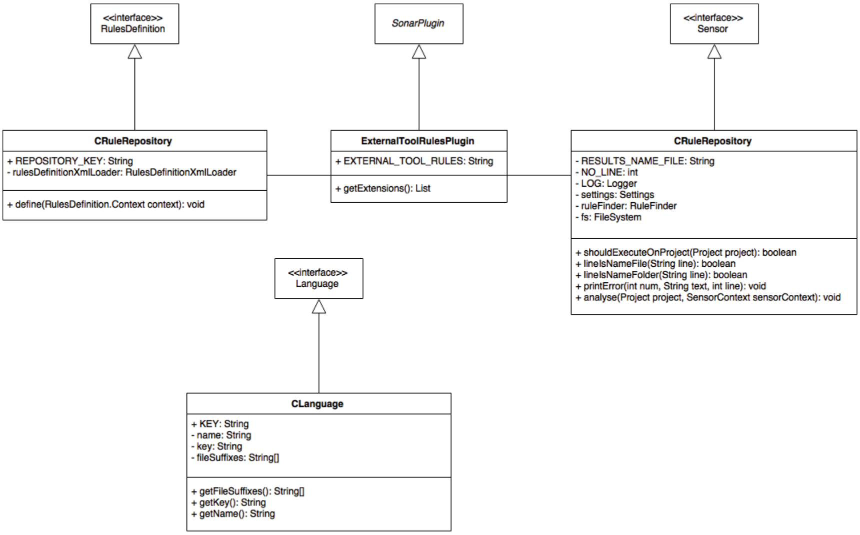

<<interface>> RulesDefinition
SonarPlugin
<<interface>> Sensor
CRuleRepository
+ REPOSITORY_KEY: String
- rulesDefinitionXmlLoader: RulesDefinitionXmlLoader
+ define(RulesDefinition.Context context): void
ExternalToolRulesPlugin
+ EXTERNAL_TOOL_RULES: String
+ getExtensions(): List
CRuleRepository
- RESULTS_NAME_FILE: String
- NO_LINE: int
- LOG: Logger
- settings: Settings
- ruleFinder: RuleFinder
- fs: FileSystem
+ shouldExecuteOnProject(Project project): boolean
+ lineIsNameFile(String line): boolean
+ lineIsNameFolder(String line): boolean
+ printError(int num, String text, int line): void
+ analyse(Project project, SensorContext sensorContext): void
<<interface>> Language
CLanguage
+ KEY: String
- name: String
- key: String
- fileSuffixes: String[]
+ getFileSuffixes(): String[]
+ getKey(): String
+ getName(): String

Fig. 2 Class diagram rule plugin

The architecture of the analyzer is shown in figure 2 that contains the following clases: (1) *ExternalToolRulesPlugin*, (2) *CRuleRepository*, (3) *CLanguage*.

ExternalToolRulesPlugin exports the location of the results generated by some other external tool that analyses coding rules.

CRulerepository represents the repository of rules that will be added to SonarQube. It defines the programming language in a single method (*define*) and it has a single constructor for SonarQube to invoke it upon initialization. Following an example of this class is shown.

```
package com.sonar;
import org.sonar.api.server.rule.RulesDefinition;
import org.sonar.api.server.rule.RulesDefinitionXmlLoader;
public final class CRuleRepository implements RulesDefinition{
  public static final String REPOSITORY_KEY = "CRules";
  private final RulesDefinitionXmlLoader rulesDefinitionXmlLoader;
  public UnderstandCRuleRepository(RulesDefinitionXmlLoader
       rulesDefinitionXmlLoader) {
    this.rulesDefinitionXmlLoader = rulesDefinitionXmlLoader;
  }
  @Override
  public void define(RulesDefinition.Context context) {
    RulesDefinition.NewRepository repo = context.
        createRepository(REPOSITORY_KEY, CLanguage.KEY).
        setName("Example");
    rulesDefinitionXmlLoader.load(repo, getClass().
        getResourceAsStream("/org/sonar/plugins/rules-
␣␣␣␣␣␣␣␣externatool/rules-c.xml"), "UTF-8");
    repo.done();
  }
}
```

CLanguage class defines the programming language that will be interpreted by SonarQube.

ExternalToolSensor is the Sensor class related to the coding rules analyser plugin. It is in charge of starting the analysis of the external tool results file and it marks the rule violations, indicating also the type and location of the violation.

Lastly, the *coding rules* of the specific selected programming language must be indicated inside the plugin. Each rule must have a unique identifier (key), a priority (info, minor, major, critical, or blocker) based on the importance of the rule. Rules are expressed in an XML file. An example is shown below.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<rules>
<rule key="no_comments" priority="MINOR">
 <name>Comments are not permitted</name>
 <configkey>NO_COMMENTS</configkey>
 <description>
 <p>This line is a comment and comments are not permitted</p>
 </description>
</rule>
<rule key="example" priority="INFO">
```

```
 <name>Example Rule</name>
 <configkey>EXAMPLE</configkey>
 <description>
   <p>This is an example</p>
 </description>
</rule>
</rules>
```

5 Validation

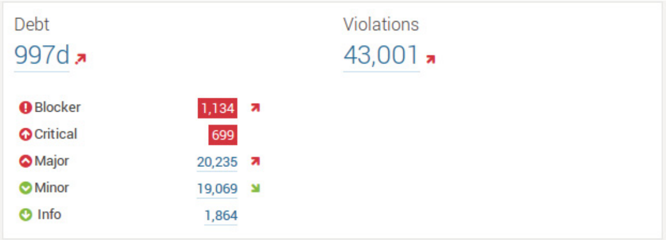
For the validation of the use of the plugin, a real project being developed under [24] and [1] was analysed.

Understand has been selected as the external tool to validate the analyser module that integrates different sources of analysis results. Understand is a well-known static analysis tool that offers rich information to the verification engineers with respect to the characteristics of the source code of a given project that are not given by SonarQube. This difference is particularly evident for specific programming languages such as C, as Understand provides analysis for C language whereas there are hardly open analysis functions for SonarQube in C and C++ language. Some of the rich set of metrics provided by Understand for a C project are: max/min/average complexity of a function, number of nested loops, or average number of function parameters, among others.

The first step to integrate Understand analysis into the SonarQube framework was to develop an extension of the Understand metrics to provide the required files for Sonar. The output file of Understand has a structure similar to the following one (although with more information):

```
Results     Entity  Line    Column  Check
Number of Results: N
Project Violation
Max complexity: (Value)
Average complexity: (Value)
Min complexity: (Value)
Max nesting: (Value)
Max number of lines: (Value)
Average nesting: (Value)
Min nesting: (Value)
Min number of lines: (Value)
Max number of parameters: (Value)
Average lines: (Value)
Average parameters: (Value)
Min number of parameters: (Value)
```

In that structure, it is possible to see all the metrics that the integrated analyser module is able to detect, although it is not necessary to include every one, but only the necessary ones. The text marked as *Free text to include comments* will not be analysed by the integrated analyser module. The order in which the metrics appear is not important, but the name and parenthesis with the metric value is. In this specific project validation example, it is suggested that, for example, although the average complexity of the source code is not too high, there is at least one function whose complexity is to high and needs to be reviewed. The rest results can be used to evaluate if the project source code characteristics comply to the coding norms and specific project standards. Following, an example of the presentation of a real source code coding rules is shown in figure 3. The number of rule violations in the source codes very high according to this specific project, with 4.2% being either blocker or critical, that indicates that they should be fixed; the estimated technical debt associated to solving these issues is 997 days.



Debt
997d ↗

Violations
43,001 ↗

Blocker	1,134	↗
Critical	699	
Major	20,235	↗
Minor	19,069	↘
Info	1,864	

Fig. 3 Display of results of source code rules analysis

6 Conclusion

The paper has presented the design and implementation of additional functionality to SonarQube framework regarding the analysis of coding rule violations in C source code. It has been implemented in practice to cover the need to collect the new metrics introduced by a powerful static source code analysis tool as Understand. Its execution inside the SonarQube framework results in the integration and connection of both tools that imporves the capabilities of both of them, yielding a single colloborative remote working space that supports the interaction of verification teams working over specific projects.

The inclusion of the metrics information in the SonarQube platform helps in the managing of the technical quality information of a project, leads to a more efficient project management and clarity of the information. The indication of the technical debt is a key indicator to project leaders of the estimated needed effort, which allows to organize human effort to improve software quality and fix all identified issues. It has been validated in a real project that requires compliance with norms related to the development of critical software (like DO-178C [24] and software quality, like AQAP-2210 [1].

Acknowledgement This work has been partly funded by the project REM4VSS (TIN2011-28339) funded by the Spanish Ministry of Economy and Competitiveness, and UC3M & Indra Sistemas S.A. research project under contract no. 2014/00476/001.

References

1. AQAP: AQAP 2110. NATO Quality Assurance Requirements for Design, Development and Production. 2nd edition (2006)
2. Balachandran, V.: Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation. In: Proc. of International Conference on Software Engineering – ICSE (2013)
3. Bersani, M., García Valls, M.: The cost of formal verification in adaptive CPS. An example of a virtualized server node. In: 17th IEEE International Symposium on High Assurance Systems Engineering – HASE (2016)
4. Campbell, G.A., Papapetrou, P.P.: SonarQube in Action. Manning Publications (2013) ISBN-9781617290954
5. CMMI Product Team. CMMI for Development v1.3. Improving processes for developing better products and services. CMU/SEI-2010-TR-033 (2010)
6. Coleman, D., Ash, D., Lowther, B., Oman, P.: Using metrics to evaluate software system maintainability. IEEE Computer **27**(8), 44–49 (2002)
7. Duvall, P.M., Matyas, S., Glover, A.: Continuous integration: improving software quality and reducing risk. Pearson Education (2007) ISBN 13: 978-0-321-33638-5
8. Escribano-Barreno, J., García-Valls, M.: Supporting the monitoring of the verification process of critical systems' software (2015). CoRR/Abs, arXiv:1512.04782
9. Escribano-Barreno, J., García-Muñoz, J., García-Valls, M.: Integrated metrics handling in open source software quality management platforms. ITNG (2016)
10. Fenton, N., Bieman, J.: Software metrics: a rigorous and practical approach. CRC Press (2014)
11. García-Valls, M., Cucinotta, T., Lu, C.: Challenges in real-time virtualization and predictable cloud computing. Journal of Systems Architecture **60**(2), 726–740 (2014)
12. García-Valls, M., Fernández-Villar, L., Rodríguez-López, I.: iLAND An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems. IEEE Transactions on Industrial Informatics **9**(1), 228–236 (2013)
13. García-Valls, M.S.: Calidad de servicio en sistemas multimedia empotrados mediante gestión dinámica de recursos. Universidad Politécnica de Madrid (2001)
14. García Valls, M., Baldoni, R.: Adaptive middleware design for CPS: Considerations on the OS, resource managers, and the network run-time. In: Proc. 14th Workshop on Adaptive and Reflective Middleware (ARM@Middleware) (2015)
15. García-Valls, M., et al.: Time-Sensitive Adaptation in CPS through Run-Time Configuration Generation and Verification. COMPSAC (2014)
16. Cappa-Banda, L., García-Valls, M.: Experimenting with a load-aware communication middleware for CPS domain. ITNG(2016)
17. Jenkins (2015). http://jenkins-ci.org/
18. Krutchen, P.: Contextualizing agile software development. Journal of Software: Evolution and Process **25**, 351–361 (2013)
19. LDRA (2015). http://www.ldra.com/
20. Maven (2015). http://maven.apache.org/
21. di Ruscio, D., Pelliccione, P.: A model-driven approach to detect faults in FOSS systems. Journal of Software: Evolution and Process **27**(4), 294–318 (2015)
22. PC-Lint (2015). http://www.gimpel.com/
23. PMD (2015). http://pmd.sourceforge.net/
24. RTCA Inc.: Software Considerations in Airborne Systems and Equipment Certification. RTCA Inc. DO-178C. 12/13/2011 (2011)
25. Scitools Understand (2015). https://scitools.com/
26. SonarQube (2015). http://www.sonarqube.org/
27. Splint (2015). http://www.splint.org/

Introducing Dynamic Argumentation to UbiGDSS

João Carneiro, Diogo Martinho, Goreti Marreiros and Paulo Novais

Abstract Supporting and representing the group decision-making process is a complex task that requires very specific characteristics. The current existing argumentation models cannot make good use of all the advantages inherent to group decision-making. There is no monitoring of the process or the possibility to provide dynamism to it. These issues can compromise the success of Group Decision Support Systems if those systems are not able to provide freedom and all necessary mechanisms to the decision-maker. We investigate the use of argumentation in a completely new perspective that will allow for a mutual understanding between agents and decision-makers. Besides this, our proposal allows to define an agent not only according to the preferences of the decision-maker but also according to his interests towards the decision-making process. We show that our definition respects the requirements that are essential for groups to interact without limitations and that can take advantage of those interactions to create valuable knowledge to support more and better.

Keywords Argumentation · Automatic negotiation · Multi-agent systems

1 Introduction

The UbiGDSS support the decision-making process by using the main characteristics of ubiquity ("anytime and "anywhere") [1, 2]. In order to support the decision-making process in an ubiquitous context it makes sense that UbiGDSS should: allow automatic negotiation, represent interests of decision-makers, allow the

J. Carneiro(✉) · D. Martinho · G. Marreiros
GECAD, Institute of Engineering – Polytechnic of Porto, Porto, Portugal
e-mail: {jomrc,1090557,mgt}@isep.ipp.pt

J. Carneiro · P. Novais
ALGORITMI Centre, University of Minho, Braga, Portugal
e-mail: pjon@di.uminho.pt

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_51

existence of a process, generate ideas, discuss points of view, etc [3, 4]. However, something is wrong with GDSS that we know today. Everyone has acknowledged the benefit of this type of systems; however, we do not actually see these systems being used in reality and it is not because the concept is still fresh. In fact, it is impossible to identify the reasons that lead to their absence and why they are not accepted in the industry sector nowadays. On the other hand, what we know is that most of artificial intelligence techniques proposed in the literature that could be used in UbiGDSS go against what are considered to be the benefits of group decision-making [5-7].

In this work, we propose a refreshing look into the concept of what and how should the artificial intelligence mechanism that composes an UbiGDSS be. For that, we introduce a dynamic argumentation framework, which provides the system with the features that are necessary for decision-makers to enjoy the benefits of group decision-making. Our proposal intends to follow decision-makers throughout the entire process. Our approach allows a group of decision-makers, where each agent represents a decision-maker, to seek a possible solution to a problem (choosing between several alternatives) while taking into account all the preferences of the decision-makers. Besides this, and considering that the decision-makers can understand the conversation performed between agents, those agents will also be able understand the new arguments that are created and exchanged between decision-makers. These new arguments can be processed and used by agents not only to advice decision-makers, but also to find solutions throughout the decision-making process.

The rest of the paper is organized as follows: in the Section 2 our approach is presented and the definition of the argumentation model is described. In Section 3, we present some conclusions and the work to be done hereafter.

2 The Argumentation Model

In this paper, we consider the following structure of a decision problem: there are a set of possible alternatives A, a set of criteria C, and a set of agents Ag, such that an alternative $a \in A$ has a value for all the defined criteria C. The decision problem has a defined communication language $\mathcal{L}_c$ which allow agents Ag to communicate. In order to operate with the defined $\mathcal{L}_c$, there is a set of algorithms $\mathcal{L}_a$, which specify for each locution $\varphi \in \mathcal{L}_c$ its effect. The relations between alternatives, criteria, agents, communication language and algorithms jointly form a decision system, represented as follows:

Definition 1: A decision system $(C, A, Ag, \mathcal{L}_c, \mathcal{L}_a)$, consists of:

- a set of criteria $C = \{c_1, c_2, \dots, c_n\}, n > 0$;
- a set of alternatives $A = \{a_1, a_2, \dots, a_m\}, m > 0$;
- a set of agents $Ag = \{ag_1, ag_2, \dots, ag_k\}, k > 0$;
- a communication language $\mathcal{L}_c$, consisting of a set of all locutions;

— a set of algorithms working as regulation $\mathcal{L}_a$ for $\mathcal{L}_c$, specifying for each locution $\varphi \in \mathcal{L}_c$ its effects.

Rule 1: Each alternative is related with each criterion. There cannot be an existing alternative with values for criteria that is not considered in the problem.

Definition 2: A criterion $c_i = \{id_{c_i}, v_{c_i}, m_{c_i}\}$ consists of:

— $\forall\, c_i \in C, i \in \{1, 2, \dots, n\}$;
— id_{c_i} is the identification of a particular criterion;
— v_{c_i} is the value of a particular criterion (Numeric, Boolean or Classificatory);
— m_{c_i} is the greatness associated with the criterion (Maximization, Minimization, Positivity, Negativity and Without Value).

Definition 3: An alternative $a_i = \{id_{a_i}, \left[c_{1_{a_i}}, c_{2_{a_1}}, \dots, c_{n_{a_i}}\right]\}$ consists of:

— $\forall\, a_i \in A, i \in \{1, 2, \dots, n\}$;
— id_{a_i} is the identification of a particular alternative;
— $[c_{1_{a_i}}, c_{2_{a_i}}, \dots, c_{n_{a_i}}]$ is the instantiation of each criterion.

The way each criterion is defined, allows an agent to know (in the previous example) that $c_{1_{a_1}} > c_{1_{a_2}} \wedge c_{1_{a_1}} > c_{1_{a_3}} \wedge c_{1_{a_2}} > c_{1_{a_3}}$, and $c_{2_{a_1}} \neq c_{2_{a_2}} \wedge c_{2_{a_1}} \neq c_{2_{a_3}} \wedge c_{2_{a_2}} = c_{2_{a_3}}$ and $c_{3_{a_2}} > c_{3_{a_1}} \wedge c_{3_{a_2}} > c_{3_{a_3}} \wedge c_{3_{a_1}} = c_{3_{a_3}}$.

An agent has a special structure that allows him to act according to the interests of the decision-maker he represents. Besides the agent's identification code, the decision-maker can also define the agent's style of behaviour for a certain time interval. The decision-maker may change that style of behaviour whenever he thinks to be appropriate. The proposed styles of behaviour (previously defined in [8]) allow the agent to act according to 4 dimensions: activity, resistance to change, concern for other and concern for self. Agents also include a protocol where it is defined a set of locutions available to ag_i in a time instant of t. An agent also holds the information about the evaluation done by the decision-maker about the preference for each alternative and the importance given to each considered criterion.

Definition 4: An agent $ag_i = \{id_{ag_i}, uid_{ag_i}, \beta_{ag_i}, Pr_{ag_i}, C_{ag_i}, A_{ag_i}, O_{ag_i}, K_{ag_i}\}$ consists of:

— $\forall\, ag_i \in Ag, i \in \{1, 2, \dots, n\}$;
— id_{ag_i} is the identification of a particular agent;
— uid_{ag_i} is the identification of the decision-maker represented by the agent ag_i;
— β_{ag_i} is the agent's behaviour (Dominating, Compromising, Obliging, Integrating, Avoiding and No Style);
— Pr_{ag_i} is the agent's protocol for $\mathcal{L}_c$, specifying the 'legal' moves at each instant. A protocol on $\mathcal{L}_c$ is a set of locution available to ag_i, where $Pr_{ag_i} \subseteq \mathcal{L}_c$;

- C_{ag_i} is the agent's evaluation of each criterion, $C_{ag_i} = \{E_{c_1}, E_{c_2}, \dots, E_{c_n}\}$, $Ev_{c_i} \in \{[0,1], \bot\}$;
- A_{ag_i} is the agent's evaluation of each alternative, $A_{ag_i} = \{E_{a_1}, E_{a_2}, \dots, E_{a_n}\}$, $Ev_{a_i} \in \{[0,1], \bot\}$;
- O_{ag_i} is the set of agent's objectives, $O_{ag_i} \subseteq A \cup C$, $preference\ relation \geq on\ the\ set\ O_{ag_i}$;
- K_{ag_i} is the agent's knowledge, where he can access the list of all sent and received messages, as well as the preferences of other agents, according to the knowledge he possess in a certain time instant of t.

Definition 5: A behaviour $\beta_i = \{Rc_{\beta_i}, Al_{\beta_i}, Cs_{\beta_i}, Co_{\beta_i}\}$ consists of (according to [8]):

- Rc_{β_i} is the agent's resistance to change dimension value;
- Al_{β_i} is agent's activity level dimension value;
- Cs_{β_i} is the agent's concern for self dimension value;
- Co_{β_i} is the agent's concern for others dimension value.

Definition 6: A locution $\varphi_i = \{id_{\varphi_i}, Tp_{\varphi_i}, Tx_{\varphi_i}, Ct_{\varphi_i}, Vr_{\varphi_i}, Dr_{\varphi_i}, Dm_{\varphi_i}, Av_{\varphi_i}\}$ consists of:

- $i \in \{1, 2, \dots, n\}$;
- id_{φ_i} is the locution's id (unique);
- Tp_{φ_i} is the locution's type (Question, Statement and Request);
- Tx_{φ_i} is the text associated to the locution;
- Ct_{φ_i} is the locution's context (Alternative, Criterion or Without Context);
- Vr_{φ_i} is the set of variables associated to the locution (Alternative or Criterion);
- Dr_{φ_i} is the direction associated to the locution (infavor, against, null);
- Dm_{φ_i} is the locution's domain (General or Specific);
- Av_{φ_i} is the locution's state (Available or Not Available).

Rule 2: Whenever a locution is added to Pr_{ag_i} in the time instant t its state will be $Av_{\varphi_i}{}^{t}, Av = Available.$

Rule 3: Whenever a locution is used at a time instant t its state will change to $Av_{\varphi_i}{}^{t}, Av = Not\ Available.$

Rule 4: Whenever a locution $Dm_{\varphi_i}, Dm = Specific$ is added to Pr_{ag_k} at the time instant t, then $\forall Dm_{\varphi_j} \in \mathcal{L}_{Dm} \subset Pr_{ag_k}, Dm = General$ and its state will be $Av_{\varphi_j}{}^{t}, Av = Available.$

Rule 5: For any locution $\varphi_j \in \mathcal{L}_{spc_i} \wedge c_i \subset Vr_{\varphi_j}$ there cannot be another locution $\varphi_k\ where\ c_i \subset Vr_{\varphi_k} \wedge \varphi_k \notin \mathcal{L}_{spc_i}.$

Rule 6: For any locution $\varphi_j \in \mathcal{L}_{sp_{a_i}} \land a_i \subset Vr_{\varphi_j}$ there cannot be another locution $\varphi_k\ where\ a_i \subset Vr_{\varphi_k} \land \varphi_k \notin \mathcal{L}_{sp_{a_i}}$.

Definition 7: A message $\psi_i = \left\{id_{\psi_i}, \varphi_{\psi_i}, tr_{\psi_i}, \alpha_{\psi_i}, en_{s_{\psi_i}}, En_{r_{\psi_i}}\right\}$ consists of:

- $i \in \{1, 2, \dots, n\}$;
- id_{ψ_i} is the conversation code;
- tr_{ψ_i} is the target associated with the message (can be null or be another message);
- φ_{ψ_i} is the locution sent in the message;
- α_{ψ_i} is the justification associated to the locution (can be an argument or can be null);
- $en_{s_{\psi_i}}$ is the agent/user who sent the message;
- $En_{r_{\psi_i}}$ is the set of agents/users who will receive the message (can be 1 or *).

Rule 7: For any message ψ created by a decision-maker $Dr_{\varphi_\psi}, Dr \neq null$. This means that the message's locution can only be either infavor or against Vr_{φ_ψ}.

Definition 8: An argument $\alpha_i = \left\{id_{\alpha_i}, Tx_{\alpha_i}, Vr_{\alpha_i}\right\}$ consists of:

- $i \in \{1, 2, \dots, n\}$;
- id_{α_i} is the identification of a particular argument;
- Tx_{α_i} is the text associated to a particular argument;
- Vr_{α_i} is the set of variables associated to a particular argument (can contain alternatives and criteria).

Up until now we have presented the definition for the proposed argumentation model only considering the agents' point of view. From this point forward all the definitions presented will be directed towards to the interactions between decision-makers. However these definitions are an extent of what has been proposed so far which will allow both agents and humans to use the same model definition.

Decision-makers, compared to agents, can also create messages, and those messages will hold the new knowledge as well as a new set of dynamic arguments. Our approach allows decision-makers to create messages that include arguments that are infavor or against. Each created message may lead to $n - 1$ messages, where n is the number of decision-makers involved in the process. Decision-makers can argue against a message through the use of an attack or support and argue in favour of a message through the use of reinforcement. The way the model is defined allows the message content to be of any sort of format like text or voice, since that information is irrelevant to the agent. A decision-maker may evaluate messages sent by other decision-makers, and this will allow agents to understand human interactions and the impact of every conversation.

Example 1 (message pro): $Dr_{\varphi_{\psi}}, Dr = infavor$. That means something positive related to $Vr_{\varphi_{\psi}}$. Such message is said pro the $Vr_{\varphi_{\psi}}$. For our previous example a message ψ pro could have $Vr_{\varphi_{\psi}}, Vr = a_1 \wedge Tx_{\alpha_{\psi}}, Tx =$ ''1st maintenance service is free''.

Example 2 (message cons): $Dr_{\varphi_{\psi}}, Dr = agaist$. That means something negative related to $Vr_{\varphi_{\psi}}$. Such message is said cons the $Vr_{\varphi_{\psi}}$. For our previous example a message ψ cons could have $Vr_{\varphi_{\psi}}, Vr = c_3, c_{3_a} = no \wedge Tx_{\alpha_{\psi}}, Tx =$ "The high temperatures in our area will damage the product in a car without air conditioning".

Definition 9: Let $\Psi = \{\psi_1, \psi_2, \ldots, \psi_n\}, n > 0$ denote a finite set of n messages that are exchanged during a human discussion.

Let us now define two functions that relate the messages in favour or against an $a_i \in A$, $c_j \in C$, $A_k \subseteq A$ or $C_l \subseteq C$ (Let us consider $X \asymp a_i \vee c_j \vee A_k \vee C_l$):

- $F_{infavor}: X \rightarrow \forall \psi \in \Psi, Dr_{\varphi_{\psi}} = infavor$, is a function that returns the messages in favour of X. Such messages are said pro the X;
- $F_{against}: X \rightarrow \forall \psi \in \Psi, Dr_{\varphi_{\psi}} = against$, is a function that returns the messages against X. Such messages are said cons the X.

Rule 8: A message is either in favour of the X or against the X. It cannot be both, so: $x \in X, \forall \psi \in \Psi s.t. \neg(\psi \in F_{infavor}(x) \wedge \psi \in F_{against}(x))$.

Definition 10: A message evaluation $\xi = \{en_{\xi_i}, \psi_{\xi_i}, ev_{\xi_i}\}$ consists of:

- $i \in \{1, 2, \ldots, n\}$;
- en_{ξ_i} is the user who performed the evaluation;
- ψ_{ξ_i} is the message being evaluated;
- ev_{ξ_i} is the evaluation value [-1, 1].

Definition 11: Let $\Xi = \{\xi_1, \xi_2, \ldots, \xi_n\}$ denote a finite set of n evaluations that are made during a human discussion.

Let us now define two functions that return all the evaluations approving or disapproving a message ψ_i:

- $F_{approval}: \psi_i \rightarrow \forall \xi \in \Xi, \psi_{\xi} = \psi_i \wedge ev_{\xi_i} > 0$, is a function that returns the messages approving ψ_i;
- $F_{disapproval}: \psi_i \rightarrow \forall \xi \in \Xi, \psi_{\xi} = \psi_i \wedge ev_{\xi_i} < 0$, is a function that returns the messages disapproving ψ_i.

Let us now define two functions that relate a message ψ_1 to the messages reinforcing it and to the messages attacking it:

- $F_{reinforcement}: \psi_1 \rightarrow \forall\psi \in \Psi, tr_\psi = \psi_1 \cap F_{approval}(\psi_1)$, is a function that returns the messages reinforcing ψ_1. Such messages are said pro the message ψ_1;
- $F_{attack}: \psi_1 \rightarrow \forall\psi \in \Psi, tr_\psi = \psi_1 \cap F_{disapproval}(\psi_1)$, is a function that returns the messages attacking ψ_1. Such messages are said cons the message ψ_1.

Let us define a function which returns the messages sent by en_1:

- $F_{sentby}: en_1 \rightarrow \forall\psi \in \Psi, en_\psi = en_1$, is a function that returns the messages sent by en_1.

Let us define a function that returns all the evaluation done by en_1:

- $F_{evaluatedby}: en_1 \rightarrow \forall\xi \in \Xi, en_\xi = en_1$, is a function that returns the evaluations done by en_1.

Let us now define a function to identify the lasts messages sent during a human discussion:

- $F_{last}: \Psi \rightarrow \forall\psi \in \Psi, F_{reinforcement}(\psi) \cup F_{attack}(\psi) = \emptyset$, is a function that returns the messages that are not attacked or reinforced thus being the last messages exchanged in a human discussion.

Let us now define a function that returns all the messages done until message ψ_1 which can be more than one reinforcement or just only one attack:

- $F_{reinforcementrow}: \psi_1 \rightarrow \forall\psi \in \Psi, \begin{cases} F_{reinforcementrow}(\psi), \psi \in F_{reinforcement}(\psi_1) \\ \psi_1, \psi \in F_{attack}(\psi_1) \\ \psi_1, \psi \notin F_{attack}(\psi_1) \vee F_{reinforcement}(\psi_1) \end{cases}$, is a function that returns all the reinforcement or the attack done until ψ_1.

Let us now define a function that returns all the messages prior to each last message in a human discussion that are either reinforcements (in the limit it could be all reinforcements until the initial message) or an attack:

- $F_{lastdiscussions}: \Psi \rightarrow \forall\psi \in F_{last}, F_{reinforcementrow}(\psi)$, is a function that returns all the messages prior to each last message that are either reinforcements or an attack.

Let us define a function that returns all the last evaluations done by en_1:

- $F_{lastevaluatedby}: en_1 \rightarrow \forall\xi \in F_{evaluatedby}(en_1), \psi_\xi \in F_{lastdiscussions}$, is a function that returns all the last evaluations done by en_1.

In order to identify all the last evaluations done by en_1 in a certain dialogue, the id_ψ can be used to filter those evaluations, and therefore the function which returns all the last evaluations done by en_1 in a dialogue with id_{ψ_1} would be:

- $F_{lastevaluationsin}: id_{\psi_1} \rightarrow \forall \xi \in F_{lastevaluatedby}(en_1), id_{\psi_\xi} = id_{\psi_1}$, is a function that returns all the last evaluations done by en_1 for the human dialogue with the id id_{ψ_1}.

In [9] the researchers state that argumentation-based decision process can be decomposed into the following steps: (1) Constructing arguments in favour/against statements (beliefs or decisions), (2) Evaluating the strength of each argument, (3) Determining the different conflicts among arguments, (4) Evaluating the acceptability of arguments and (5) Comparing decisions on the basis of relevant "accepted" arguments. We consider that in our work, we deal with the first 3 points, with the advantage of integrating those points in a perspective that deals with both agents and humans using the same definition. Points 4 and 5 frame what we intend to do as future work and that is described in Section 4.

3 Conclusions and Future Work

In this work, we introduced the possibility to create dynamic arguments through a definition of an argumentation model. With our proposal it is possible (and motivates) the interaction between decision-makers, allowing a decision-maker to create new arguments (in favour or against) and also reinforce or attack other arguments created by other decision-makers. The way decision-makers can evaluate each argument will allow the agents to understand the impact of the interactions for the decision-maker they represent and not the content of the conversation (which could be text, sound, etc.). Besides this, since agents share the same problem definition, it allows decision-makers to understand all the interactions between those agents and the reason why they suggest a certain solution for the problem.

As future work, we intend to design all the necessary algorithms so that our proposal can be implemented. One major goal is to work in an algorithm that allows the agent to understand each message decision tree that is associated with each interaction between decision-makers. Besides this, the agent should also take his defined behaviour into account when doing any analysis in order to properly support and defend the perspective of the decision-maker he represents. Finally we intend to work on the branch of how and what type of information should be presented to the decision-maker, through what will call as intelligent reports.

Acknowledgements This work has been supported by COMPETE Programme (operational programme for competitiveness) within project POCI-01-0145-FEDER-007043, by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foun-dation for Science and Technology) within the Projects UID/CEC/00319/2013, UID/EEA/00760/2013, and the João Carneiro PhD grant with the reference SFRH/BD/89697/2012 and by Project MANTIS - Cyber Physical System Based Proactive Collaborative Maintenance (ECSEL JU Grant nr. 662189).

References

1. Kwon, O., Yoo, K., Suh, E.: UbiDSS: a proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. Expert Systems with Applications **28**, 149–161 (2005)
2. Daume, S., Robertson, D.: An architecture for the deployment of mobile decision support systems. Expert Systems with Applications **19**, 305–318 (2000)
3. Carneiro, J., Santos, R., Marreiros, G., Novais, P.: UbiGDSS: A Theoretical Model to Predict Decision-Makers' Satisfaction. International Journal of Multimedia and Ubiquitous Engineering **10**, 191–200 (2015)
4. Carneiro, J., Santos, R., Marreiros, G., Novais, P.: Overcoming the Lack of Human-Interaction in Ubiquitous Group Decision Support Systems (2014)
5. Dennis, A.R.: Information exchange and use in small group decision making. Small Group Research **27**, 532–550 (1996)
6. Huber, G.P.: Issues in the design of group decision support sytems. MIS quarterly, 195–204 (1984)
7. Hill, G.W.: Group versus individual performance: Are N+ 1 heads better than one? Psychological Bulletin **91**, 517 (1982)
8. Martinho, D., Carneiro, J., Marreiros, G., Novais, P.: Dealing with agents' behaviour in the decision-making process. In: Workshop Proceedings of the 11th International Conference on Intelligent Environments, p. 4. IOS Press (2015)
9. Rahwan, I., Simari, G.R., van Benthem, J.: Argumentation in Artificial Intelligence. Springer (2009)

Part II
Special Session on AI–driven Methods for Multimodal Networks and Processes Modeling

Modelling and Performance Evaluation of Fractal Topology Streets Network

Grzegorz Bocewicz, Andrzej Jardzioch and Zbigniew Banaszak

Abstract The paper introduces the concept of a Fractal Topology Streets Network (FTSN) in which different transportation modes interact each other via distinguished subsets of common shared hubs as to provide a variety of mass customized passenger services. The network of repetitively acting local transportation modes encompassing FTSN's fractal structure provides a framework for passengers' origin-destination trip routing and scheduling. The passenger travel schedules can be estimated easily while taking into account cyclic behaviour of both: local transportation modes and the whole transportation network. In that context the paper's objective concerns of assessment of a Fractal-like Network of periodically acting Local Transportation Modes (FNLTM) infrastructure from the perspective of possible mass-customized oriented requirements.

Keywords Fractal network · Multimodal processes · Cyclic scheduling

1 Introduction

Numerous road network patterns deployed in cities range from the tightly structured fractal network with perpendicular roads in a regular raster pattern to the hierarchical network with sprawling secondary and tertiary roads feeding into arterial roads in a branch like system [4, 6]. In that context, the concept of a fractal-like multimodal transportation network in which arbitrarily chosen parts

G. Bocewicz(✉) · Z. Banaszak
Department of Electronics and Computer Science, Koszalin University of Technology,
Sniadeckich 2, Koszalin, Poland
e-mail: bocewicz@ie.tu.koszalin.pl

A. Jardzioch
Department of Mechanical Engineering, West Pomeranian University of Technology,
Piastów Ave 19, Szczecin, Poland
e-mail: andrzej.jardzioch@zut.edu.pl

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_52

supported by different transportation modes interact each other via distinguished subsets of common shared sites (e.g. bus stations) as to provide a variety of transportation/handling services. Regular structure (e.g. fractal topology networks) are found in different application domains, such as multimodal passenger transport network combining several unimodal networks (bus, tram, metro, etc.) as well as service domains (including passenger/cargo transportation systems, e.g. ferry, automated guided vehicle (AGV), train networks) [1, 2]. An example of city districts growth and its corresponding fractal representation provides a real-life example of multimodal transportation network, see Fig.1 (in subsequent iterations the single edges and then two split edges are added alternately).

Passenger flows following assumed travel destinations transit through different transportation modes. Multimodal processes [2, 3] executed in a FNLTM (Fractal-like Network of Local Transportation Modes, see Fig. 2), providing connection from origin to destination, can be seen as passengers and/or goods flows transferred through different modes to reach their destination. Problems arising in this kind of network concern of routing and scheduling of multimodal processes of passengers flows which are in general case also NP-hard. However, the passenger travel schedules can be estimated easily while taking into account cyclic behaviour of both: local transportation modes and the whole transportation network, i.e. assuming that period of multimodal processes depends on a period of the relevant FNLTM.

Many models and methods aimed at cyclic scheduling have been considered so far [5]. Among them, the mathematical programming approach [1, 11], max-plus algebra [7], constraint programming [2], Petri nets [10] frameworks belong to the more frequently used. Most of them are oriented at finding of a minimal cycle or maximal throughput while assuming deadlock-free processes flow. However, the approaches trying to estimate the cycle time from cyclic processes structure and the synchronization mechanism employed while taking into account deadlock phenomena are quite unique.

A new modelling framework enabling to evaluate a cyclic steady state of the given FNLTM encompassing possible transportation services (see Fig. 1a)) in the FTSN (Fractal Topology Streets Network) is our main contribution. In that context, the paper can be seen as a continuation of our former work [2, 3] regarding AGVs fleet scheduling in a regular, mesh-like layout of an FMS transportation routes.

The rest of the paper is organized as follows: Section 2 introduces to a concept of fractal structure of multimodal transportation network modelled in terms of Systems of Cyclic Concurrent Executed Processes (SCCP). Section 3 provides the declarative problem formulation focused at FNLTMs prototyping. Cyclic scheduling issues regarding cyclic steady state processes and conditions enabling match-up structures coupling as well as computational experiments are presented in Sections 4 and 5, respectively. Some directions of further research are discussed in Section 6.

2 Fractal Structure Networks

Note that Fig. 1b) shows results of the fourths, fifths, sixths and seventh iterations of FTSN's growth. The FTSN can be modelled in terms of FNLTM, i.e. can be seen as a network of bus/trams lines providing periodic service along cyclic routes. In turn, the FNLTM can be modelled in terms of a System of Concurrently executed Cyclic Processes (SCCP) as shown in Fig. 2b) [2, 3]. Consequently, each edge of FNLTM is replaced by a cyclic process executed along four resources. Two resources (distinguished by dots) correspond to FNLTM vertices and model interchange stations (hubs) shared by other cyclically operating lines. The remaining two, so called unshared, resources correspond to pavements travelled by considered transportation mode.

Among following patterns "┤", "├", "┬", "┴", "┼" (see Fig. 2 b)) composing FTSN let us distinguish "┼" as an Elementary Pattern Structure (EPS). In terms of SCCP representation, it will be called Elementary Isomorphic pattern Structure (EIS) ${}^{(i)}SC_1$ composed of four **local cyclic processes**, viz. P_1, P_2, P_3, P_4 (distinguished in Fig. 2 c)) by ${}^{(i)}P_1$, ${}^{(i)}P_2$, ${}^{(i)}P_3$, and ${}^{(i)}P_4$, respectively and where (i) denotes the EIS's index).

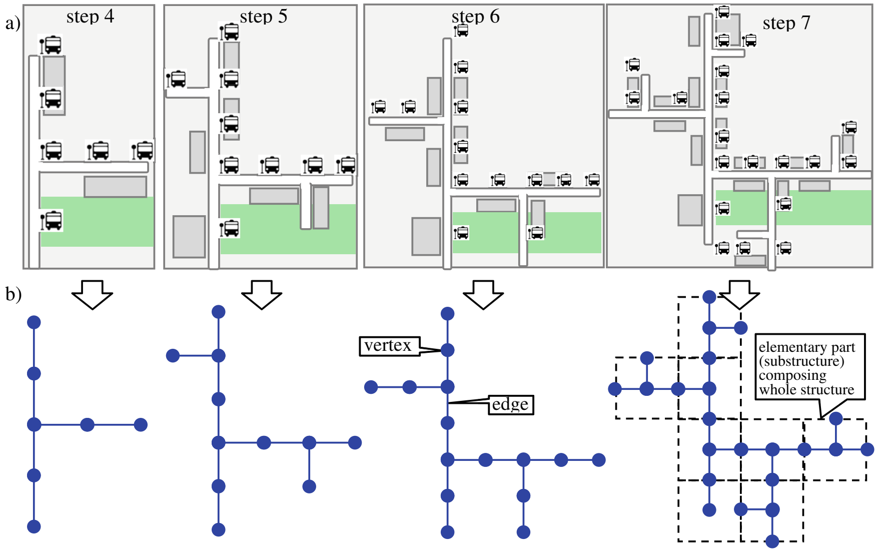


Fig. 1 Exemplary snapshots of city districts growth a) and corresponding to them fractal model representations b)

Due to the assumed SCCP model, the processes encompassing transportation modes follow the **routes** composed of arcs and nodes corresponding to interchange stations distinguished by the **set of resources** $R = \{R_1, \dots, R_c, \dots, R_{11}\}$, where R_c – the c-th resource (see ${}^{(i)}SC_1$ shown in Fig. 2c). Each P_i contains the set of **streams** $P_i = \left\{P_i^1, \dots, P_i^k, \dots, P_i^{ls(n)}\right\}$, where P_i^k – the k-th stream of the i-th local process P_i. In case considered: $P_1 = \{P_1^1\}$, $P_2 = \{P_2^1\}$, $P_3 = \{P_3^1\}$, $P_4 = \{P_4^1\}$.

Apart from local processes, we consider two **multimodal processes** (i.e. processes executed along the routes consisting parts of the routes of local processes): mP_1, mP_2. For example, the transportation route depicted by the red colour line corresponds to the multimodal process mP_1 supported by different transportation modes (bus and, tram lines), i.e. following P_2^1 and P_4^1.

Processes interact with each other through shared resources, e.g. hubs. A resource conflicts are resolved with the aid of a priority dispatching rules [2, 3], determining orders in which streams access shared resources. For instance, in the case of the shared resource R_1 (see Fig. 2 c)), the priority dispatching rule $\sigma_1^0 = (P_1^1, P_2^1, P_3^1, P_4^1)$, determines the order in which streams of local processes can access R_1. In general, the following notation is used:

- $p_i^k = (p_{i,1}^k, \dots, p_{i,j}^k, \dots, p_{i,lr(i)}^k)$ **the route of the stream of the local process** $\boldsymbol{P_i^k}$ (the k-th stream of the i-th local process P_i), where $o_{i,j}^k$ - **the** $\boldsymbol{j}$**-th operation executed in the stream** $\boldsymbol{P_i^k}$; $lr(i)$ - the length of the cyclic process route,
- $x_{i,j}^k(l) \in \mathbb{N}$ - the moment of operation beginning $o_{i,j}^k$ in the l-th cycle,
- $t_i^k = (t_{i,1}^k, t_{i,2}^k, \dots, t_{i,j}^k, \dots, t_{i,lr(i)}^k)$ specifies **the operation times of local processes**, where $t_{i,j}^k$ denotes the time of execution of operation $o_{i,j}^k$,
- $mp_i^k = \left(mp_{i,1}^k, \dots, mp_{i,j}^k, \dots, mp_{i,lm(i)}^k\right)$ specifies **the route of the stream** $\boldsymbol{mP_i^k}$ **from the multimodal process** $\boldsymbol{mP_i}$ (the k-th stream of the i-th multimodal process mP_i), e.g.: $p_1^1 = (R_1, R_3, R_{10}, R_2)$ - see ${}^{(i)}SC_1$ in Fig. 2c, where $mo_{i,j}^k$ - **the** $\boldsymbol{j}$**-th operation executed in the stream** $\boldsymbol{mP_i^k}$
- $mx_{i,j}^k(l) \in \mathbb{N}$ - the moment of operation beginning $mo_{i,j}^k$ in the l-th cycle,
- $mt_i^k = (mt_{i,1}^k, mt_{i,2}^k, \dots, mt_{i,j}^k, \dots, mt_{i,ldm(i)}^k)$ specifies **the operation times of multimodal processes**, $mt_{i,j}^k$ denotes the time of execution of operation $mo_{i,j}^k$,
- $\Theta = \{\Theta^0, \Theta^1\}$ is the set of **priority dispatching rules**, $\Theta^i = \left\{\sigma_1^i, \dots, \sigma_c^i, \dots, \sigma_m^i\right\}$.

Using the above notation, the SCCP's structure can be defined as the tuple:

$$SC = \big((R, SL), SM\big), \tag{1}$$

where: $R = \{R_1, \dots, R_c, \dots, R_m\}$ – the set of resources, $SL = (P, U, T, \Theta^0)$ – the structure of local processes of SCCP, i.e.: P – the set of local process, U – the set of routes of local process, T – the set of sequences of operation times in local processes, $\Theta^0 = \{\sigma_1^0, \dots, \sigma_c^0, \dots, \sigma_m^0\}$ – the set of priority dispatching rules for local processes. $SM = (MP, M, mT, \Theta^1)$ – the structure of multimodal processes of SCCP, i.e.: MP – the set of multimodal process, M – the set of routes of a multimodal process, mT – the set of sequences of operation times in multimodal processes, Θ^1– the set of priority dispatching rules for multimodal processes.

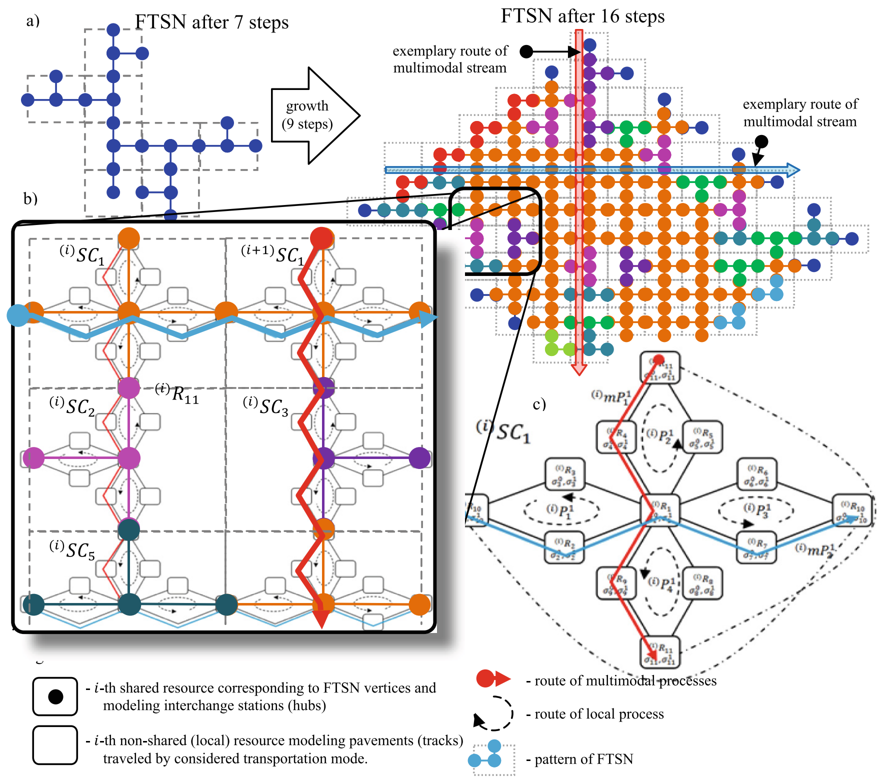

a)
FTSN after 7 steps
growth (9 steps)
FTSN after 16 steps
exemplary route of multimodal stream
exemplary route of multimodal stream
b)
$^{(i)}SC_1$
$^{(i+1)}SC_1$
$^{(i)}SC_2$
$^{(i)}R_{11}$
$^{(i)}SC_3$
$^{(i)}SC_5$
c)
$^{(i)}SC_1$
$^{(i)}mP_1^1$
$^{(i)}P_2^1$
$^{(i)}P_1^1$
$^{(i)}P_3^1$
$^{(i)}P_4^1$
$^{(i)}mP_2^1$
- i-th shared resource corresponding to FTSN vertices and modeling interchange stations (hubs)
- i-th non-shared (local) resource modeling pavements (tracks) traveled by considered transportation mode.
- route of multimodal processes
- route of local process
- pattern of FTSN

Fig. 2 FTSN obtained from network shown in Fig. 1 a) and its FNLTM-interpreted part represented in terms of SCCP model b), exemplary EISs c)

The behaviour of the structure SC (1) is characterized by the schedule (2):

$$X' = \big((X,\alpha),(mX,m\alpha)\big) \tag{2}$$

where: $X = \left\{x_{1,1}^1, \dots, x_{i,j}^k, \dots, x_{n,lr(n)}^{ls(n)}\right\}$ – the set of the moments of local processes operations beginning in $l = 0$ of the cycle, $x_{i,j}^k$ – determines the value: $x_{i,j}^k(l) = x_{i,j}^k + \alpha$, $\alpha -$ the period of local processes executions, $mX = \left\{mx_{1,1}^1, \dots, mx_{i,j}^k, \dots, mx_{w,lm(w)}^{lsm(w)}\right\}$ – the set of moments of operations of multimodal processes beginning in $l = 0$ of cycle, $mx_{i,j}^k$ – determines the value $mx_{i,j}^k(l) = mx_{i,j}^k + m\alpha \cdot l$, $m\alpha -$ the period of multimodal processes executions.

Consider two measures describing processes periodically executed in the SCCP:

- Tc_i / mTc_i – the **takt time** of the process P_i / mP_i, i.e. the time between two consecutive moments of streams completion in the process P_i / mP_i: $Tc_i = x_{i,j}^k(l) - x_{i,j}^{k-1}(l)$ / $mTc_i = mx_{i,j}^k(l) - mx_{i,j}^{k-1}(l)$.

- mFT_i – the **flow time** of the multimodal process mP_i, i.e. the time required for each stream of process mP_i completion: $mFT_i = mx^k_{i,lm(i)}(l) + mt^k_{i,ldm(i)} - mx^k_{i,1}(l) = m\alpha - \Delta F_i$, where: ΔF_i – the buffer time encompassing the difference between completion time of a travel and a beginning time of the subsequent one, e.g.: $\Delta F_i = mx^k_{i,1}(l+1) - \left(mx^k_{i,lm(i)}(l) + mt^k_{i,ldm(i)}\right)$.

3 Problem Formulation

Given a structure SC (1) with unknown dispatching rules Θ. The conditions guaranteeing the values Θ implying the cyclic schedule X' (2) there exist are sought. Note, that the structure SC (1) can be decomposed into a set of isomorphic elementary substructures $SC^* = \{SC_1, \ldots, SC_i, \ldots, SC_{lc}\}$. If, for any substructure SC_i, there is a subset of parameters Θ guaranteeing its cyclic behavior, then the considered above mentioned question boils down to the following one: **Does there exist a way of composition of SC^* guaranteeing its cyclic behavior represented by a cyclic schedule?**

In order to answer this question the **operator of substructures composition** $\oplus$ is introduced such that the result of composition of two substructures SC_a, SC_b through mutually shared resources results in $SC_a \oplus SC_b = SC_c$ [3].

4 Cyclic Scheduling

In terms of the introduced operator of substructures composition $\oplus$ the SC can be shown as a multiple composition:

$$SC = {}^{(1)}SC_1 \oplus \ldots \oplus {}^{(i)}SC_Z \oplus \ldots \oplus {}^{(q)}SC_w \tag{3}$$

where each substructure ${}^{(i)}SC_j$ is put together with the others by means of integrating the resources belonging to the same set of corresponding resources. For example, the structure ${}^{(i)}SC_1$ from Fig. 2b) is put together with ${}^{(i)}SC_2$ by the resource ${}^{(i)}R_{11}$. Due to the same manner of process execution the cyclic schedule of the whole structure can be perceived as a composition of corresponding cyclic schedules:

$$X' = {}^{(1)}X'_1 \mathbin{\dot\cup} \ldots \mathbin{\dot\cup} {}^{(i)}X'_Z \mathbin{\dot\cup} \ldots \mathbin{\dot\cup} {}^{(q)}X'_w \tag{4}$$

where: ${}^{(i)}X'_Z$ – the cyclic schedule of the substructure ${}^{(i)}SC_Z$:

$${}^{(i)}X'_Z = \left(\left({}^{(i)}X_Z,\ {}^{(i)}\alpha_Z\right), \left({}^{(i)}mX_Z,\ {}^{(i)}m\alpha_Z\right)\right) \tag{5}$$

${}^{(i)}X_Z$/ ${}^{(i)}mX_Z$ – the set of the initiation moments of local / multimodal process operations of ${}^{(i)}SC_Z$; ${}^{(i)}\alpha_Z$ / ${}^{(i)}m\alpha_Z$ – the period of local/multimodal processes executions;

$$ {}^{(a)}X' \uplus {}^{(b)}X' = \Big(\big({}^{(a)}X \cup {}^{(b)} X, lcm\big({}^{(a)}\alpha, {}^{(b)}\alpha\big)\big), \big({}^{(a)}mX \cup {}^{(b)} mX, \mathrm{c} \cdot lcm\big({}^{(a)}\alpha, {}^{(b)}\alpha\big)\big). \tag{6} $$

Assuming that each substructure ${}^{(i)}SC_j$ in considered SCCP's structure can be represented by ${}^{(i)}SC_1$ from Fig. 2c) (i.e., each ${}^{(i)}SC_j$ is a copy of ${}^{(i)}SC_1$) then the schedule X' (4) can be deduced as the following composition:

$$ X' = \biguplus_{i=1}^{lc}\big({}^{(i)}X'_1\big), \tag{7} $$

where: ${}^{(i)}X'_1$ – the cyclic schedule (5) of substructure ${}^{(i)}SC_1$, $\biguplus_{i=1}^{lc}\big({}^{(i)}X'\big) = {}^{(1)}X' \uplus \ldots \uplus {}^{(lc)}X'$ – composition of schedules ${}^{(i)}X'$.

In this case the periods $\alpha'/m\alpha'$, takt times $Tc_i'/\ mTc_i'$ and flow times mFT_i can be determined due to the following equations: $\alpha' = {}^{(i)}\alpha_1$; $m\alpha' = c \cdot {}^{(i)} m\alpha_1$; $Tc_i' = {}^{(i)}Tc_i$; $mTc_i' = {}^{(i)}mTc_i$; $mFT_i' = m\alpha' - \Delta F_i$, where: $c \in \mathbb{N}^+$, and $c \leq lc$.

In order to determine dispatching rules ${}^{(i)}\Theta_1$ of the substructure ${}^{(i)}SC_1$, (see Fig. 2c) that guarantee the attainability of the cyclic schedule ${}^{(i)}X'_1$, it is possible to apply the constraint satisfaction problem (8) [8,9]:

$$ PS'_{REX_i} = \big(\big(\{{}^{(i)}X'_1, {}^{(i)}\Theta_1, {}^{(i)}\alpha'_1\}, \{D_X, D_\Theta, D_\alpha\}\big), \{C_L, C_M, C_D\}\big) \tag{8} $$

where: ${}^{(i)}X'_1$, ${}^{(i)}\Theta_1$, ${}^{(i)}\alpha'_1$ – the decision variables, ${}^{(i)}X'_1$ – the cyclic schedule (5) of substructure ${}^{(i)}SC_1$, ${}^{(i)}\Theta_1 = \{{}^{(i)}\Theta_1^0, {}^{(i)}\Theta_1^1\}$ – the set of priority dispatching rules for ${}^{(i)}SC_1$, ${}^{(i)}\alpha'_1 = \big({}^{(i)}\alpha_1, {}^{(i)}m\alpha_1\big)$ – the period of local/multimodal processes for ${}^{(i)}SC_1$, D_X, D_Θ, D_α – the domains determining admissible value of decision variables, $\{C_L, C_M, C_D\}$ – the set of constraints C_L and C_M describing SCCP behavior [2], C_D – the constraints that guarantee the smooth execution of the stream on mutual resources.

The schedule ${}^{(i)}X'_1$ that meets all the constraints from the given set $\{C_L, C_M, C_D\}$ is the solution sought for the problem (8). The constraints C_L, C_M guarantee that in the substructure ${}^{(i)}SC_1$ from Fig. 2c) the processes are executed in a cyclic manner [2, 3], however, cannot ensure the lack of interferences between the operations of neighbouring substructures. In order to avoid such interferences additional constraints C_D following a principle of match-up structures coupling are introduced. As a consequence, the behaviour of the resulting substructure $SC_c = SC_a \oplus SC_b$ can be determined by the cyclic schedule $X_c' = X_a' \uplus X_b'$ (4), i.e. a composition of the schedules X_a', X_b', if: the value of the period of schedule X_a' is the least common multiple of the period of schedule X_b' and such that $m\alpha_a\ MOD\ m\alpha_b = 0$; and $\alpha_a\ MOD\ \alpha_b = 0$.

The sufficient conditions guaranteeing such a match-up coupling of two substructures $SC_c = SC_a \oplus SC_b$ of the known cyclic behaviours, while leading to the cyclic behaviour X_c' can be specified by:

Constraints for local process operations. The two operations $o_{i,j}^{h}$, $o_{q,r}^{s}$ do not interfere (on the mutually shared resource R_{k_i}) if the operation $o_{i,j}^{h}$ begins (moment $x_{i,j}^{h}$) after the release (with the delay Δt) of the resource by the operation $o_{q,r}^{s}$ (moment x_{q,r^*}^{s} of the subsequent operation initiation) and releases the resource (moment x_{i,j^*}^{h} of the subsequent operation initiation) before the beginning of the next execution of the operation $o_{q,r}^{s}$ (moment $x_{q,r}^{s}+\alpha_b$). The collision-free execution of the local process operations is possible if the following constraint holds:

$$\left[\left(x_{i,j}^{h} \geq x_{q,r^*}^{s} + k'' \cdot \alpha_b + \Delta t\right) \wedge \left(x_{i,j^*}^{h} + k' \cdot \alpha_a + \Delta t \leq x_{q,r}^{s} + \alpha_b\right)\right]$$
$$\vee \left[\left(x_{q,r}^{s} \geq x_{i,j^*}^{h} + k' \cdot \alpha_a + \Delta t\right) \wedge \left(x_{q,r^*}^{s} + k'' \cdot \alpha_b + \Delta t \leq x_{i,j}^{h} + \alpha_a\right)\right] \quad (9)$$

where: $j^* = (j+1)\ MOD\ lr(i),\ r^* = (r+1)\ MODlr(q)$,

$$k' = \begin{cases} 0 \text{ when } j+1 \leq lr(i) \\ 1 \text{ when } j+1 > lr(i) \end{cases}, \qquad k'' = \begin{cases} 0 \text{ when } r+1 \leq lr(q) \\ 1 \text{ when } r+1 > lr(q) \end{cases}.$$

Satisfying the constraint (10) means that on every mutually shared resource of the composed substructures SC_a, SC_b the local processes are executed alternately.

Constraints for multimodal processes. Two operations $mo_{i,j}^{h}$, $mo_{q,r}^{s}$ can be executed without any interferences on the mutually shared resource $R_{k_i} \in RK$ if one operation is executed between the subsequent executions of the other. In this context, the collision-free execution of the multimodal process operations is possible if:

$$\left[\left(mx_{i,j}^{h} \geq mx_{q,r^*}^{s} + k'' \cdot m\alpha_b + \Delta t\right) \wedge \left(mx_{i,j^*}^{h} + k' \cdot m\alpha_a + \Delta t \leq mx_{q,r}^{s} + m\alpha_b\right)\right]$$
$$\vee \left[\left(mx_{q,r}^{s} \geq mx_{i,j^*}^{h} + k' \cdot m\alpha_a + \Delta t\right) \wedge \left(mx_{q,r^*}^{s} + k'' \cdot m\alpha_b + \Delta t \leq mx_{i,j}^{h} + m\alpha_a\right)\right] \quad (10)$$

where: j^*, r^*, k' and k'' defined as in (11), $mx_{i,j}^{h}, mx_{q,r}^{s}$ – initiation moments of the operations $mo_{i,j}^{h}$, $mo_{q,r}^{s}$ of substructures SC_a, SC_b, respectively; $mx_{i,j^*}^{h}, mx_{q,r^*}^{s}$ – initiation moments of operations executed after $mo_{i,j}^{h}$, $mo_{q,r}^{s}$, respectively.

If these constraints are satisfied, the manner of executing operations on mutual resources determines the form of priority dispatching rules $\sigma_{k,c}^{0}$, as the sequence of operations resulting from the schedules X_a', X_b' satisfying the constraints (10) and (11).

5 Computational Experiments

The evaluation of the cyclic behaviour (the existence of the schedule X') of the fractal-like structure SC from Fig. 2 can be obtained as a result of evaluating the parameters of EIS ${}^{(i)}SC_1$ from Fig. 2c) (it is assumed that the all the transportation times are the same and equal to $t_{i,j}^{k} = 5$ u.t. (units of time)).

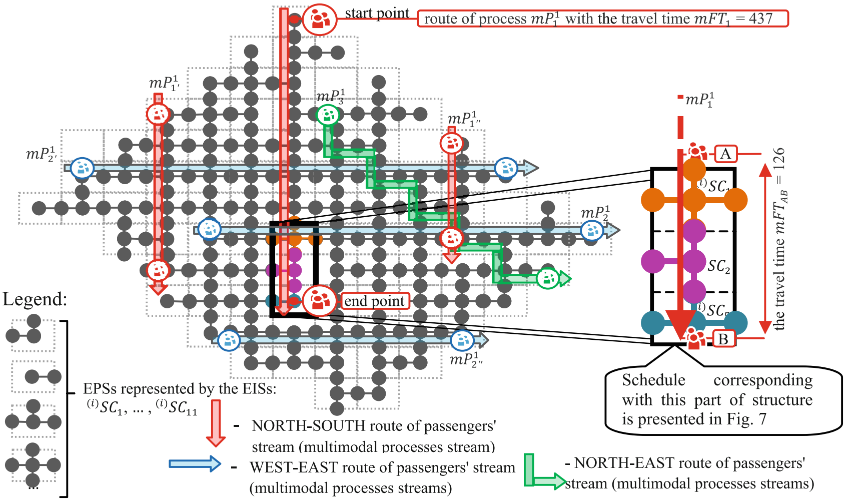

start point
route of process mP_1^1 with the travel time $mFT_1 = 437$
$mP_{1'}^1$
mP_3^1
$mP_{1''}^1$
$mP_{2'}^1$
mP_2^1
$mP_{2''}^1$
end point
mP_1^1
A
B
${}^{(i)}SC_1$
SC_2
${}^{(i)}SC_5$
the travel time $mFT_{AB} = 126$
Schedule corresponding with this part of structure is presented in Fig. 7
Legend:
EPSs represented by the EISs: ${}^{(i)}SC_1, \dots, {}^{(i)}SC_{11}$
- NORTH-SOUTH route of passengers' stream (multimodal processes stream)
- WEST-EAST route of passengers' stream (multimodal processes streams)
- NORTH-EAST route of passengers' stream (multimodal processes streams)

Fig. 3 FTSN driven scheduling of multi modal processes (see Fig. 2).

Consider FTSN shown in Fig. 3 (following Fig. 2) assuming 11 kinds of EPSs modelled in terms of SCCP by EIS s ${}^{(i)}SC_1$, ... , ${}^{(i)}SC_{11}$, and distinguishing seven passengers' route scenarios. The problem PS'_{REX_i} has been implemented and solved in the constraint programming environment OzMozart (CPU Intel Core 2 Duo 3GHz RAM 4 GB). The first acceptable solution was obtained in less than one second. The problem solution are the cyclic schedule ${}^{(i)}X'_1$ (with ods ${}^{(i)}\alpha_1 = 24$ and ${}^{(i)}m\alpha_1 = 48$ u.t) and the dispatching rules ${}^{(i)}\Theta$ shown in Tab. 1.

Table 1 The dispatching rules of SCCP following ${}^{(i)}SC_1$ from Fig. 2c)

dispatching rule for local processes				dispatching rule for multimodal processes	
${}^{(i)}\sigma_1^0$	$\left({}^{(i)}P_1^1, {}^{(i)}P_2^1, {}^{(i)}P_3^1, {}^{(i)}P_4^1\right)$			${}^{(i)}\sigma_1^1$	$\left({}^{(i)}mP_2^1, {}^{(i)}mP_1^1\right)$
${}^{(i)}\sigma_{10}^0$	$\left({}^{(i)}P_1^1, {}^{(i)}P_3^1\right)$	${}^{(i)}\sigma_{11}^0$	$\left({}^{(i)}P_2^1, {}^{(i)}P_4^1\right)$		

According to (7) the attained schedule is a component of the schedule X' that characterizes the behaviour of the whole structure SC. The part of schedule X' being a multiple composition of the schedules ${}^{(i)}X'_1$ is presented in Fig. 4. The schedule encompasses behaviour of a part of the network containing substructures ${}^{(i)}SC_1$ and ${}^{(i)}SC_2$ as well as ${}^{(i)}SC_5$ determining a part of multimodal process m^1P_1 (part of stream $m^1P_1^1$ - see Fig. 3). Referring to the network presented in Fig. 3, the obtained schedule illustrates operation of transportation modes (local processes) and a way they provide passengers' transportation route (multimodal processes) between points A and B (see Fig. 3). Therefore, since the periods of local processes are equal to $\alpha =^{(i)} \alpha_1 = 24$ u.t. thus the travel time between points A and B (i.e. mFT_1) the passengers have to spent while travelling along the route of multimodal process mP_1 – North-South (distinguished by red colour line, see Figs. 3 and 4), is equal to mFT_{AB} = 3·48 - 18= 126 u.t. ($mFT_{AB} = c \cdot {}^{(i)}m\alpha_1 - \Delta F_1$; where: c – the number of EISs consisted by a transportation route). Consequently, the whole travel time along transportation route of mP_1^1 is equal to mFT_1= 437 u.t. Tabulated summary of passengers' itinerary scenarios distinguished in Fig. 3 is provided in Table 2.

Table 2 Summary of passengers' itinerary scenarios distinguished in Fig. 5

Direction	Multimodal process	Number of EISs c	Flow time mFT_i
NORTH-SOUTH	mP_1^1	10	437
	$mP_{1'}^1$	6	245
	$mP_{1''}^1$	4	149
WEST-EAST	mP_2^1	10	437
	$mP_{2'}^1$	11	485
	$mP_{2''}^1$	6	245
NORTH-WEST	mP_3^1	11	485

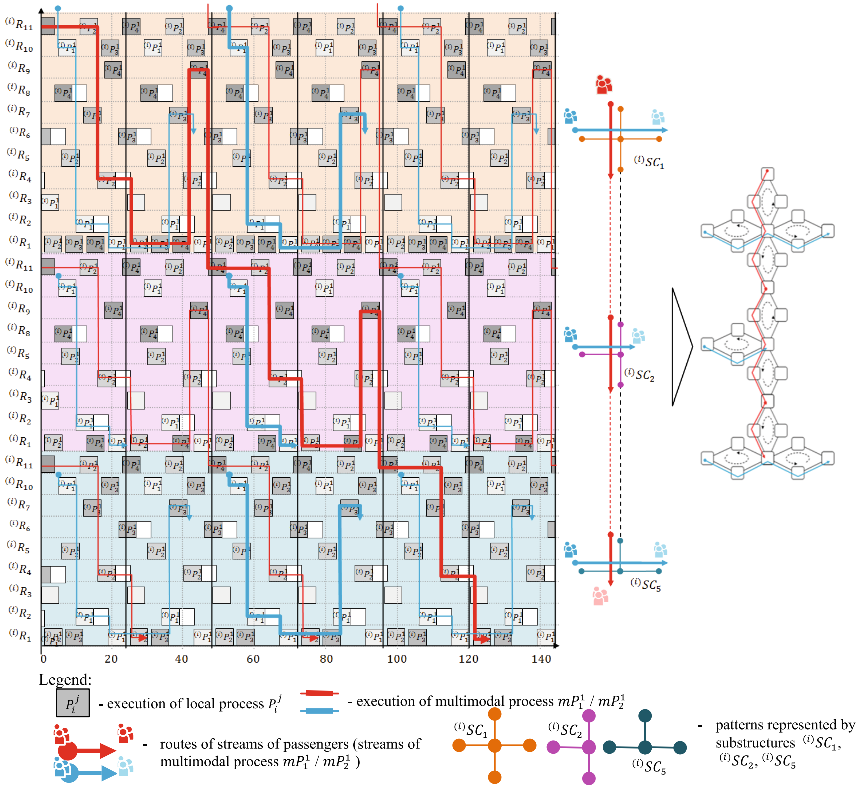


Fig. 4 The Gantt's chart of a part of fractal-like network from Fig. 3 including both local and multimodal processes

Note, that in all cases considered the takt time is equal to $mTc_i = {}^{(i)}m\alpha_1 = 48$ u.t. - according to schedule ${}^{(i)}X'_1$, i.e. the stream of the each multimodal processes arrives to each resources only once on the one period of ${}^{(i)}m\alpha_1 = 48$. The obtained schedule allows to determine travel efficiency TE_i which is defined as a ratio of execution time of the process ${}^{(i)}mP_i$ to the period ${}^{(i)}m\alpha_1$. So, in considered case each process has the same value TE_i = 41,66%.

6 Conclusions

A declarative modelling approach to passengers' itinerary planning for assumed local transportation modes timetable setting guaranteeing assumed travel time in FNLTM environment is considered. In considered fractal structure network, the passengers pass their origin-destination itineraries along routes composed of local transportation means coupled via common shared interchange stations. The solution sought assumes that schedules of locally acting subnetworks composed of a set of a given transportation lines will match-up the schedules of assumed set of passengers itinerary. The relevant sufficient conditions guaranteeing such a match-up exists were provided. Declarative approach following these conditions allows to find the cyclic schedule of the whole FNLTM using schedules of EISs (treated as substructures of FNTLN). The size of EISs used cannot, however, exceed 40 processes [2].

The most important challenge for further work deals with the analysis of different fractal topologies especially heterogeneous networks composed of different substructures of transportation modes and equipped with different numbers of servicing means.

References

1. Abara, J.: Applying integer linear programming to the fleet assignment problem. Interfaces **19**, 4–20 (1989)
2. Bocewicz, G., Nielsen, I., Banaszak, Z.: Iterative multimodal processes scheduling. Annual Reviews in Control **38**(1), 113–122 (2014)
3. Bocewicz, G., Banaszak, Z., Pawlewski, P.: Multimodal cyclic processes scheduling in fractal structure networks environment. In: Proc. of the 19th World Congress: The International Federation of Automatic Control, Cape Town, pp. 8939–8946 (2014)
4. Kelly, G., McCabe, H.: A Survey of Procedural Techniques for City Generation. ITB Journal (14), 87-130
5. Levner, E., Kats, V., Alcaide, D., Pablo, L., Cheng, T.C.E.: Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers and Industrial Engineering **59**(2), 352–361 (2010)
6. Levinson, D., Huang, A.: A positive theory of network connectivity. Environment and Planning B: Planning and Design **39**, 308–325 (2012)

7. Polak, M., Majdzik, P., Banaszak, Z., Wójcik, R.: The performance evaluation tool for auto-mated prototyping of concurrent cyclic processes. Fundamenta Informaticae **60**(1-4), 269–289 (2004)
8. Relich, M.: Using ERP database for knowledge acquisition: a project management perspective. In: Proceedings of International Scientific Conference on Knowledge for Market Practice, pp. 263–269 (2013)
9. Sitek, P., Wikarek, J.: A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management. International Journal of Production Research, 1–18 (2015). doi:10.1080/00207543.2015.1005762
10. Song, J.-S., Lee, T.-E.: Petri net modeling and scheduling for cyclic job shops with blocking. Computers & Industrial Engineering **34**(2), 281–295 (1998)
11. Von Kampmeyer T.: Cyclic scheduling problems. Ph.D. Dissertation, Mathematik/Informatik, Universität Osnabrück (2006)

Modeling and Solving the Soft Constraints for Supply Chain Problems Using the Hybrid Approach

Paweł Sitek

Abstract Many real-life problems in Supply Chain (SC) are over-constrained. Insufficient resources and time requirements result in an inability to all constraints. In some cases, their fulfillment requires very intensive computing. One way to overcome these difficulties is to soften some of the constraints. Natural environment for the modeling and solving of problems with constraints is constraint programming (CP), which is, however, ineffective on optimization problems and constraints containing a sum of many decision variables. This is when the idea to hybridize CP and other environments originated. The article presents the concept of modeling and solving soft constraints in SC problems using the hybrid approach. The illustrative example provided in the paper illustrates effectiveness and possibilities of this approach.

Keywords Soft constrains · Constraint logic programming · Optimization · Mathematical programming · Supply Chain

1 Introduction

Supply chain problems are characterized by a number of constraints (relating to manufacturing, products, transportation, distribution, resources, time, finance, environment, etc) and sustainable logistics collaboration [1]. Multiple constraints and the complexity of supply chain (SC) problems make them over-constrained. The term ”over-constrained problem” refers to the situation when it is impossible to satisfy all the constraints in the given problem. This is usually due to insufficient amount of resources and/or delivery lead time. Preferences are another terms that occurs in SC problems. Constraints and preferences are closely related

P. Sitek(✉)
Department of Information Systems, Kielce University of Technology, Kielce, Poland
e-mail: sitek@tu.kielce.pl

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_53

conceptually. Preferences can be considered as a form of “tolerant” constraints. Soft constraints are added to models to tackle over-constrained problems and preferences. Several CP-based frameworks have been developed to model soft constraints.

This paper proposes a complementation to the hybrid approach [2,3] offering a possibility of modeling and solving soft constraints. The method for the modeling of such constraints is described along with the results of the analysis of how soft constraints affect the effectiveness of the hybrid approach. The authors present their illustrative example for supply chain management cost optimization with hard and soft constraints.

The rest of the article is organized as follows. Research methodology is provided in Section 2. In Section 3 the concept of modeling and solving soft constraints in the hybrid approach is described. The illustrative example and numerical examples are provided in Section 4. The conclusions are included in Section 5.

2 Literature Review and Methodology

To Constraint programming (CP) environments are best suited to specification problems with constraints [4,5]. CP is a powerful paradigm for modeling and solving combinatorial search problems. CP is based on a wide range of methods and algorithms from AI (Artificial Intelligence) [6,7,8], CS (Computer Science) and OR (Operation Research). The basic concept in constraint programming is that a user states the set of the constraints describing the problem and the constraint solver is used to solve them. The central issue of constraint programming is a CSP (Constraint Satisfaction Problem) [5]. A CSP is a problem composed of a set of decision variables, each having a domain of values and a set of constraints. Each constraint is defined over some sub-set of the primary/original set of variables and restricts the values these variables can simultaneously take. The decision variables and domains are finite. Constraint solvers take real-world problems like scheduling, planning, configuration, location, etc., specified in term of decision variables and constraints and assign the values to all decision variables, which satisfy all the constraints. Such solvers search the solution space either regularly, as with B&B (branch-and-bound) algorithms or backtrackings, or use forms of local search (which may be incomplete) [5]. Search process is accelerated by constraint propagation. Constraint propagation is the process of determining how the constraints and the possible values of one variable affect the possible values of other variables. The constrain logic programming (CLP) is a form of constraint programming (CP), in which logic programming is extended to include concepts from constraint satisfaction [4]. A constraint logic program contains constraints in the body of clauses (predicates). There are however essential obstacles on the use of this approach to real-life problems. The first one is that the set of all constraints may not be satisfied at the same time, which results in a lack of solution to the problem (the problem is over-constrained - Section 1). The second obstacle is due to the structure of the problem, which can contribute to poor impact of constraint propagation and the occurrence of a large number of backtrackings. In many practical problems both situations may combine and limit the effective applicability of the CP approach considerably. One of the ways proposed for overcoming this difficulty is the use of the hybrid approach [2,3,14] with an added possibility of modeling and solving soft constraints. The softening of constraints for over-constrained problems has been the subject of a number of studies, analyzes and publications. The most common solutions in the context of the CP and CSP include: *Partial CSP* which maximizes the number of satisfied constraints [9]; *Weighted CSP* which associates a weight to each constraint and maximizes weighted sum of satisfied constraints [10]; *Possibilistic CSP* associates weight to each constraint representing its importance and hierarchical satisfaction of most important constraints, i.e. maximizes the smallest weight over all satisfied constraints [11]; *Fuzzy CSP* associates weight to each tuple of every constraint and maximizes the smallest preference level (over all constraints) [12]. These approaches typically use an explicit representation (tuples) or a functional representation, and are usually limited to constraints of small arity (unary and binary). Unfortunately, such constraints are a significant minority in the supply chain problems. Thus, research into other ways to address softening of constraints is necessary. One of the proposals is *Cost-based approach*. This approach [13] introduces a cost variable for each soft constraint which represents some violation measure of the constraint. Moreover, it optimizes aggregation of all cost variables (e.g., takes their sum, or max) and uses upper bound on cost variable to apply cost-based filtering (with backpropagation).

3 The Concept of Modeling and Solving Soft Constraints in the Hybrid Approach

The idea of constraint softening will be presented using the example of a MILP (Mixed Integer Linear Programming) model, although it may involve many types of optimization and decision-making models. This is due to the fact that the vast majority of problems in SC are modeled in the form of MILP [15,16]. A general diagram of the proposed approach is shown in Figure 1. First, the MILP model for a specific SC problem is transformed. Reduced and transformed are its decision-making variables and constraints. Then a constraint or constraints are selected to be soft constraints. The softening process consists of complementing these constraints with additional decision-making variables related to the level at which the acceptable values of the parameters (e.g. storage capacities, capacity, the number of means of transport, etc.) are exceeded. In addition, these variables together with coefficients of penalty for exceeding the limit values are entered into the objective function, obviously worsening its value. The proposed approach is different from that proposed in the literature (Section 2) for environments that use the CSP. It is most similar to *the Cost-based approach.* A practical way to apply the proposed solution will be presented step by step for the illustrative example (Section 4).

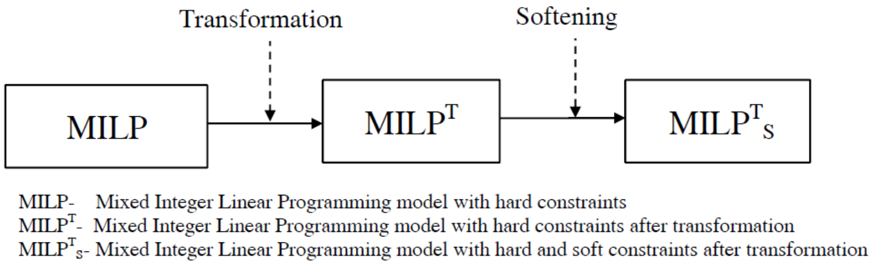

Transformation
Softening
MILP
MILP^T
MILP^T_S
MILP- Mixed Integer Linear Programming model with hard constraints
MILP^T- Mixed Integer Linear Programming model with hard constraints after transformation
MILP^T_S- Mixed Integer Linear Programming model with hard and soft constraints after transformation

Fig. 1 The concept scheme of the modeling and solving soft constraints in hybrid approach

From a variety of tools for the implementation of CLP techniques in the hybrid approach for modeling, transformation and softening, ECL^iPS^e software [17] was selected. ECL^iPS^e is an open-source software system for the cost-effective development and deployment of constraint programming applications. Environment for the implementation of MP for solving $\text{MILP}^T{}_S$ model was LINGO [18].

4 Illustrative Example and Numerical Examples

The proposed approach was verified and tested for illustrative example. The example is the authors' original model of cost optimization of supply chain with multimodal transport. The mathematical model and its discussion are presented in [19]. The model was formulated as a MILP problem. The simplified structure of the supply chain network for this model is composed of manufacturers, distributors and customers. The objective function (F1) is a cost of entire supply chain (production, delivery, environmental, distribution). The decision variables, constraints and parameters of the model are shown in Tab. 2. Practical application of the proposed approach (Fig. 1) to the modeling and solving of soft type constraints will be presented on a selected constraint and objective function of the MILP model in the illustrative example [19]. The objective function and the constraint representing the capacity of distributors are formulated in the form of (F1) and (C4) respectively. The model is then transformed (transformation of decision variables and constraints). As a results of the transformation, the objective function (F1) and the constraint (C4) take the form of (F1T) and (C4T). In the next step, after the softening process constraint (C4T) is supplemented with additional decision variables (*Xrc_b*) forming two constraints (C4TS) and (C5TS). The same decision variables but multiplied by penalty coefficients (krc_b,vr_b) for exceeding allowable capacities complement the objective function, which takes the form of (F1TS).

$$\begin{aligned}&\sum_{e=1}^{E}Od_e\Big(\sum_{b=1}^{B}\sum_{c=1}^{C}Yc_{b,c,e}+\sum_{a=1}^{A}\sum_{b=1}^{B}Xc_{a,b,e}\Big)+\sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{d=1}^{D}\sum_{e=1}^{E}K2_{b,c,d,e}\cdot Yp_{b,c,d,e}+\\&\sum_{a=1}^{A}\sum_{b=1}^{B}\sum_{d=1}^{D}\sum_{e=1}^{E}K1_{a,b,d,e}\cdot\sum_{c=1}^{C}Xp_{a,b,d,e,c}+\sum_{a=1}^{A}\sum_{d=1}^{D}\Big(C_{a,d}\cdot\sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{e=1}^{E}Xp_{a,b,d,e,c}\Big)+\sum_{b=1}^{B}F_b\cdot Tc_b\end{aligned}\qquad\text{(F1)}$$

$$\sum_{e=1}^{E} Od_{E} \cdot (\sum_{a=1}^{A}\sum_{b=1}^{B} Xc_{a,b,e} + \sum_{b=1}^{b}\sum_{c=1}^{c} Yc_{b,c,e}) + \sum_{a=1}^{A}\sum_{b=1}^{B}\sum_{e=1}^{O} (Xc_{a,b,e} \cdot Ksc_{a,b,e}) + \sum_{b=1}^{B} Tc_{b} \cdot F_{b}$$
$$\sum_{a=1}^{A}\sum_{d=1}^{D}\sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{e1=1}^{E}\sum_{e2=1}^{E} (Xp^{T}_{a,d,b,c,e\ 1,e2} \cdot Kz_{a,d,b,c,e\ 1,e2}) + \sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{e=1}^{E} (Yc_{b,c,e} \cdot Ksm_{b,c,e}) \qquad \text{(F1T)}$$

$$\sum_{e=1}^{E} Od_{E} \cdot (\sum_{a=1}^{A}\sum_{b=1}^{B} Xc_{a,b,e} + \sum_{b=1}^{b}\sum_{c=1}^{c} Yc_{b,c,e}) + \sum_{a=1}^{A}\sum_{b=1}^{B}\sum_{e=1}^{O} (Xc_{a,b,e} \cdot Ksc_{a,b,e})$$
$$\sum_{a=1}^{A}\sum_{d=1}^{D}\sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{e1=1}^{E}\sum_{e2=1}^{E} (Xp^{T}_{a,d,b,c,e1,e2} \cdot Kz_{a,d,b,c,e1,e2}) + \qquad \text{(F1TS)}$$
$$+ \sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{e=1}^{E} (Yc_{b,c,e} \cdot Ksm_{b,c,e}) + \sum_{b=1}^{B} Tc_{b} \cdot F_{b} + \sum_{b=1}^{B} Krc_{b} \cdot Xrc_{b}$$

$$\sum_{d=1}^{D} (P_{d} \cdot \sum_{a=1}^{A}\sum_{e=1}^{E}\sum_{c=1}^{C} Xp_{a,b,d,e,c}) \le Tc_{b} \cdot V_{b} \forall b = 1..B \qquad \text{(C4)}$$

$$\sum_{a=1}^{A}\sum_{d=1}^{D}\sum_{c=1}^{C}\sum_{e1=1}^{E}\sum_{e2=1}^{E} P_{d} Xp^{T}_{a,d,b,c,e1,e2} \le V_{b} \cdot Tc_{b} \forall b = 1..B \qquad \text{(C4T)}$$

$$\sum_{a=1}^{A}\sum_{d=1}^{D}\sum_{c=1}^{C}\sum_{e1=1}^{E}\sum_{e2=1}^{E} P_{d} Xp^{T}_{a,d,b,c,e1,e2} \le V_{b} \cdot Tc_{b} + Xrc_{b} \forall b = 1..B \qquad \text{(C4TS)}$$

$$Xrc_{b} \cdot Vr_{b} \le V_{b} \cdot Tc_{b} \cdot Xrc_{b} \forall b = 1..B \qquad \text{(C5TS)}$$

Table 1 Summary indices, parameters, decision variables, constrains for illustrative example

Symbol	Description		
	Indices		
A	number of manufacturers/factories	d	product item *(d=1..D)*
C	number of customers	c	customer/retailer *(c=1..C)*
B	number of distributors	a	factory (manufacturer) *(a=1..A)*
D	number of products	b	distributor *(b=1..B)*
E	number of mode of transport	e	mode of transport *(e=1..E)*
	Input parameters		
F_b	cost of distributor (fixed) *b*		
P_d	volume/capacity occupied by product item *d*		
V_b	distributors' *b* maximum capacity/volume		
$C_{a,d}$	cost of product item *d* at factory *a*		
Od_e	environmental cost of using mode of transport *e*		
$Ksc_{aa,b,e}$	cost of the course/route from the factory *a* to distributor *b* using mode of transport *e*		
$Ksm_{b,c,e}$	cost of the course/route from the distributor *b* to customer *c* using mode of transport *e*		
$Kz_{a,d,b,c,e1,e2}$	variable cost of delivery of product *d* from manufacturer *a* to customer *c* via distributor *b* using mode of transport *e1* and *e2* static		
Krc_b	coefficient of penalty if the capacity of the distributor *b* is exceeded more than the permissible standard		

Table 2 (*Continued*)

Vr_b	coefficient determines the maximum pecentage at which the capacity of the distributor *b* can be exceeded
$K1_{a,b,d,e}$	variable cost of supplying product item *d* from manufacturer *a* to distributor *b* by transport mode *e*
$K2_{b,c,d,e}$	variable cost of supply of product item *d* from distributor *b* to customer/retailer *c* by transport mode *e*
	Decision variables
$Xp_{a,b,d,e,c}$	quantity supplied of a product item *d* from manufacturer *a* to distributor *b* using mode of transport *e* to customer/retailer *c*
$Xc_{a,b,e}$	number of courses from manufacturer *a* to distributor *b* using mode of transport *e*
$Yc_{b,c,e}$	number of courses from distributor *b* to customer *c* using mode of transport *e*
Tc_b	if the distributor *b* is working than $Tc_b=1$ otherwise $Tc_b=0$
$Xp^T_{a,d,b,c,e1,e2}$	decision variable after transformation Xp^T, unlike the initial decision variables *Xp,Yp* is generated only for technologically possible indices combinations. It defines the allocation size of product *d* to the route of deliveries.
Xrc_b	unless the permissible capacity of the distributor is exceeded
	Constraints
C1	all supplies of product item *d* produced by the factory *a* to all distributors *b* using mode of transport *e* do not exceed the production capacity at factory *a*.
C2	coverage of all customer/retailer *c* orders ($Z_{c,d}$) for product *d* as a result of supply by distributors *b*.
C3	balance flow of the products *d* through the distributor *b*
C4	possibility of supply by the distributor *b* (depends on its technical capabilities-capacities)
C5	delivery time conditions
C6	necessary numbers of means of transport for the supply

4.1 Numerical Experiments

All the experiments relate to the supply chain with five manufacturers (*a=1..5*), four distributors (*b=1..4*), twenty customers (*c=1..20*), four modes of transport (*e=1..4*), twenty types of products (*d=1..20*), and six sets of orders *($P_1(10)$, $P_2(20)$, $P_3(40)$, $P_4(40)$, $P_5(50)$,$P_6(60)$ -(n)* - the number of orders in set *Pi)* and $V_1=V_2=V_3=V_4=800$. Computational experiments consisted in solving MILP^{T} and $\text{MILP}^{\text{T}}_{\text{S}}$ models by means of the proposed hybrid approach (H) and MILP and MILP_{S} model using mathematical programming (L). Table 2 shows the results obtained. The conclusions are the following. Firstly, the superiority of the hybrid approach (H) over the classic one using mathematical programming (L) is confirmed. This applies to both MILP/MILP^{T} models, as confirmed earlier [3,5,19,20], and MILP_{S}/ $\text{MILP}^{\text{T}}_{\text{S}}$ models. For examples P1..P4, the shortening of the calculation time is manifold. In the case of MILP/MILP^{T} models (with hard constraints), no solution was found, which was due to insufficient capacity of distributors *(V_b).* The solution was found after softening constraints C4 and C4T, whereby the solution is acceptable for model the MILP_{S} model optimal for the

$\text{MILP}^{T}{}_{S}$ model. Obtained optimization times for the MILP^{T} and $\text{MILP}^{T}{}_{S}$ models are virtually identical.

In conclusion, softening allowed finding the optimal solution, which was impossible to find when the constraints were hard. Further experiments examined the influence of the coefficient (vrc, krc) values on the value of the objective function and the calculation time (Tab. 3) and implementation of fuzzy logic [21].

Table 2 Results of numerical experiments

P_i(n)	**M**	**V**	**C**	**Fc^{opt}**	**T**	**V**	**C**	**Fc^{opt}**	**T**
		Hard MILP (L) and MILP^T (H)				**Soft** $vrc_1=vrc_2=vrc_3=vrc_4=30\%$ $krc_1=krc_2=krc_3=krc_4=10$ MILP_S (L) and $\text{MILP}^T{}_S$ (H)			
P_1(10)	L	80117(79204)	47430	16518	22	80121(79204)	47430	16518	21
	H	817(788)	1370	16518	9	822(788)	1371	16518	8
P_2(20)	L	80117(79204)	50630	57071	148	80121(79204)	50630	57071	335
	H	987(948)	1380	57071	54	992(948)	1381	57071	41
P_3(30)	L	80117(79204)	53830	110335	534	80121(79204)	53830	102998	639
	H	1264(1220)	1385	110335	234	1269(1220)	1389	102998	209
P_4(40)	L	80117(79204)	57030	219647	757	80121(79204)	57030	176272	822
	H	1536 (1492)	1365	219647	356	1541(1492)	1356	176272	319
P_5(50)	L	80117(79204)	60230	NSF	---	80121(79204)	60230	371246*	900**
	H	1696(1652)	1385	NSF	---	1701(1652)	1386	360005	345
P_6(60)	L	80117(79204)	63430	NSF	---	80121(79204)	63429	470324*	900**
	H	2086(2036)	1385	NSF	---	2085(2036)	1386	468398	389

P_n(n)	Set of orders (n)	**C**	the number constraints
M	Method L- MP, H- Hybrid	**Fc^{opt}**	the value of the objective function
V	the number of decision variables (integer decision variables	**T**	time of finding solution (in seconds)
		NSF	no solution found
*	the feasible value of the objective function after the time T	**	the calculation was stopped after 900 s

Table 3 Results of numerical experiments form MILP^T and $\text{MILP}^T{}_S$ (for different krc)

P_i(n)	Vrc=25											
	HARD		krc=1		krc=5		krc=10		krc=15		krc=20	
	Fc	T	Fc	T	Fc	T	Fc	T	Fc	T	Fc	T
P_1(10)	16518	6	16518	5	16518	6	16518	6	16518	7	16518	6
P_2(20)	57071	15	57071	14	57071	16	57071	15	57071	15	57071	17
P_3(30)	110756	123	100611	143	106986	154	110473	131	110609	142	110756	130
P_4(40)	234344	234	175786	272	185471	243	200291	211	216098	222	220080	241
P_5(50)	NSF	1	353747	323	368123	345	387705	345	409399	339	426629	333

5 Conclusion

This paper provides a robust and effective hybrid approach to modeling and solving of SC problems with soft and hard constraints, which incorporates two environments (i) mathematical programming (LINGO) and (ii) constraint logic programming (ECLiPSe).

The modeling of soft constraints is in many cases the only alternative possible in order to solve an over-constrained problem. Supplementing the hybrid approach with the possibility of modeling and solving soft constraints increases its versatility and application scope without impairing its performance. Comparable solving times obtained for $\mathrm{MILP}^{\mathrm{T}}$ and $\mathrm{MILP}^{\mathrm{T}}_{\mathrm{S}}$ were significantly shorter than those in the case of MILP and $\mathrm{MILP}_{\mathrm{S}}$ (Tab.2). The application of this approach to illustrative example leads to a substantial reduction in (i) the number of decision variables (up to one hundred times), (ii) the number of constraints (more than forty times) (iii) computing time. The proposed approach allows modeling and solving any soft constraints within an acceptable time period. It is also possible to evaluate and select penalty coefficients for violation (Tab. 3) and estimate their impact on the value of the objective function.

References

1. Grzybowska K., Awasthi A., Hussain M.: Modeling enablers for sustainable logistics collaboration integrating Canadian and Polish perspectives. In: Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.) Federated Conference on Computer Science and Information Systems, ACSIS, vol. 2, pp. 1311–1319 (2014). doi:10.15439/2014F90
2. Sitek, P., Wikarek J.: A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management. International Journal of Production Research, 6611–6628 (2015). doi:10.1080/00207543.2015.1005762
3. Sitek, P., Wikarek, J.: A hybrid approach to the optimization of multiechelon systems. In: Mathematical Problems in Engineering, Article ID 925675. Hindawi Publishing Corporation (2014). doi:10.1155/2014/925675
4. Apt, K., Wallace, M.: Constraint Logic Programming using Eclipse. Cambridge University Press, Cambridge (2006)
5. Rossi, F., Van Beek, P., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York (2006)
6. Bzdyra K., Bocewicz G., Banaszak Z.: Mass customized projects portfolio scheduling-imprecise operations time approach. In: Applied Mechanics and Materials, vol. 791, pp. 70–80 (2015). doi:10.4028/www.scientific.net/AMM.791.70
7. Bocewicz, G., Nielsen, I., Banaszak, Z.: Iterative multimodal processes scheduling. Annual Reviews in Control **38**(1), 113–132 (2014)
8. Relich, M.: A computational intelligence approach to predicting new product success. In: Proceedings of the 11th International Conference on Strategic Management and its Support by Information Systems, pp. 142–150 (2015)
9. Freuder, E.C., Wallace, R.J.: Partial constraint satisfaction. Artificial Intelligence **58**, 21–70 (1992)
10. Larrosa J.: On arc and node consistency in weighted CSP. In: Proc. AAAI 2002, Edmondton, (CA) (2002)
11. Schiex, T.: Possibilistic constraint satisfaction problems or "How to handle soft constraints?". In: Proc. of the 8th Int. Conf. on Uncertainty in Artificial Intelligence, Stanford, CA, July 1992, p. 4 (1994)

12. Dubois, H., Prade H.: The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction. In: Proc. 2nd IEEE Conference on Fuzzy Sets, San Francisco, CA, March 1993
13. Petit, T., Regin, J.-C., Bessi C.: ere. Meta-constraints on violations for over constrained problems. In: IEEE-ICTAI 2000 International Conference, pp. 358–365. Vancouver, Canada (2000)
14. Milano, M., Wallace, M.: Integrating Operations Research in Constraint Programming. Annals of Operations Research **175**(1), 37–76 (2010)
15. Mula, J., Peidro, D., Diaz-Madronero, M., Vicens, E.: Mathematical programming models for supply chain production and transportation planning. European Journal of Operational Research **204**, 377–390 (2010)
16. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons, New York (1998)
17. Eclipse: Eclipse - The Eclipse Foundation open source community website, Accessed August 12, www.eclipse.org (2015)
18. Lindo Systems INC: LINDO™ Software for Integer Programming, Linear Programming, Nonlinear Programming, Stochastic Programming, Global Optimization, Accessed May 4, www.lindo.com (2015)
19. Sitek, P.: A hybrid CP/MP approach to supply chain modelling, optimization and analysis. In: Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, pp. 1345–1352 (2014). doi:10.15439/2014F89
20. Wikarek, J.: Implementation aspects of Hybrid Solution Framework (HSF). In: Recent Advances in Automation, Robotics and Measuring Techniques Advances in Intelligent Systems and Computing, vol. 267, pp. 317–328 (2014)
21. Kłosowski G., Gola A., Świć, A: Application of fuzzy logic controller for machine load balancing in discrete manufacturing systems. In: IDEAL 2015. Lecture Notes in Computer Science, vol. 9375, pp. 256–263 (2015). doi:10.1007/978-3-319-24834-9_31

Application of Fuzzy Logic in Assigning Workers to Production Tasks

Grzegorz Kłosowski, Arkadiusz Gola and Antoni Świć

Abstract The paper demonstrates the concept of selecting workers for production tasks with the use of fuzzy logic. Mamdani-type fuzzy inference method was used to design a controller whose task was to aid the decision-making process. Fuzzy interference system converted a number of qualitative features into quantitative variables, which enabled calculating deviations and their comparison. Moreover, a simulation model of a discrete manufacturing system with an implemented fuzzy controller was developed. conducted simulation provided detailed data regarding the investigated selection process.

Keywords Simulation · Modelling · Control · Fuzzy logic · Manufacturing system

1 Introduction

Current manufacturing systems are characterised by a high technological level of applied means of labour. Modern technological machines and both logistic and informational systems are nowadays a standard [1-2]. Still, what seems to be an indispensable element for the operation of most manufacturing systems is a human factor. Human labour has often been considered as the only cost-effective alternative to expensive automated solutions, as well as an easily interchangeable highly flexible resource able to adapt production capacity and to quickly change product features [3-5]. The issue gains in importance especially in the context of analysis

G. Kłosowski · A. Gola(✉) · A. Świć
Faculty of Management, Department of Enterprise Organization,
Lublin University of Technology, Lublin, Poland
e-mail: {g.klosowski,a.gola,a.swic}@pollub.pl

G. Kłosowski · A. Gola · A. Świć
Faculty of Mechanical Engineering, Institute of Technological Systems of Information,
Lublin University of Technology, Lublin, Poland

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_54

of causes and effects concerning operational problems as well as manufacturing systems reliability. It is common knowledge that a measure of reliability of every system is the level of its weakest link solidity. As confirmed by the research conducted several years ago, human errors in the field of production accounted for over 82% of all errors [6]. Another study, focused on the mechanical engineering industry, showed that workers spend as much as 65% of their work time correcting skill-based errors, rule-based errors accounted for 32% and knowledge-based errors constituted 3% [7].

What seems necessary for that purpose is to create personal profiles for each production worker [8-9]. The method of describing production tasks with a set of qualities was presented. Considerable complexity of the processes taking place in discrete manufacturing systems is a justifiable reason for applying methods suitable for their analysis [10-11], which frequently involves the application of computer software that uses expert knowledge [12].

Since multiple qualities describing production workers and production tasks are of qualitative nature, it is difficult to assign to them numerical qualifier values. For that reason, it is fuzzy logic that was applied in the matching process. A new two-level fuzzy inference system was developed. It comprised a range of heuristics which facilitated the conversion of qualitative variables to values, thus enabling their comparison. In order to test the capabilities of a fuzzy controller in solving problems of assigning workers to tasks, a simulation model of a manufacturing system was created.

2 Problem Description

The aim of the research is a preliminary verification of methods for selection of production workers for technological tasks, operations and machines in order to reduce the costs resulting from human errors.

The object of the study is a discrete manufacturing system of a middle-sized or a large enterprise. The system consists of a number of workstations with different machine tools (lathes, milling machines, drilling machines *etc*). Among workers employed in the production department there are technological machine operators. Based on the criterion of technological and design similarities as well as the scope and the type of undertaken operations, machine tools are divided into groups. It is assumed that each operator demonstrates competence suitable for using several similarly operated machine tools. Such an approach enables a more in-depth analysis of the process of assigning production tasks to particular workers and technological machines.

In middle-sized and large enterprises, management encounters challenges relating to a great complexity of manufacturing systems. This is what triggers the need to seek proper advanced managing methods. One of the elements of an efficient large organisation is profiling professional personalities of workers. Typically, profiling focuses on two cases:

- initial profiling – when candidates are selected for established requirements for a position
- transfer profiling – when working conditions are matched to the preferences and strengths of a worker

Usually initial profiling relates to the recruitment of new staff members, whereas transfer profiling concerns so-called career path forming as far as already employed workers are concerned.

Manufacturing companies, which perform numerous orders, can rate their importance. For obvious reasons, identifying priorities is not what should influence the sequence of their realisation. It is the order of jobs and contractual obligations that are in most cases decisive. The importance of an order can have a significant impact on the selection of production workers engaged in particular stages of job realisation. Examples of evaluation criteria that determine the rank of orders are presented in Table 1.

Table 1 Assessment of production order features (order profiling)

No.	Qualitative order features	Assessment method
1.	Urgency (due date)	Likert scale classification from 1 to 5
2.	Order prestige	
3.	Level of technological complexity	
4.	Production staff requirements (communication level, creativity, invention, diligence)	
	Quantitative order features	
5.	Order value	[PLN]
6.	Order scale	[pieces]
7.	Unit cost of material purchase	[PLN/pc.]
8.	Production cycle	[time]
9.	Gross margin with reference to the whole order	[%]
10.	Acceptable shortage level	[%]

Source: own work

The first five order features are of qualitative nature, which makes the evaluation even more challenging. It is necessary to adopt subjective evaluation methods such as questionnaires or surveys. Another more sophisticated technique to deal with the subject draws on developing heuristics which would automatically use specific quantitative variable vectors downloaded directly from databases of information systems to estimate the values of qualitative variables.

Fig. 1 demonstrates the structure of an employee profile, which can be applied in the process of selecting workers for production tasks.

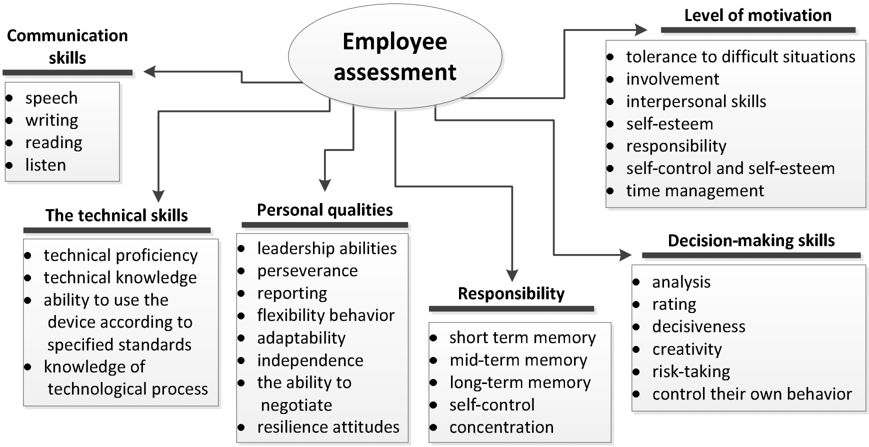


Fig. 1 The structure of an employee profiling process [14]

A personal profile consists of six general qualities, which can furthermore be divided into specific qualities. Each general quality can therefore be treated as a black box with several inputs and one output where quantified values of specific qualities are the inputs and a calculated value of a general quality constitutes an output. In order to estimate the indicators of specific qualities, questionnaires and surveys can be applied.

3 Modelling of the Manufacturing System with Fuzzy Logic Based Decision Support Controller

In order to solve the problem of assigning workers to production tasks computer simulation was used. As a part of the study, a new manufacturing system model comprising the material flow subsystem and the information subsystem was developed. The information subsystem is responsible for executing the processes of profiling production workers and tasks as well as assigning workers to appropriate tasks within the machine route. Both selection and assigning processes are executed by a two-step controller, which first converts quantified specific qualities to general quality values and afterwards, using general qualities as inputs, ranks a given worker for a particular task. A crucial feature of the presented solution is applying the matching degree of all six general qualities. Fig. 2 shows a subsystem which executes the matching of a worker's qualities from *Communication Skills* to a particular task expressed by the quality *Need of communication skills*. Numerical value of an order quality can be calculated deterministically based on quantitative data included in databases of the company's information system or with the use of heuristics.

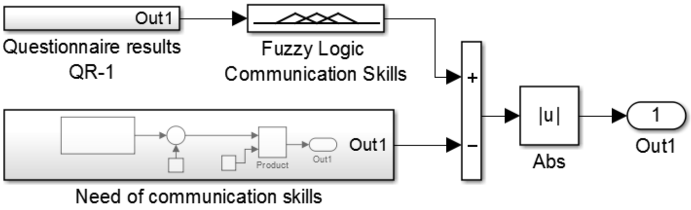

Out1
Questionnaire results QR-1
Fuzzy Logic Communication Skills
+
−
|u|
Abs
1
Out1
Product
Out1
Out1
Need of communication skills

Fig. 2 Assessment of Communication Skills for a particular production task

The results of the survey were processed by the fuzzy interference system of the first level. As shown in Fig. 1, *Communication Skills* embrace four qualities: *speech*, *writing*, *reading* and *listening.* Each of those employee qualities was assigned a numerical value from 1 to 5. Fuzzy logic controller converted a vector of four inputs into a single scalar output, which was represented by a real number in a range (0,1). The values of specific qualities were determined with heuristics embedded in a controller based on artificial intelligence techniques, *i.e.* on fuzzy logic. One of the major advantages of fuzzy logic is the ease of formulating interference rules based on linguistic variables. Heuristics were expressed in the form of interference rules applied in Mamdani fuzzy interference systems. After calculating both values of the same quality with respect to an employee and a production task, the system specifies the absolute value of variance (deviation). The smaller the deviation, the better accuracy of matching a particular employee quality to a task. As a result, six deviations are obtained - one for each general quality. Calculating the values of each of the general qualities is accomplished by a separate fuzzy controller. Gaussian distributions served as membership functions (1).

$$f(x;\sigma,c)=e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{1}$$

where: x – input value, σ^2 – variance, c – position.

Defuzzification takes place with the use of five triangular functions depicted by the following pattern (2).

$$f(x;a,b,c)=\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \tag{2}$$

where:

x – input value, *a, c* – position of the triangle base points, *b* – position of the triangle vertex.

The last stage of the process of assigning production workers to tasks focused on the conversion of six deviations, which reflect a matching degree into a single scalar value - a real number in a range (0,1).

A general scheme of a production process model is presented in Fig. 3. At the output, every subsystem generated a real number from 0 to 1, based on which the *Fitness_function* selected the most suitable employee for conducting a particular operation of the job in question. *Fitness_function* code (made in Matlab language) is presented below:

```
function y = Fitness_function(e1, e2, e3, e4)
%#codegen
% e1, e2, e3, e4 – scores for employees No 1, 2, 3, 4.
% Function chooses the machine for given employee.
  employees = [e1, e2, e3, e4];
  [~, y] = max(employees);
```

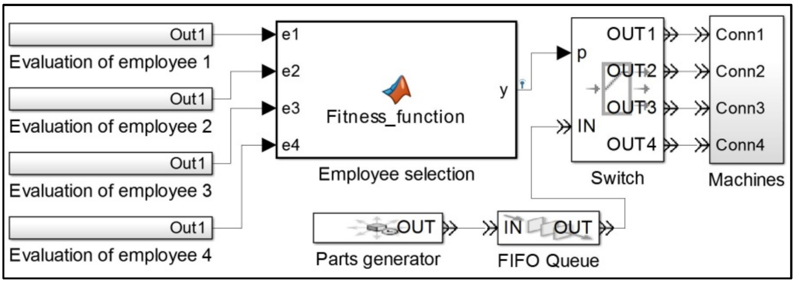


Fig. 3 The model for selecting production workers for tasks

Fig. 4 shows a graphical interpretation of the interference and deffuzification processes. Each of five lines represents a single fuzzy rule. First six columns demonstrate separate input variable, whereas the last one on the right side indicates the output variable. The rectangle in the lower right corner is the representation of the deffuzification with the application of a centroid method.

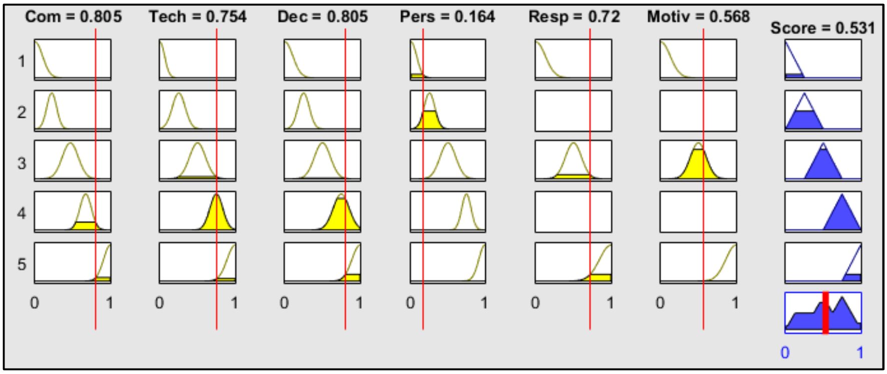


Fig. 4 Graphical interpretation of interference and deffuzification processes

Fig. 5 shows a response surface model generated with a two-step fuzzy controller for two out of six input variables – *Motivation_level* and *Personal_qualities*.

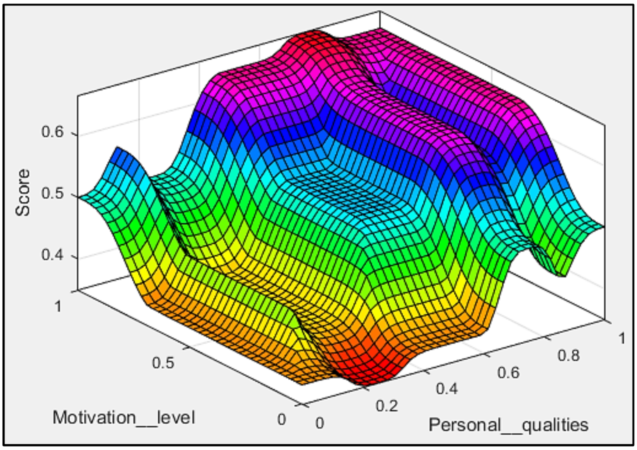


Fig. 5 Response surface of a two-step fuzzy controller for a pair of selected input data

High level of surface diversity is a reflection of a considerable complexity of the analysed problem. Developing a mathematical function performing similar conversions would be considerably time-consuming, which justifies the choice of the fuzzy logic selection tool. One more argument in favour of fuzzy logic is the fact that owing to the heuristics it is particularly adept at solving problems requiring the conversion of qualitative variables into the quantitative ones.

4 Analysis of Simulation Experiments

The conducted study embraced a simulation experiment based on a manufacturing system model which drew on a fuzzy interference system. The simulation run time was set to 100 seconds. In this case, though, the time unit is rather arbitrary, as long as it is applied consistently throughout all elements and subsystems of the model; therefore, seconds could easily be substituted by minutes or hours.

In the course of the simulation, the production line generated 51 new production tasks in the form of the elements requiring machining. All specific qualities of the workers were generated arbitrarily with the use of properly adjusted generators of natural numbers in a range (1,5) to simulate a Likert grading scale. A single production task was entered into the system every two seconds. Fig. 6a shows a chart of the values illustrating the extent to which employee no. 1 matches a particular production task while Fig. 6b presents a chart of *Fitness_function* (Fig. 4) outputs meaning a sequence of selections of one out of four employees for specific production tasks in the course of the simulation.

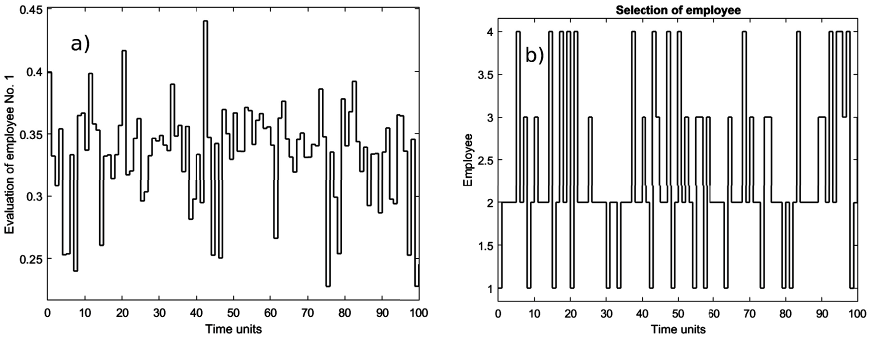


Fig. 6 Employee evaluation chart No. 1 (a); Selection chart of a machine tool (employee) for a task (b)

For reasons of simplicity, it was assumed that each of four employees was permanently assigned to a technological machine. Assigning an employee to a particular production task in the context of the discussed technological operation could therefore be perceived a choice of a specific machine tool. In order to enable unequivocal machine-operator matching, when designing a manufacturing system model, it was highlighted that the discussed number of the employees should correspond to the number of technological machines.

5 Summary

The present paper deals with the concept of optimising the process of selection of production workers in a middle-sized or a large enterprise and assigning them to different production tasks. For this purpose, the method of profiling employees and tasks based on selected qualities was presented. The proposed solution consists in conversion of qualitative variables into scalar values with heuristics included in a fuzzy interference system. A new unconventional method of analysing deviations with respect to corresponding qualities of production workers and tasks was applied. A two-level fuzzy interference mechanism in combination with a deterministic method of calculating variances in terms of specific qualities was proposed.

The fundamental aim of the experiment was not verifying the effectiveness of applied heuristics, selection of input variables or testing a controller based on the data referring to a specific real case. Its principal purpose was to carry out an initial verification of the concept of assigning production workers to tasks with the use of fuzzy logic.

References

1. Gola, A., Świć, A.: Computer-Aided Machine Tool Selection for Focused Flexibility Manufacturing Systems Using Economical Criteria. Actual Problems of Economics **124**(10), 383–389 (2011)
2. Kłosowski, G., Gola, A., Świć, A: Application of fuzzy logic controller for machine load balancing in discrete manufacturing systems. In: Intelligent Data Engineering and Automated Learning–IDEAL 2015. Lecture Notes in Computer Science, vol. 9375, pp. 256–263 (2015)
3. Mossa, G., Boenzi, F., Digiesi, S., Munnolo, G., Romano, V.A.: Productivity and ergonomic risk in human based production systems: A job-rotation scheduling model. International Journal of Production Economics **171**, 471–477 (2016)
4. Relich, M., Świć, A., Gola, A.: A knowledge-based approach to product concept screening. In: 12th International Conference on Distributed Computing and Artificial Intelligence. Advances in Intelligent Systems and Computing, vol. 373, pp. 341–348 (2015)
5. Gola, A., Relich, M., Kłosowski, G., Świć, A.: Mathematical Models for Manufacturing Systems Capacity Planning and Expansion. Applied Mechanics and Materials **791**, 125–131 (2015)
6. Ganguly, S.: Human error vs. work place management in modern organizations. International Journal of Research in Management and Technology **1**(1), 13–17 (2011)
7. Factors leading to human errors. https://oshwiki.eu/wiki/Human_error
8. Wilsher, S.: Behavior profiling: implications for recruitment and team building. Strategic Direction **31**(9), 1–5 (2015)
9. Grayson, M.L., Macesic, N., Huang, G.K., Bond, K., Fletcher, J., Gilbert, G.L., et al.: Use of an Innovative Personality-Mindset Profiling Tool to Guide Culture-Change Strategies among Different Healthcare Worker Groups. PLoS ONE **10**(10), e0140509 (2015)
10. Kłosowski, G., Gola, A., Świć, A.: Human Resource Selection for Manufacturing System Using Petri Nets. Applied Mechanics and Materials **791**, 132–140 (2015)
11. Md Saad, R., Ahmad, M.Z., Abu, M.S., Jusoh, M.S.: Some fuzzy techniques for staff selection process: a survey. In: AIP Conference Proceedings, vol. 1522, p. 462 (2013)
12. Samuel, O.W., Omisore, M.O., Atajeromavwo, E.J.: Online fuzzy based decision support system for human resource performance appraisal. Measurement **55**, 452–461 (2014)
13. Sitek, P.: A hybrid CP/MP approach to supply chain modelling, optimization and analysis. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1345–1352 (2014)

The Model of Quality Assessment of Implementation of Design Patterns

Rafał Wojszczyk and Robert Wójcik

Abstract In software engineering, there are many methods and good practices which aim at ensuring quality of developed software. One of these practices is using design patterns. The article aims at introducing the main frameworks of the authors' model of quality assessment of implementation of design patterns. The model is composed of relevant elements to be described below: an equivalent of the code of the analysed software, information on implementation of patterns and detailed measures. On the basis of the model frameworks, one can verify the structure of implementation as well as determine and improve the quality of implementation of design patterns.

Keywords Design patterns · Software quality · Formal methods

1 Introduction

The trade of software development has been a human creation since its dawn. In the past, it was restrained by hardware limitations, e.g. it was easier to process binary data than decimal data. However, at present, high-level programming languages are very good at hiding the fact that, eventually, software code is compiled to binary code. This results from, among others, the object-oriented paradigm. This paradigm is used in many different models applied in information technology and, especially, in software development: UML modelling language, object-oriented

R. Wojszczyk(✉)
Departament of Computer Science and Management,
Koszalin University of Technology, Koszalin, Poland
e-mail: rafal.wojszczyk@tu.koszalin.pl

R. Wójcik
Department of Computer Engineering, Faculty of Electronics,
Wrocław University of Science and Technology, Wrocław, Poland
e-mail: robert.wojcik@pwr.wroc.pl

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_55

analysis and design, object-oriented databases or, also, in object-oriented programming. Object-oriented programming languages are currently the most popular ones. Simultaneously, this means that they are used to develop a high amount of software which is taken advantage of by many people on a daily basis.

The development of information technology in general, including application software, has benefited many people in their often very responsible jobs or in overcoming difficulties in life. Taking the popularity of computers and mobile devices as well as omnipresence of application software into consideration, internal software quality [13], that is, source code quality, is worth discussing. There are various methods ensuring this quality basically at every stage of software life cycle. One can also see to it already at the very beginning, that is, at the level of the object-oriented paradigm, e.g. through using a set of rules customarily called good practices [5]. Good practices are developed in tandem with the trade of software development. In case of the object-oriented paradigm, good practices are, among others: Solid [5] based on the basic assumptions of object-orientation, logical division of application layers, e.g. MVC, and a relevant division of relations between classes. The latter can be executed by implementation of design patterns described in [4]. Design patterns are templates of ready mechanisms which can be used for solving typical problems occurring regularly in object-oriented designing and programming.

The article aims at presenting the main frameworks of the authors' model which has been developed to assess the quality of implementation of design patterns. Among others, the model enables one to determine the expected quality of implementation of design patterns and compare different software releases.

The selected models connected with software development and quality assessment were introduced in Chapter 2. Chapter 3 described the proposal of the model and explained verification of implementation of design patterns on the basis of the model, while Chapter 4 constitutes a summary.

2 Models in Information Technology

Information technology offers a wide range of different models. In software development, several stages have been commonly accepted [6]: requirements collection, system design, implementation, testing and maintenance. Software development models [6] organise these stages, for instance, the sequence model determines that the stages are successive, while the iterative model predicts the cycles composed of recurring development stages. On the basis of the mentioned models, there were several methodologies developed [6]; these, which are rather general (e.g. waterfall model, V model), separate stages and force relevant sequence. Finally, these which are the more developed (e.g. RUP) describe the role of shareholders, generated artefacts (e.g. documentation, tests, source code and other), and relative labour-intensity of a certain stage in detail. However, the mentioned methodologies do not include the use of design patterns [4], although it could be very beneficial. In the stage of designing, the patterns are useful in developing the architecture and copying verified solutions; during implementation, they support making local decisions, and, finally, in the stage of testing, they facilitate error repair and further development of software.

One should start the deliberation concerning quality with a brief discussion on ISO 9126 standard, which describes the quality of the software product. The third part of the mentioned standard directly concerns the internal quality of software, that means the whole program code. Equally, the quality of the design pattern implementation applies to the extract of the program code which includes the patterns. The standard is described with a tree structure which includes characteristics and in those, recursively, there are subcharacteristics. Characteristics, which do not have other subcharacterictis, have metrics - functions assigned to them, which define certain values on the basis of measurable software attributes. The definition of characteristics is informal - it expresses a certain intent, while the metrics are formalised. By definition, ISO 9126 standard, together with its third part, is not intended for determining the quality of special implementation structures as design patterns. Therefore, the characteristics of the norm apply only to a few aspects of quality (that are) connected with the implementation of design patterns.–

Other metrics of object-oriented programming (e.g. CK, MOOD, Martin's) [6] enable one to express different measurable software properties as numbers. Still, the mentioned metrics cannot be directly used to measure the quality of implementation of design patterns [7] or they require relevant over-interpretation [13]. Finally, the methods directly intended for analysing software with patterns often only count the number of instances of design patterns [8] [10] without taking implementation quality into consideration. What is more, there are studies conducted which mainly aim at showing structural correctness of implementation of the patterns [2], and those also provide too little information, which is insufficient to consider it as quality characteristics. Regardless of using the existing methods, a few methods, which are the basis, can be depicted. In [11] [12] the authors relying on the class diagrams and graphs showed that in such approach there is not possible to distinguish, for example, the Strategy pattern from the State [12] and also identification of the Singleton [11] pattern. Well-known UML (and similar, e.g. OMT) used for the design patterns, is found to be semi-formal representation [9] [10]. It may result from the omission in the UML diagrams some of the aspects of pattern implementation, and also [2]: UML cannot be used to describe an infinite set of pattern instances because the language is not designed for that purpose. In few attempts [2] [3] the logic programming is used. In some of the mentioned approach used to add a new design pattern it is necessary to keep repeating the data which is common for many variants. What is more, in [10] it was noticed that adding new data is often difficult and even impossible at times. Finally both in [10] it is proved that not all the design patterns can be verified only on the basis of the rule-based programming.

3 The Model of Quality Assessment of Implementation of Design Patterns

3.1 General Assumptions

One of the testing rules says that testing depends on a context. There is no one, universal approach to testing, or model describing software or verifying implementation. The same relates to the models presented in the previous chapter - each is dedicated to a specific goal. Therefore, when the goal is assessing the quality of implementation of design patterns, one should develop a model dedicated especially to it.

Design patterns [4] are implemented in object-oriented software, therefore, a model enabling one to assess this implementation should be based (at least partially) on the paradigm of object-oriented programming. The main advantage of this approach is its intuitive and user-friendly quality to people who know the paradigm of object-oriented programming, especially to software developers, but also researchers interested in the field of software engineering. Another advantage is its independence of any specific programming language - what is important is that this language should be object-oriented. Other advantages of the proposed model result from the attempt of getting the answer about deficiency of the existing attempts, especially:

- Taking into consideration many different quality aspects connected with the design patterns implementation – so that the model is better matched to the purpose,
- Being open to new data – it provides the ability of adding new variants of design patterns without repeating the rest of the data,
- Integration of different ways of representing the information about the design patterns implementation – it allows to choose appropriate type of metrics,
- Analysing the context of the design patterns implementation – so that there is available more information than due to the structure creating the pattern.

Data included in the model enable one to: determine the quality of implementation of design patterns, improve the obtained quality through suggesting relevant amendments and predict possible software defects.

3.2 Construction of the Model

The offered model of quality assessment of implementation of design patterns is composed of the following elements:

- Formal software representation (FSR) – equivalent of source code of an analysed software which includes design patterns,

- Definitions of patterns (DEF) – characteristics, in other words, general features occurring in design patterns,
- Reference model (RM) – generic implementations of design patterns,
- Specific measures (SM) – different functions operating on measurable properties of object-oriented software.

Figure 1 shows a logical combination of elements occurring in the model. RM and SM form metric repository, while metrics repository and DEF form the repository of implementation of design patterns.

FSR shows the artefacts of object-oriented software, therefore, there are relations corresponding with the elements of the object-oriented paradigm. Classes, interfaces and data types were generalized to *Type* relation expressed with relation scheme (1). The components of classes and interfaces are relations *Field*, *Property*, *Method*, which are functionally dependent on *Type*; an example for *Property* relation is shown by relation scheme (2) and functional dependency (3). The most significant relations are shown in Table 1.

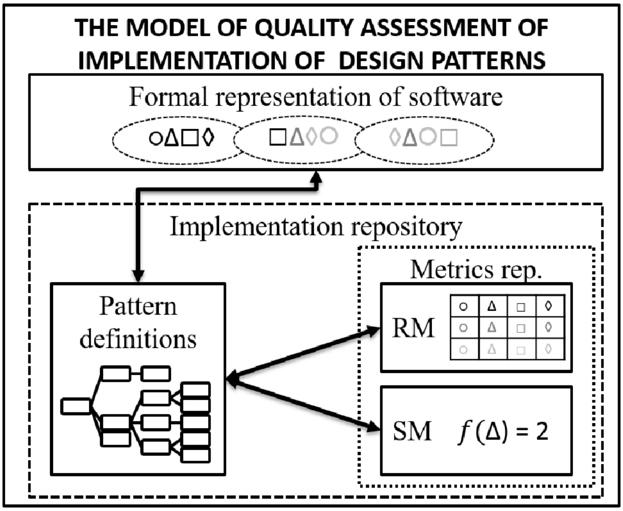

THE MODEL OF QUALITY ASSESSMENT OF IMPLEMENTATION OF DESIGN PATTERNS
Formal representation of software
○Δ□◊
□Δ◊○
◊Δ○□
Implementation repository
Metrics rep.
Pattern definitions
RM
SM $f(\Delta) = 2$

Fig. 1 General scheme of the model; geometric shapes symbolize artefacts in software.

$$S_{Type} = \{\#pk_{Type}, amod, mod, kot, name\} \tag{1}$$

$$S_{Property} = \{\#pk_{Property}, \#fk_{Property}, amod, mod, dir, name\} \tag{2}$$

$$\#fk_{Property} \rightarrow \#pk_{Type} \tag{3}$$

Where: amod ∈ AMOD = {public, private, …}, mod∈ MOD ={static, partial, …}, kot ∈ KOT = {class, interface, …}, dir ∈ DIR = {set, get, set/get}, name ∈ alphanumerical, #pk is a primary key, #fk is a foregin key.

Table 1 Relations included in FSR.

Relation name	Example	Functional dependency
Type	Classes, interfaces, vale types, generic types	——
Property	The so-called setter or getter	$\#fk_{Property} \rightarrow \#pk_{Type}$
Field	Global variables	$\#fk_{Fieldy} \rightarrow \#pk_{Type}$
Method	The so-called functions	$\#fk_{Method} \rightarrow \#pk_{Type}$
Parameter	Parameters accepted by a function	$\#fk_{Parameter} \rightarrow \#pk_{Method}$
MethodBody	Code content included in functions	$\#fk_{MethodBody} \rightarrow \#pk_{Method}$

In the repository of implementation of design patterns, one can find elements analogous to the ones found in ISO 9126, that is, characteristics equal DEF, while metrics equal metrics repository.

DEF is a certain kind of a list of contents which represents properly selected features of patterns irrespective of detailed implementation [13]. A feature of design pattern is one of essential (or optional) artefacts in pattern implementation. Features are grouped in categories within which they form a hierarchy, while a set of features of the lowest level describes a particular pattern.

Metrics repository constitutes a source of information on possible variants of implementation of design patterns. Each of the occurred metrics is classified according to the level of matching. The highest level means the best adjustment to the assumed ideal. RM structure is derived from FSR, therefore, each FSR relations has its representation in a relation in RM. The difference is that each RM relation is associated with an additional relation representing the level of matching. SM is a set of indicators (that means functions that are mapping the extract of the program by the numerical value) operating on the measurable properties of object-oriented software [13]. Both described types of metrics are polymorphic with respect to each other, that is, they can determine values for the same feature interchangeably.

3.3 Verification of Implementation Based on the Model

Generally, the verification of implementation of a design pattern consists in showing conformity of the analysed software code with rules and principles of implementation adopted for a particular pattern e.g. in [4]. The verification of implementation based on the offered model uses the metrics repository, which plays a role of a set of rules and principles of implementation, to achieve this goal. Therefore, the verification comes down to determining results of metrics describing a certain pattern. In case of RM, this consists in comparing an atomic artefact with FSR to a set of possible values included in RM which correspond with this artefact. Each value is associated with a level of matching which, simultaneously, is a result for a certain metric. Determining metrics on the basis of SM consists in performing a mathematical function measuring the properties of an artefact from FSR, and then mapping the obtained result into a set of possible values.

The results are classified in a set of values according to the ranges: from, to, exactly. Each range is associated with a relevant level of matching which is again a final result of a certain metric. Figure 2 presents the details of the discussed verification, described by means of symbols corresponding with the artefacts in FSR.

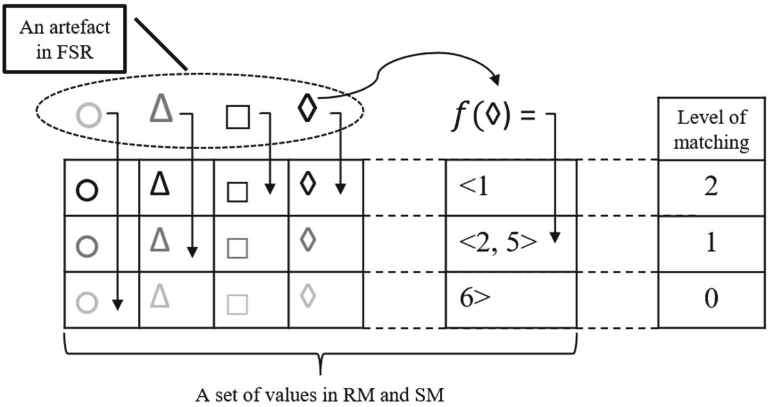


Fig. 2 Determining the level of matching - geometric shapes correspond with software artefacts.

3.4 Examples of Metrics

Most of design patterns [4] are composed of many aspects which should be taken into consideration during implementation. The basic aspect occurring in any pattern is a implementation structure constituting a pattern [13], that is, relevant classes, interfaces and dependencies between them. In this respect, Singleton seems to be the relatively least complex pattern which, from generalized perspective, is composed of three elements, after: [8]: 1 - only private constructors are allowed; 2 - there must be at least one public method (or public static final field) that has the same return type as the class it is defined in; 3 - that public method must return the same instance of the class, no matter how many times called. However, it is too much generalization to use it as a source of information required for quality assessment. Therefore, the aspect of a structure creating a pattern can be decomposed to 16 atomic elements [13]. With the atomic elements included in other aspects (e.g. use of shareholders, behaviour in multithreaded environment) added to it, one will obtain a developed set of information on implementation of Singleton pattern and this will make it possible to briefly describe all the metrics necessary to verify the whole set. Below, one can find two metrics discussed in order to present a general idea. Examples in this form are applied only with the Singleton pattern.

An exemplary metric based on RM means verification of the scope of visibility of a method which returns the instance of Singleton pattern. According to the above-mentioned generalized description of the structure constituting a pattern, a method in the analysed software should be marked with public access modifier (the highest level of matching). However, the method can be also marked with internal modifier (lower level of matching), which will reduce the scope of visibility. On the other hand, private access modifier occurrence is unacceptable here (zero level of matching). The consequences of the occurrence of three mentioned cases are described in Table 2.

Table 2 Consequences of the occurrence of the selected access modifiers.

Access modifier	Level of matching	Consequences of occurrence
public	2	Recommended, it makes Singleton instance accessible to every class, does not limit the use.
internal	1	Acceptable, still it limits the scope of visibility only to the library. It may cause problems when attempting integration with the analysed software.
private	0	Unacceptable, it will make an access to instance impossible, a pattern will not serve its purpose.

An exemplary metric based on SM for Singleton pattern is MHF from set [1] expressed in formula (4); this metric concerns the content of pattern class which is not a part of a structure composing a pattern. By substituting into formula (4) arguments for an exemplary implementation of a pattern where there is 4 private methods and 20 of all methods, one will obtain the result of 0.2. Subsequently, the result should be mapped in a set of possible values; in this case, the result meets the highest level of matching. Mapping is necessary as the recommended values of MHF for the general case of software may be different for Singleton pattern. The consequences of the occurrence of different cases are presented in Table 3.

$$MHF = \frac{\sum private\ method}{\sum method} \tag{4}$$

Table 3 A set of possible values of MHF metric.

Metric result	Level of matching	Consequences of occurrence
<0.3	2	Recommended, it means typical covering of public methods in relation to private methods.
<0.4, 0.9>	1	Acceptable, still, it means too high internal complexity of a class.
1	0	Unacceptable, it will make an access to instance impossible; the pattern will not serve its purpose.

3.5 The Practical Use

The presented model offers the basis for practical use. In [13] there were presented preliminary research results obtained with help of the dedicated software and the real life examples applying to the undertaken ways of using the model, that is the quality check, improvement and error prediction in the design pattern implementation. Improvement of the implementation is necessary in order to keep the quality of the produced software. It is beneficial for small development teams (e.g. Scrum), where the model integrates with the existing productive methodology. Improving the implementation is necessary when the design patterns implementation has no errors but it can be fixed. It is beneficial for the beginner programmers or for the implementation of not popular patterns. Prediction of the errors resulting from bad implementation of the design patterns is necessary in the terminal phase of the production, so that the software can be protected from the potential problems. Other examples of using the model: detection of the patterns instances, refactoring to patterns, automatic documentation of the pattern code or automatic implementation of the design patterns.

4 Summary

The article briefly presented the selected models of software development and ISO 9126 standard describing software quality. It also emphasized a great importance of the paradigm of object-oriented programming.

The authors' model of assessing the quality of implementation of design patterns, which is composed of elements intended for representation of software equivalent, characteristics and relevant metrics, was discussed. Previous works confirmed correctness of the accepted resolutions and are the basis for further research. On the basis of the model, one can, among others, verify the structure of implementation of design patterns and improve the implementation itself.

Further works related to the model will include providing more details concerning the model's components and dependencies between those. Next, the practical experiment is predicted, which a few programmers will take part in. In the experiment the model will be used to keep the quality and to improve the design patterns implementation.

References

1. Abreu, F.B., et al.: Toward the design quality evaluation of object-oriented software systems. In: Proceedings of the 5th International Conference on Software Quality, Austin, Texas (1995)
2. Blewitt, A.: HEDGEHOG: Automatic Verification of Design Patterns in Java. PhD Thesis, University of Edinburgh (2006)
3. Fabry, J., Mens, T.: Language-Independent Detection of Object-Oriented Design Patterns. Journal Computer Languages: Systems and Structures **30**(1-2), 21–33 (2004)
4. Gamma, E., et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, Boston (1994)
5. McConnell, S.: Code Complete: A Practical Handbook of Software Construction, 2nd edn. Microsoft Press (2004)

6. Kan, S.H.: Metrics and models in software quality engineering. Addison-Wesley Professional, Boston (2003)
7. Md. Khaer, A., et al.: An empirical analysis of software systems for measurement of design quality level based on design patterns. In: 10th International Conference on Computer and Information Technology, pp. 1–6, Dhaka (2007)
8. Kirasić, D., Basch, D.: Ontology-Based Design Pattern Recognition. Knowledge-Based Intelligent Information and Engineering Systems, Zagreb (2008)
9. Pavlic, L., et al.: Improving design pattern adoption with Ontology-Based Design Pattern Repository. Informatica an International Journal of Computing and Informatics **33**, Ljubljana (2009)
10. Rasool, G.: Customizable Feature based Design Pattern Recognition Integrating Multiple Techniques. PhD Thesis, Technische Universitat Ilmenau, Ilmenau (2010)
11. Singh Rao, R., Gupta, M.: Design Pattern Detection by Greedy Algorithm Using Inexact Graph Matching. International Journal of Engineering and Computer Science **2**(10), 3658–3664 (2013)
12. Tsantalis, N., et al.: Design Pattern Detection Using Similarity Scoring. IEEE Transactions on Software Engineering **32**(11), 896–908 (2006)
13. Wojszczyk, R.: The model and function of quality assessment of implementation of design patterns. Applied Computer Science **11**(3). Institute of Technological Systems of Information, Lublin University of Technology, Lublin (2015)

Application of Particle Swarm Optimization to Maximize Efficiency of Straight and U-Shaped Robotic Assembly Lines

Mukund Nilakantan Janardhanan, Peter Nielsen and S.G. Ponnambalam

Abstract This paper focuses on implementing particle swarm optimization (PSO) to optimize the robotic assembly line balancing (RALB) problems with an objective of maximizing line efficiency. By maximizing the line efficiency, industries tend to utilize their resources in an efficient manner. In this paper, two layout configurations of robotic assembly lines are proposed. In this robotic assembly line balancing problem, the tasks are assigned to the workstations and the efficient robots to perform the assigned tasks are chosen based on the objective of maximizing line efficiency. Performance of the proposed PSO algorithm is evaluated on benchmark problems and compared with the best known results reported in the literature. Computational time of the proposed algorithm is better than the one reported in the literature. Comparative studies on the performance of the two layouts are also done and the results are reported.

Keywords Line efficiency · Assembly line layouts · Cycle time · Particle swarm optimization

1 Introduction

Assembly Line balancing (ALB) problems mainly deal with allocation of tasks to different workstations such a way that the precedence relations of the tasks are not

M.N. Janardhanan(✉) · P. Nielsen
Department of Mechanical and Manufacturing Engineering, Aalborg University,
Aalborg, Denmark
e-mail: {mnj,peter}@m-tech.aau.dk

S.G. Ponnambalam
School of Engineering, Monash University Malaysia, 46150 Bandar Sunway, Malaysia
e-mail: sgponnambalam@monash.edu

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_56

violated. Allocation is done based on a given objective function [1]. Different researchers have worked extensively on the traditional assembly line balancing problems with different objective functions to be optimized [2]. Advancement in technologies helps industries to move away from traditional assembly line and adopt robotic based assembly lines. Robots help in improving the productivity, flexibility and provide a safe environment for the labor. Since installing an assembly line is a very cost intensive and long term, decision, it is very important to assign the tasks to the workstations in an efficient manner by balancing the workload at each workstations. Assembly lines are classified based on their models and configuration. In this paper, a single model robotic assembly line with straight and U-shaped configuration is considered. From the literature it could be analyzed that different researchers have proposed different models to balance the workloads in robotic assembly lines with an objective of minimizing cycle time, number of workstations and energy consumptions [3-5]. The two common types of RALB problems addressed in the literature are: RALB-I (Robotic Assembly Line Balancing-Type I), where the objective is to minimize the number of workstations when the cycle time is fixed and RALB-II (Robotic Assembly Line Balancing-Type II) which mainly aims at minimizing the cycle time when the number of workstations are fixed [4]. Researchers mainly focused on proposing models for straight robotic assembly line where as there is very less focus on U-shaped robotic assembly lines. Scheduling, assembly line balancing problems all are difficult optimization problems and all of them fall under the category of NP-hard. Different researchers have proposed different techniques like constraint programming, linear programming, and metaheuristics to solve these problems [6-8]. This paper aims at proposing an efficient algorithm by using particle swarm optimization (PSO) which can generate better solutions than the one reported in the literature [9] in terms of maximizing line efficiency for straight and U-shaped robotic assembly lines referred as RALB-E. In this paper, the assembly tasks are to be allocated to workstations, and each workstation needs to be allotted with a robot which performs the allotted tasks in minimum time which in turns helps in maximizing line efficiency for a fixed number of workstations for both straight and U-shaped assembly lines.

The remainder of the paper is structured as follows: Section 2 presents the problem definition along with the assumptions considered in this paper. Section 3 presents the details of the implementation of PSO. Section 4 reports the experimental results of the proposed algorithm and compared with the results reported in the literature and Section 5 concludes the findings of this research.

2 Problem Definition and Assumptions

In this paper, two configurations (straight and U-shaped) of robotic assembly lines are considered. In these configurations, different types of assembly tasks are to be allotted to a workstation where a specific type of robot will be selected to execute the allotted tasks. Tasks are to be assigned to the workstation in such a manner that the precedence constraints are met. Assignment of tasks determines the order in which tasks are to be executed. For a robotic assembly line, there are a set of workstations and robots. There is a need of efficiently balancing the workload in each workstation and to select the suitable robot to achieve an efficient balanced assembly line. Tasks are to be assigned to the workstation and the best fit robot which can perform the tasks in minimum time is to be allotted. The major objective of this problem addressed in this paper is to maximize the efficiency of the robotic assembly line. In case of straight robotic assembly line, tasks are assigned to the workstation by checking the predecessor tasks where as in U-shaped tasks are assigned by checking the predecessor as well as successor tasks. The following assumptions are considered in this work [3, 4].

- Splitting of a task into different workstations is not allowed.
- Task time is based on the type of the robot assigned to perform them.
- In a workstation, only one robot can be assigned.
- In this paper, number of workstations is equal to the number of robots.
- Any task can be executed by any robot in any workstation.
- Purchase cost of robots is not considered. There are no limitations in the availability of the robots.
- Assembly line is designed for a unique model of a single product.

Since the proposed problem falls under the category of NP-hard, there is a need of using metaheuristic search algorithm to generate efficient solutions for RALB-E. In this paper, particle swarm optimization (PSO) is implemented.

3 Design of PSO for RALB-E

Particle Swarm Optimization (PSO) algorithm is applied to solve this RALB-E problem due to its simplicity in implementation and easiness in fine tuning the parameters. It is reported in the literature that PSO is better in terms of computational speed when compared with other metaheuristic algorithms [10]. PSO is a population based metaheuristic algorithm proposed by Kennedy and Eberhart [11]. Concept of PSO was inspired by the social behavior of migrating birds trying to fly to unknown destination. In PSO, birds are referred as particles. For the adaptability to assembly line balancing problems, the particle is an ordered sequence of tasks arranged based on the precedence relationship. Based on the social behavior, a set of particles called swarm tries to move towards the best region in the search space. A flowchart of the PSO is presented in Figure 1 and different steps involved in PSO implementation are presented in the following section.

3.1 Initialization

Initial particles and velocity are randomly generated randomly. In this paper, instead of starting the search process with random particles, few heuristic rules reported in the literature are applied to generate the particles. This helps in reducing the computation time. Different heuristic rules are applied by using the task information and precedence relationship data. Task information contains the time required to complete the tasks by a set of different set of robots. For example, Table 1 presents the time data for 11 task problem with 4 problems. Each particle updates its position based on the velocity in every iteration. A sample particle structure representation and velocity vector is shown in Figure 2. In this paper, number of velocity pairs are fixed and generated randomly. For e.g., in case of 20 tasks problem then number of velocity pairs is fixed as 4. Number of initial particles is fixed to 25 and different sizes of initial population were tested and it is concluded that when it is 25 better solutions are obtained.

3.2 Velocity Update

Each particle updates it position towards the global optimum solution using the velocity vector. Two factors that guides the velocity vector: best position encountered by the particle itself (local best) and best position (global best) encountered by the whole population (swarm). The velocity is updated using Equation 1.

$$v_i^{t+1} = v_i^t + \underbrace{c_1 * [U_1 * ({}^{lo}P_i^t - P_i^t)]}_{\text{cognitive part}} + \underbrace{c_2 * [U_2 * (G^t - P_i^t)]}_{\text{social part}} \tag{1}$$

Where U_1 and U_2 are known as velocity coefficients (random numbers between 0 and 1), c_1 and c_2 are known as acceleration coefficients, v_i^t is the initial velocity, ${}^{lo}P_i^{\mathrm{t}}$ is the local best, G^t is the global best solution at generation 't' and P_i^t is the current particle position. Due to the characteristic of the problem, transposition rule is used for updating the velocity. An example is shown how the transposition works when there is a local best particle and current particle. Local best particle (${}^{lo}p_i^t$) (1, 5, 4, 2, 3, 6) and Current particle (P_i^t): (1, 3, 2, 4, 5, 6). The transposition pairs are: (1, 5, 4, 2, 3, 6) - (1, 3, 2, 4, 5, 6) = (2, 5) (3, 4). Detailed explanation on how transposition works is reported in [5]. In the velocity equation, coefficient values c_1, U_1 and c_2, U$_2$ work like a probability percentage which decides how many pairs would be copied by doing the transpositions to form the updated velocity.

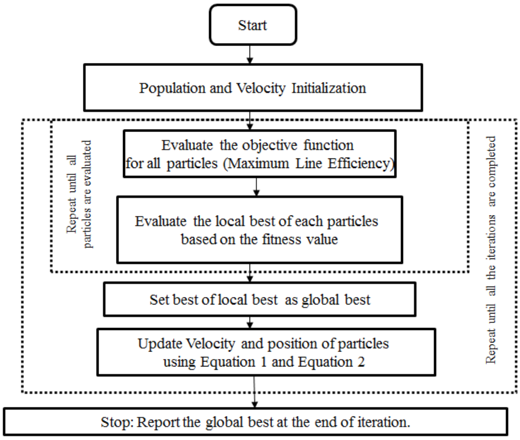

Start
Population and Velocity Initialization
Evaluate the objective function for all particles (Maximum Line Efficiency)
Evaluate the local best of each particles based on the fitness value
Repeat until all particles are evaluated
Set best of local best as global best
Update Velocity and position of particles using Equation 1 and Equation 2
Repeat until all the iterations are completed
Stop: Report the global best at the end of iteration.

Fig. 1 Flow chart of PSO

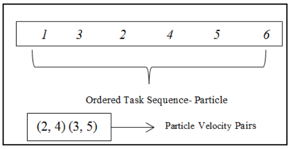


Fig. 2 Particle and Velocity structure

3.3 Position Update

Position of the particle is updated using the updated velocity. This updating helps in finding the position of the new particle using Equation 2. In this problem, it helps in finding a new particle with a new tasks sequence.

$$P_i^{t+1} = P_i^t + v_i^{t+1} \tag{2}$$

An example is presented to show how the position update works.

Current particle (P_i^t): (1, 3, 2, 4, 5, 6), Update Velocity v_i^{t+1}: (2, 5) (3, 4).
Update position: (1, 3, 2, 4, 5, 6) + (2, 5) (3, 4) = (1, 5, 4, 2, 3, 6)

Since the problem addressed in this paper is constrained by the precedence relationship, a repair mechanism [3] is incorporated to convert an infeasible particle (tasks sequence) to a feasible particle which meets the precedence relationship.

3.4 Fitness Evaluation- Line Efficiency Calculation

The objective function which is evaluated in this paper is the line efficiency *(LE)*. Line efficiency is to be calculated for both straight and U-shaped layout configurations of a robotic assembly line. The workstations are assigned with the tasks based on the precedence relationship and the robots are selected based on their ability to perform the assigned tasks in minimum time. Steps involved in evaluating the line efficiency for both straight and U-shaped layouts are presented below.

Step 1: Tasks are allotted to the workstation in straight layout based on the task sequence generated using the procedure reported in [5]. For U-shaped, allocation procedure reported in is adopted. These procedures allocate the tasks to the workstation by minimizing the workstation time. In U-shaped allocation, tasks from the either side of the tasks sequence can be allotted to the workstation. **Step 2:** Robots are selected to perform the allotted tasks. These robots are required to be executing the tasks in minimum time. **Step 3:** Based on the tasks allocation and robot allocation, the workstation time is calculated. The workstation with the maximum workstation time is the cycle time of the allocation. **Step 4:** Line efficiency *(LE)* is calculated using Equation 3. Line efficiency is the direct indication of the efficiency of a given assembly line.

$$LE = \frac{\sum_{k=1}^{N_w} S_k}{N_w * c} * 100 \tag{3}$$

Here Sk is the workstation time, Nw is total number of workstations and C is the cycle time. The line efficiency provides results in percentage from 0 to 100%. To illustrate the evaluation of the line efficiency of straight robotic assembly line for 11 tasks and 4 robot problem. Table 1 presents the time taken by four robots to perform the 11 tasks. Figure 3 presents the allocation of straight layout configuration. Based on the workstation times, line efficiency is calculated as follows. LE = (143+136+115+84)/ (4*143)*100 = 83.56%.

Similarly, the line efficiency of U-shaped robotic assembly line is evaluated. Figure 4 presents the allocation of tasks and robots to each workstation in U-shaped robotic assembly line. Line efficiency is calculated as: (116+108+121+124)/ (4*124) = 94.5%.

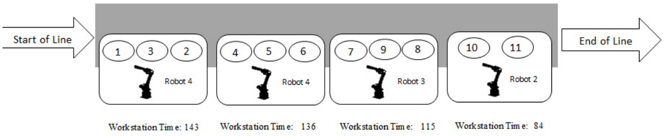


Fig. 3 Tasks and Robot allocation in straight robotic assembly line

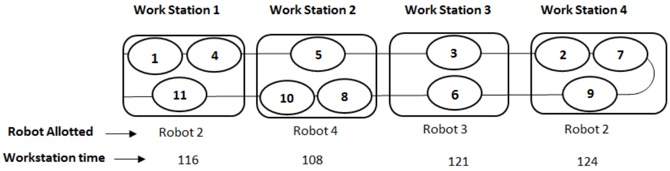


Fig. 4 Tasks and Robot allocation in U-shaped robotic assembly line

Table 1 Performance time for 11 tasks by 4 robots

Tasks	Robot 1	Robot 2	Robot 3	Robot 4	Precedence Relationship
1	81	37	51	49	-
2	109	101	90	42	1
3	65	80	38	52	1
4	51	41	91	40	1
5	92	36	33	25	1
6	77	65	83	71	2
7	51	51	40	49	3,4,5
8	50	42	34	44	6
9	43	76	41	33	7
10	45	46	41	77	8
11	76	38	83	87	9,10

4 Computational Results

The computational experiments are conducted in order to test the performance of the proposed PSO algorithm to solve RALB-E problems. Thirty two problems available in literature [4] for robotic assembly line balancing problems are used. These problems have different problems with different tasks size (25- 297 tasks) and different robot sizes. Each problem is run ten times and most of the runs converged to the same solution for all the problems. The best sets of parameters are selected after trial experiments. The parameters used here are: number of initial particles (population size) - 25, number of iterations- 30, learning coefficient values: c_1=1 and c_2=2. Results obtained by the proposed algorithm are compared with best solution (Differential Evolution-DE) [9] reported so far in the literature. The algorithm used in the literature is differential evolution. Table 2 presents the comparative results for two layouts of robotic assembly lines using PSO and DE. Problem 1-16 is small size problems and Problem 17-32 is large size problems. From the reported results it can be seen that for straight layout proposed PSO reports better solutions than DE for 13 problems in the small size category and for large size problems 11 out of 16 problems reported better efficiency than DE. For U-shaped layout, results obtained by using PSO could find better solutions for 12 problems in small size category.

Table 2 Computational Results for RALB-E problems

Dataset	**Proposed PSO**		**DE [4]**		**Dataset**	**Proposed PSO**		**DE [4]**	
	St. line	**U-shaped**	**St. line**	**U-shaped**		**St. line**	**U-shaped**	**St. line**	**U-shaped**
25-3	**97.3**	**99.1**	97.3	**99.1**	89-8	80.6	83.4	**82.3**	84.0
25-4	**97.1**	**98.0**	88.6	91.5	89-12	**96.2**	94.5	94.3	**96.0**
25-6	**90.5**	**96.9**	88.2	**96.9**	89-16	**98.8**	**98.2**	90.8	98.2
25-9	**87.4**	**89.0**	84.5	88.9	89-21	**90.6**	**90.4**	88.2	88.6
35-4	**99.4**	**98.6**	98	**98.6**	111-9	97.2	96.7	**97.8**	96.8
35-5	**95.2**	96.7	93	**97.4**	111-13	**96.1**	**95.4**	95.6	94.4
35-7	93.0	94.6	92	**94.8**	111-17	**94.0**	**93.8**	93.2	93.8
35-12	**90.5**	**90.9**	82.3	87.3	111-22	**91.7**	90.0	**91.7**	90.1
53-5	**97.5**	**98.6**	92.1	97.3	148-10	**98.0**	**97.2**	96.2	96.7
53-7	**97.8**	**95.7**	93.4	93.8	148-14	96.3	**94.2**	**96.4**	94.0
53-10	**94.5**	**94.5**	91.4	93.7	148-21	**95.2**	92.6	**94.5**	92.7
53-14	**91.2**	**92.3**	82.2	90.0	148-29	90.8	**90.2**	**92.8**	89.9
70-7	**95.9**	**97.7**	95	97.4	297-19	**97.3**	**96.7**	97.1	96.4
70-10	95.4	**95.6**	93.8	94.0	297-29	**94.2**	**94.0**	93.3	93.4
70-14	**93.2**	91.7	87.6	**92.8**	297-38	**94.2**	**94.1**	91.2	93.6
70-19	**90.7**	88.9	87.5	**89.2**	297-50	92.0	**91.0**	**92.4**	90.9

When comparing the performance of PSO in large size problems in U-shaped layout, 11 problems from the set reported better solution than DE. However, when a comparison is done between straight and U-shaped, U-shaped robotic assembly line produces better line efficiency for all the small size datasets (up to 70 task problems) when compared with straight robotic assembly line. For large size datasets, line efficiency reported using PSO is better for straight robotic assembly line when compared to the U-shaped assembly for most of the datasets in this group. The variation between the workstation times is lower when assembled through straight line; hence the line efficiency is higher for the large size dataset problems.

5 Conclusion

A Robotic Assembly Line Balancing (RALB) problem with an objective of maximizing line efficiency is presented. Two layouts of robotic assembly line are considered: Straight and U-shaped. Particle Swarm Optimization is proposed to solve the proposed problem and benchmark problems available in the open literature is used for evaluating the performance. Proposed PSO is compared with the results reported in the literature. It is found that proposed PSO could find better solutions when compared for both straight and U-shaped layouts. From the analysis it could be concluded that U-shaped layout reports higher efficiency for small size problems whereas for large size problems straight layout is better. Parameters for the algorithm are selected through a series of fine tuning experiments and best combination of parameters is reported here.

References

1. Scholl, A., Becker, C.: State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research **168**, 666–693 (2006)
2. Sivasankaran, P., Shahabudeen, P.: Literature review of assembly line balancing problems. The International Journal of Advanced Manufacturing Technology **73**, 1665–1694 (2014)
3. Levitin, G., Rubinovitz, J., Shnits, B.: A genetic algorithm for robotic assembly line balancing. European Journal of Operational Research **168**, 811–825 (2006)
4. Gao, J., Sun, L., Wang, L., Gen, M.: An efficient approach for type II robotic assembly line balancing problems. Computers & Industrial Engineering **56**, 1065–1080 (2009)
5. Nilakantan, J.M., Huang, G.Q., Ponnambalam, S.: An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. Journal of Cleaner Production **90**, 311–325 (2015)
6. Yoosefelahi, A., Aminnayeri, M., Mosadegh, H., Ardakani, H.D.: Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. Journal of Manufacturing Systems **31**, 139–151 (2012)
7. Relich, M., Muszyński, W.: The use of intelligent systems for planning and scheduling of product development projects. Procedia Computer Science **35**, 1586–1595 (2014)

8. Sitek, P.: A hybrid CP/MP approach to supply chain modelling, optimization and analysis. In: 2014 Federated Conference on Proceedings of Computer Science and Information Systems (FedCSIS), pp. 1345–1352. IEEE (2014)
9. Nilakantan, J.M., Ponnambalam, S.: Optimizing the Efficiency of Straight and U-Shaped Robotic Assembly Lines. Swarm, Evolutionary, and Memetic Computing, pp. 582–595. Springer (2014)
10. Dou, J., Li, J., Lv, Q.: A hybrid particle swarm algorithm for assembly line balancing problem of type 1. In: 2011 International Conference on Proceedings of Mechatronics and Automation (ICMA), pp. 1664–1669. IEEE (2011)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Proceedings of Neural Networks, 1995, vol. 1944, pp. 1942–1948 (1995)

3D Pallet Stacking with Rigorous Vertical Stability

Tina Sørensen, Søren Foged, Jeppe Mulbjerg Gravers, Mukund Nilakantan Janardhanan, Peter Nielsen and Ole Madsen

Abstract A number of studies have been conducted on the distributor's pallet loading problem. However, the problem has rarely been solved for and applied on a real robot. Therefore, a model for solving such problems is created with focus on the solutions being applicable in real life. The main findings of the presented research is that for large problems it is necessary to utilize other approaches than optimization such as meta-heuristics. Furthermore, some model aspects are necessary to address such as stability in order to ensure working solutions.

Keywords Three-dimensional pallet loading · Optimization · MILP · Vertical stability · Robot

1 Introduction

When a company wants to ship a great amount of items, these need to be packed securely, e.g. in containers or on pallets. All industries seek to maximize profits and minimize costs and it makes perfect sense to find ways to ship items safely whilst taking up the least amount of space. Furthermore letting a robot pack the pallet can lead to further reductions in costs and increased profits. This paper will investigate how to design a mathematical model with solutions that can be implemented on a real robot. In order to do this, one must create a stacking plan, and to do it optimally,

T. Sørensen · S. Foged · J.M. Gravers
Department of Mathematical Sciences, Aalborg University, Aalborg, Denmark
e-mail: {tsaren11,sfoged11,jgrave11}@student.aau.dk

M.N. Janardhanan(✉) · P. Nielsen · O. Madsen
Department of Mechanical and Manufacturing Engineering,
Aalborg University, Aalborg, Denmark
e-mail: {mnj,peter,om}@m-tech.aau.dk

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_57

an optimization model, such as a Mixed Integer Linear Programming Model (MILP), relating to this problem is needed.

The formal name of this problem depends on whether or not the given boxes are identical. If they are, the problem is titled the *manufacturer's pallet loading problem* (MPLP) [1], [2], [3], [4], since manufacturing companies often need to ship large amounts of the same product. Beside manufacturing companies, distribution centres also need to ship a large amount of items. The difference is that the distribution centres do not typically ship identical items but rather a mix of items, since distribution centres provide multitudes of product types and thus this problem is called the *distributor's pallet packing problem* (DPLP) [5], [6], [7]. The distributor's pallet packing problem is NP-hard [8], and so, for large problems, it is impractical to find optimal solutions and heuristics are sometimes applied to circumvent this [9]. Since the MPLP is a special case of the DPLP, and the latter is harder and involves more aspects, the paper revolves around finding a model for this type of problem. The problem consists of at least two dimensions: the length and width of the boxes. Adding a third dimension, the height of the boxes, makes this problem strongly NP-hard [2], and therefore an optimal solution is hard to find within a reasonable time frame.

As the DPLP is not a new problem, research has been conducted within this field. Below, a table of relevant research is presented. References are categorised by which type of the problem they attempt to solve.

Table 1 Literature Review.

MPLP	
Method	Reference
• ILP using a two-phase approach.	[1]
• MILP with rotation of boxes implemented by doubling the set of boxes.	[2]
• ILP using Best-First Branch & Bound with a staircase method.	[3]
• MILP with implemented vertical stability.	[4]

DPLP	
Method	Reference
• MILP with vertical stability and weight constraints, non-square boxes and multiple containers.	[5]
• MILP model considering multiple containers.	[6]
• MILP with vertical and horizontal stability.	[7]

All the papers in the above table address rotation and no overlapping and pallets are used unless stated otherwise. In the papers that consider vertical stability, it is either implemented as a percentage of the bottom area of the box that must be supported by boxes below [4], as all of the bottom area of a box being supported below [7], or by forcing three corners of a box to be supported from below [5]. The first and the latter of these approaches are flawed, since they assume that the gravitational midpoint of a box lies in the geometrical centre of it or is directly above the boxes supporting it.

Reviewing literature, many disregard the fact that if the models are to be of use, they need to be able to be implemented in practice, by e.g. a robot. Therefore the scope of this paper is to create an optimization model that can effectively create a solution which can then be realized by a palletizing robot.

The approach also underlines the NP-hard nature of the problem and how this kind of issue must be solved using heuristics to be applicable in a real multimodal transportation network. The remainder of the paper is organized as follows. In Section 2, the formulation of the MILP where overlapping of boxes is avoided is presented. In Section 3, the model is extended to include vertical stability. In Section 4 the work with the robot is presented. Lastly, in Section 5 the results are presented and discussed.

2 Mathematical Model

The model is formulated as a problem of maximizing utilized space on the pallet, given the fact that no boxes must overlap and that the boxes that are placed on the pallet cannot exceed the pallet dimensions. The model is inspired by the models reported in [7] and [5], where constraints similar to those in [7] are applied to ensure no overlapping occurs and that boxes are placed within the confines of the pallet. Simultaneously, stability constraints are introduced from [5]. However, unlike the work of [5], this model requires that all four corners of a box are supported if said box is to be placed on top of others. This way, no assumptions about the gravitational midpoints of boxes are required and the optimization results can be readily implemented. Stability constraints are presented in Section 3. The following variables are required to be known before the problem is solved:

2.1 Input

Types $=$ Number of different box types.

$\mathrm{B}_i =$ Set containing indexes of boxes of type i, $i \in \{1...\text{Types}\}$.

$n_T =$ The total number of boxes.

$n_R =$ A range from $1..n_T$.

$L =$ The length of the pallet along the x-axis.

$W =$ The width of the pallet along the y-axis.

$H =$ The maximum allowed stacking height on the pallet along the z-axis.

$l_i, w_i, h_i =$ The length, width, and height of box type i, $i \in n_R$.

$X, Y, Z =$ The x, y, and z-coordinate of the front left corner of the pallet.

$M =$ A sufficiently large number that is used to balance constraints.

The front left corner of the pallet is defined as the corner of the pallet that has the lowest x and y values.

2.2 Decision Variables

The model includes the following decision variables.

$x_i, y_i, z_i =$ The x, y, and z-value of the lower front left corner of box i, $i \in n_R$.
$x_i', y_i', z_i' =$ The x, y, and z-value of the upper back right corner of box i, $i \in n_R$.

$$P_i = \begin{cases} 1, & \text{if the } i\text{'th box is on the pallet.} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

$$r_{i_0} = \begin{cases} 1, & \text{if box } i \text{ is not rotated.} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$$r_{i_{90}} = \begin{cases} 1, & \text{if box } i \text{ is rotated 90 degrees.} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

$$a_{i,j} = \begin{cases} 1, & \text{if box } i \text{ is entirely to the left of box } j \text{, that is: } x_i' \leq x_j. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$b_{i,j} = \begin{cases} 1, & \text{if box } i \text{ is entirely to the right of box } j \text{, that is: } x_i \geq x_j'. \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

$$c_{i,j} = \begin{cases} 1, & \text{if box } i \text{ is entirely in front of box } j \text{, that is: } y_i' \leq y_j. \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

$$d_{i,j} = \begin{cases} 1, & \text{if box } i \text{ is entirely behind box } j \text{, that is: } y_i \geq y_j'. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$$e_{i,j} = \begin{cases} 1, & \text{if box } i \text{ is entirely below box } j \text{, that is: } z_i' \leq z_j. \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$$f_{i,j} = \begin{cases} 1, & \text{if box } i \text{ is entirely over box } j \text{, that is: } z_i \geq z_j'. \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

For a graphic illustration of the position variables see Figure 1.

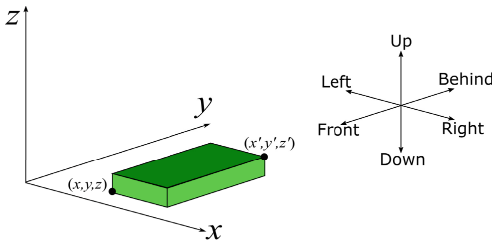


Fig. 1 Position Variables.

2.3 Objective Function and Constraints

From this, the problem is as follows:

$$\text{maximize } \textstyle\sum_{i \in n_R} l_i \cdot w_i \cdot h_i \cdot P_i. \tag{10}$$

subject to

$$x_i \geq X \cdot P_i \ \forall i \in n_R. \tag{11}$$

$$y_i \geq Y \cdot P_i \ \forall i \in n_R. \tag{12}$$

$$z_i \geq Z \cdot P_i \ \forall i \in n_R. \tag{13}$$

$$x'_i \leq X + L \ \forall i \in n_R. \tag{14}$$

$$y'_i \leq Y + W \ \forall i \in n_R. \tag{15}$$

$$z'_i \leq Z + H \ \forall i \in n_R. \tag{16}$$

$$x'_i = r_{i_0} \cdot l_i + r_{i_{90}} \cdot w_i + x_i \ \forall i \in n_R. \tag{17}$$

$$y'_i = r_{i_{90}} \cdot w_i + r_{i_0} \cdot l_i + y_i \ \forall i \in n_R. \tag{18}$$

$$z'_i = h_i + z_i \ \forall i \in n_R. \tag{19}$$

$$1 = r_{i_0} + r_{i_{90}} \ \forall i \in n_R. \tag{20}$$

$$x'_i \leq x_j + (1 - a_{ij})M \ \forall i \in n_R,\ j > i. \tag{21}$$

$$x'_j \leq x_i + (1 - b_{ij})M \ \forall i \in n_R,\ j > i. \tag{22}$$

$$y'_i \leq y_j + (1 - c_{ij})M \ \forall i \in n_R,\ j > i. \tag{23}$$

$$y'_j \leq y_i + (1 - d_{ij})M \ \forall i \in n_R,\ j > i. \tag{24}$$

$$z'_i \leq z_j + (1 - e_{ij})M \ \forall i \in n_R,\ j > i \tag{25}$$

$$z'_j \leq z_i + (1 - f_{ij})M \ \forall i \in n_R,\ j > i. \tag{26}$$

$$x'_i \geq x_j + 1 - a_{ij}M \ \forall i \in n_R,\ j > i. \tag{27}$$

$$x'_j \geq x_i + 1 - b_{ij}M \ \forall i \in n_R,\ j > i. \tag{28}$$

$$y'_i \geq y_j + 1 - c_{ij}M \ \forall i \in n_R,\ j > i \tag{29}$$

$$y'_j \geq y_i + 1 - d_{ij}M \ \forall i \in n_R,\ j > i. \tag{30}$$

$$z'_i \geq z_j + 1 - e_{ij}M \ \forall i \in n_R,\ j > i. \tag{31}$$

$$z'_j \geq z_i + 1 - f_{ij}M \ \forall i \in n_R,\ j > i. \tag{32}$$

$$1 \leq a_{ij} + b_{ij} + c_{ij} + d_{ij} + e_{ij} + f_{ij} \quad \forall i \in n_R,\ j > i. \tag{33}$$

$$P_j \geq P_{j+1} \ \forall j \in \mathrm{B}_i, \forall i \in \{1\text{...Types}\}. \tag{34}$$

$$0 \leq x_i, y_i, z_i, x'_i, y'_i, z'_i \tag{35}$$

The objective function (10) maximizes utilized space on the pallet. Equations (11) to (16) ensure that all boxes placed on the pallet lie completely within the boundaries of the pallet. Note that X, Y, and Z need to be defined sufficiently large to make space for the boxes not placed on the pallet. Equations (17) to (20) enables the option of 90° rotation around the z-axis. If r_{i_0} equals one, it means that the length of a box is parallel to the x-axis. Similarly, if $r_{i_{90}}$ is one, the length is parallel to the y-axis. Only one of these instances can occur at any one time. Constraints (21) to (33) ensure that no boxes, not even those that are not on the pallet, overlap each other. As an example, equation (21) includes the indicator variable a_{ij} which is equal to 1 if and only if box i is entirely to the left of box j, that is $x'_i \leq x_j$. Equation (21) will force a_{ij} to be 0 when box i is not to the left of box j but not force it to be 1 when box i is entirely to the left of box j. The latter is however ensured in equation (27). Equation (33) ensures that at least one of $a, b, c, \ldots, f$ is equal to 1, hence no boxes will overlap.

Constraint (34) is redundant in terms of reducing the feasible region of the problem but is added to improve computation time. It ensures that if a certain box is excluded from the solution, there will be no further searching for a solution including boxes of the same type that has not been placed yet.

3 Stability

To ensure that the stacking plan is vertically stable, that is, boxes are not allowed to be placed in a floating position with no support of its bottom surface, a constraint is added to the model. This constraint forces all corners of a box (excluding boxes that are placed on the ground level or not placed on the pallet) to be supported. For this the following variables and constraints are added:

3.1 Decision Variables

$$\beta_{ilj} = \begin{cases} 1, & \text{If corner } l \text{ of box } i \\ & \text{is supported by box } j. \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathrm{g}_i = \begin{cases} 1, & \text{If box } i \text{ stands on the ground} \\ & \text{level of the pallet, i.e. } z_i = Z. \\ 0, & \text{otherwise.} \end{cases}$$

$$\beta_{ilj} = \begin{cases} 1, & \text{If corner } l \text{ of box } i \\ & \text{is supported by box } j. \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathrm{g}_i = \begin{cases} 1, & \text{If box } i \text{ stands on the ground} \\ & \text{level of the pallet, i.e. } z_i = Z. \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathrm{SH}_{ij} = \begin{cases} 0, & \text{If box } j \text{ is on a sufficient level} \\ & \text{to support box } i\text{, i.e. } z_i = z'_j. \\ 1, & \text{otherwise.} \end{cases}$$

$$\mathrm{p}_{ij} = \begin{cases} 0, & \text{If the projection on the} \\ & xy\text{-plane of boxes } i \text{ and } j \\ & \text{have a non-empty intersection.} \\ 1, & \text{otherwise.} \end{cases}$$

$$s_{ij} = \begin{cases} 1, & \text{If box } j \text{ has the right conditions} \\ & \text{ to support box } i, \\ & \text{i.e. } \mathrm{p}_{ij} = \mathrm{SH}_{ij} = 0. \\ 0, & \text{otherwise.} \end{cases}$$

$$\eta^1_{ij} = \begin{cases} 0, & \text{If } x_j \le x_i. \\ 1, & \text{otherwise.} \end{cases}$$

$$\eta^2_{ij} = \begin{cases} 0, & \text{If } y_j \le y_i. \\ 1, & \text{otherwise.} \end{cases}$$

$$\eta^3_{ij} = \begin{cases} 0, & \text{If } x'_i \le x'_j. \\ 1, & \text{otherwise.} \end{cases}$$

$$\eta^4_{ij} = \begin{cases} 0, & \text{If } y'_i \le y'_j. \\ 1, & \text{otherwise.} \end{cases}$$

3.2 Constraints

From the decision variables, stability constraints are defined as:

$$\sum_{j\in n_R}\sum_{l=1}^{4}\beta_{ilj} \geq 4(1-\mathrm{g}_i) \ \ \forall i \in n_R \tag{36}$$

$$z_i \leq Z + (1-\mathrm{g}_i)\cdot H \ \ \forall i \in n_R \tag{37}$$

$$v_{ij} \geq z'_j - z_i \ \ \forall i, j \in n_R \tag{38}$$

$$v_{ij} \geq z_i - z'_j \ \ \forall i, j \in n_R \tag{39}$$

$$\mathrm{SH}_{ij} \leq v_{ij} \ \ \forall i, j \in n_R \tag{40}$$

$$v_{ij} \leq \mathrm{SH}_{ij}\cdot H \ \ \forall i, j \in n_R \tag{41}$$

$$\mathrm{p}_{ij} \leq a_{ij} + b_{ij} + c_{ij} + d_{ij} \ \ \forall i \in n_R, j > i \tag{42}$$

$$2\mathrm{p}_{ij} \geq a_{ij} + b_{ij} + c_{ij} + d_{ij} \ \ \forall i \in n_R, j > i \tag{43}$$

$$\mathrm{p}_{ij} = \mathrm{p}_{ji} \ \ \forall i, j \in n_R \tag{44}$$

$$1 - s_{ij} \leq \mathrm{SH}_{ij} + \mathrm{p}_{ij} \ \ \forall i, j \in n_R \tag{45}$$

$$(1 - s_{ij})\cdot 2 \geq \mathrm{SH}_{ij} + \mathrm{p}_{ij} \ \ \forall i, j \in n_R \tag{46}$$

$$\beta_{ilj} \leq s_{ij} \ \ \forall i, j \in n_R, \ \ \forall l \in 1..4 \tag{47}$$

$$\beta_{ilj} \leq P_i \ \ \forall i, j \in n_R, \ \ \forall l \in 1..4 \tag{48}$$

$$\beta_{ilj} \leq P_j \ \ \forall i, j \in n_R, \ \ \forall l \in 1..4 \tag{49}$$

$$\eta^1_{ij} + \eta^2_{ij} \leq 2\cdot(1-\beta_{i1j}) \ \ \forall i, j \in n_R \tag{50}$$

$$\eta^2_{ij} + \eta^3_{ij} \leq 2\cdot(1-\beta_{i2j}) \ \ \forall i, j \in n_R \tag{51}$$

$$\eta^3_{ij} + \eta^4_{ij} \leq 2\cdot(1-\beta_{i3j}) \ \ \forall i, j \in n_R \tag{52}$$

$$\eta^1_{ij} + \eta^4_{ij} \leq 2\cdot(1-\beta_{i4j}) \ \ \forall i, j \in n_R \tag{53}$$

$$x_j \leq x_i + \eta^1_{ij}\cdot L \ \ \forall i, j \in n_R \tag{54}$$

$$y_j \leq y_i + \eta^2_{ij}\cdot W \ \ \forall i, j \in n_R \tag{55}$$

$$x'_i \leq x'_j + \eta^3_{ij}\cdot L \ \ \forall i, j \in n_R \tag{56}$$

$$y'_i \leq y'_j + \eta^4_{ij}\cdot W \ \ \forall i, j \in n_R \tag{57}$$

By constraint (36), all boxes that are not placed on the ground level must be supported at all four bottom corners. (37) ensures that g_i is set to zero as long as box i is not placed on the pallet floor. Constraints (38) and (39) are added to establish v_{ij}, which is a number that is at least as large as $|z_i - z'_j|$. This is used to define SH_{ij} i.e if $v_{ij} = 0$ then $\mathrm{SH}_{ij} = 0$ (40), and if $v_{ij} > 0$ then $\mathrm{SH}_{ij} = 1$ (41). Constraints (42) and (43) establish p_{ij} in an upper triangular matrix ($j > i$) which shows if the projections on the xy-plane of box i and j overlap, and (44) fills out the lower triangle of the symmetrical matrix, since all entries of it are needed in the constraints (45) and (46). (45) forces s_{ij} to be 1 if box j is on a sufficient level to support box i and there is an overlap in their projections on the xy-plane. (46) forces s_{ij} to equal 0 if not both these criteria are satisfied and constraint (47) ensures that β_{ilj} is 0 for all $l \in 1..4$ as long as s_{ij} is zero. (48) and (49) ensure that box i cannot be supported by j if they are not both placed on the pallet. The remaining constraints determine exactly which boxes support which corners of other boxes.

4 Use of the Model

To show the functionality of the model, a simple example has been calculated. Two box types are considered, the first being 19 cm long, 12.7 cm wide, and 9.2 cm high, and the latter being 19 cm long, 25.4 cm wide, and 9.2 cm high. There are eight of the first box type and two of the second. The pallet is 38 cm in length, 25.4 in width, and has a height limit of 27.6. The model solved it to optimality within 7.51 seconds with an Intel(R) Quad Core i7- 4700MQ processor, 2.4 GHZ CPU, 8 GB RAM and a 64 bit OS, placing four small boxes as the bottom layer, the two large ones are both rotated and form the middle layer, and the top layer is identical to the bottom layer. However, as can be seen from the results shown in Figures 2 and 3, the designed model is only useful for relatively small problems. In these figures, the number of

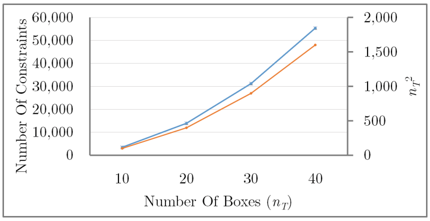


Fig. 2 Number of Constraints.

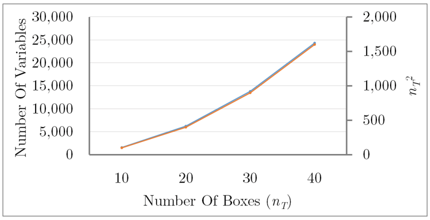


Fig. 3 Number of Variables.

variables and constraints along with number of boxes squared are shown for four problems with a gradually increasing number of boxes.

The blue lines in Figures 2 and 3 show the number of variables and constraints for 10, 20, 30, and 40 boxes. The orange line is given by the function $y = x^2$, where x is the number of boxes. Note that the blue lines follow the left hand axis and the orange line the right hand axis. Both the number of variables and the number of constraints are almost proportional to this function, meaning that the number of variables and constraints grows quadratically with the number of boxes. However, small scale experiments with the model were carried out on a FANUC M-iB 6S. The end tool that is used is a suction disc. The model is programmed in IBM's ILOG CPLEX v. 12.6.2. and in this, a heuristic is implemented that arranges the solution in the right stacking order. This first sorts for the lowest z-value, then the largest x-value, and last the largest y-value. Trials with this heuristic have proven successful in preventing the robot, and any box it may be carrying, from coming in contact with already placed boxes. In conclusion, the robot performed satisfactory in stacking the boxes, even at speeds of 2000 mm/second. Furthermore it was confirmed that the stability criteria implemented ensured vertical stability in every layer. Further experiments with the proposed model can be found in [10].

5 Discussion and Concluding Remarks

While optimality is to be strived for, this MILP model fails to give it within a reasonable time frame for larger problems. Given enough boxes, the search space becomes unmanageable. Meta-heuristics would be preferred in these cases, even though they do not guarantee optimal solutions. However, meta-heuristics have the benefit that they can solve large problems fast, making them a more flexible than the MILP model. An important observation to make of this study is that for any model to be useful, the stacking process should be taken into account when designing the model. In other words, when a robot is required to stack the boxes, its weaknesses should be taken into account. To the best of the authors' knowledge, very few papers treat stability as rigorously as in this paper. Moreover, in the papers that do include stability presented in Section 1, many assume that the gravitational midpoint is always in the centre of the box, which is an unrealistic assumption. Even though this model has proven less useful for larger problems, it may be possible to establish rules about input based on experiments with the model that will reduce computation time.

References

1. Wu, K., Ting, C.: A two-phase algorithm for the manufacturers pallet loading problem. In: 2007 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1574–1578. IEEE (2007)
2. Al-Shayea, A.M., et al.: Solving the three-dimensional palet-paking problem using mixed 0–1 model. Journal of Service Science and Management **4**, 513 (2011)
3. Ahn, S., Park, C., Yoon, K.: An improved best-first branch and bound algorithm for the pallet-loading problem using a staircase structure. Expert Systems with Applications (2015)
4. Kocjan, W., Holmström, K.: Generating stable loading patterns for pallet loading problems. In: The Fifth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, CPAIOR 2008 (2008)
5. Paquay, C., Schyns, M., Limbourg, S.: A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. International Transactions in Operational Research (2014)
6. Chen, C., Lee, S.M., Shen, Q.: An analytical model for the container loading problem. European Journal of Operational Research **80**, 68–76 (1995)
7. Junqueira, L., Morabito, R., Yamashita, D.S., Yanasse, H.H.: Optimization models for the three-dimensional container loading problem with practical constraints. In: Modeling and Optimization in Space Engineering, pp. 271–293. Springer (2013)
8. Yaman, H., Şen, A.: Manufacturers mixed pallet design problem. European Journal of Operational Research **186**, 826–840 (2008)
9. Schuster, M., Bormann, R., Steidl, D., Reynolds-Haertle, S., Stilman, M.: Stable stacking for the distributor's pallet packing problem. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3646–3651. IEEE (2010)
10. Tina, S., Sren, F., Jeppe Mulbjerg, G., Peter, N., Mukund, N.J.: Input analysis of the distributor's pallet loading problem. (Submitted to the 13th International Conference on Distributed Computing and Artificial Intelligence 2016 to be held at Sevilla, Spain.)

Input Analysis of the Distributor's Pallet Loading Problem

Tina Sørensen, Søren Foged, Jeppe Mulbjerg Gravers, Mukund Nilakantan Janardhanan and Peter Nielsen

Abstract Numerous studies have been conducted on the distributor's pallet loading problem (DPLP) in order to find solution methods that are time efficient and produces results that are applicable in the real world. It is well known, that the complexity of the problem increases by the number of boxes to be packed on the pallet, but not much research focuses on other factors of input affecting the complexity. This paper proposes a model for solving the three-dimensional single pallet DPLP. Datasets are created specifically to conduct selected experiments to identify causes to increased computational time. The results yield a strong link between computation time and certain ratios of total volume of boxes to maximum capacity of the pallet as well as the amount of small vs. large boxes to be packed.

Keywords Three-dimensional pallet loading · MILP · The distributor's pallet loading problem · Optimization

1 Introduction

In an ever increasing effort to be more competitive, companies seek to minimize costs in all facets of their business. One of these facets is the transportation of goods. Being able to load a pallet or container optimally not only reduces shipping costs, but also helps to ensure a compact and stable packing of the pallet, ultimately making for a safer transport. The *distributor's pallet packing problem* (DPLP) is the problem

T. Sørensen · S. Foged · J.M. Gravers
Department of Mathematical Sciences, Aalborg University, Aalborg, Denmark
e-mail: {tsaren11,sfoged11,jgrave11}@student.aau.dk

M.N. Janardhanan(✉) · P. Nielsen
Department of Mechanical and Manufacturing Engineering,
Aalborg University, Aalborg, Denmark
e-mail: {mnj,peter}@m-tech.aau.dk

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_58

of packing rectangular boxes of different dimensions onto a pallet in an optimal way, and is a relevant problem in many industries such as distribution centres that must load pallets with a mix of products. There are several constraints in DPLP's that make the problem complex. First, the boundaries of the pallet must not be violated i.e. every box placed on the pallet has to lie completely within the dimensions of it. Furthermore one needs to make sure that no boxes overlap physically. The stability of the pallet is another important aspect. Vertical stability refers to the fact that each box must be sufficiently supported on its bottom surface. Horizontal stability relates to the ability of a box to withstand the inertia of its own body when the pallet is in motion. Paquay et al. [1] argue that that horizontal stability is of less importance since bounding the boxes or adding special sheets between boxes to increase friction is an easy solution to the problem. Likewise (however not as simple) ensuring vertical stability can be ensured by filling unutilized space between boxes with blocks of e.g. polystyrene. Another aspect is the weight distribution on the pallet, where the target center of gravity of the pallet is as low and as close to the center as possible. The center of gravity of each box plays a role in all these stability constraints. Of course there is also the a maximum weight capacity of the pallet that must be taken into account as well as the fragility of each separate box.

The three-dimensional DPLP is a special case of the three-dimensional pallet loading problem. In pallet loading problems one further distinguishes between single- and multi-pallet loading problems. Chen et al. [2] presents a *mixed integer linear programming* (MILP) model that is capable of solving the multi-pallet loading problem in three dimensions to optimality. The model is however quite basic and lacks the inclusion of stability and fragility constraints. Paquay et al. [1] solve the same kind of problem with a model that includes both vertical stability-, weight-, and a simple form of fragility-constraints and furthermore uses a container that is not rectangular shaped. These constraints add great amount of complexity to the problem and the model struggles to find an optimal solution within their time limit of 3600 seconds when the number of boxes exceeds 20. Junqueira et al. [3] propose an MILP model that features both vertical and horizontal stability, but for problems exceeding 20 boxes the model is generally unable to find an optimal solution within 3600 seconds. The three-dimensional DPLP is strongly NP-hard [4] and therefore practically impossible to solve to optimality for larger problem instances. A recurring message in most literature is that heuristic methods are necessary when the problem increases in size. Examples of this are seen in [5], where a simple heuristic method is presented with promising computation times and [6] who uses a Genetic Algorithm approach.

This paper proposes a MILP model for solving the single pallet DPLP. The model includes constraints to ensure that no overlapping of boxes occurs, boxes placed on the pallet do not violate the dimensions of the pallet and finally vertical stability. By running this model on input of different characteristics the aim is to examine what makes a problem computational complex.

2 The Mathematical Model

The model below is a simplification of the model presented in [7], for an elaboration of this please refer to the aforementioned article. It is formulated as a problem of maximizing utilized space on the pallet and takes the following variables as input:

2.1 Input

Types = Number of different box types.
B_i = Set containing indexes of boxes of type i, $i \in \{1\text{...Types}\}$.
n_T = The total number of boxes.
n_R = A range from $1..n_T$.
L = The length of the pallet along the x-axis.
W = The width of the pallet along the y-axis.
H = The maximum allowed stacking height on the pallet along the z-axis.
l_i, w_i, h_i = The length, width and height of box type i, $i \in n_R$.
X, Y, Z = The x, y, and z-coordinate of the front left corner of the pallet.
M = A sufficiently large number that is used to balance constraints.

The front left corner of the pallet is defined as the corner with the lowest x and y values.

2.2 Decision Variables

The model includes the following decision variables:

x_i, y_i, z_i = The x, y, and z-value of the lower front left corner of box i, $i \in n_R$.
x_i', y_i', z_i' = The x, y, and z-value of the upper back right corner of box i, $i \in n_R$.

$$P_i = \begin{cases} 1, & \text{if the } i\text{'th box is on the pallet.} \\ 0, & \text{otherwise.} \end{cases}$$

$$a_{i,j} = \begin{cases} 1, & \text{if } x'_i \le x_j. \\ 0, & \text{otherwise.} \end{cases}$$

$$b_{i,j} = \begin{cases} 1, & \text{if } x_i \ge x'_j. \\ 0, & \text{otherwise.} \end{cases}$$

$$c_{i,j} = \begin{cases} 1, & \text{if } y'_i \le y_j. \\ 0, & \text{otherwise.} \end{cases}$$

$$d_{i,j} = \begin{cases} 1, & \text{if } y_i \ge y'_j. \\ 0, & \text{otherwise.} \end{cases}$$

$$e_{i,j} = \begin{cases} 1, & \text{if } z'_i \le z_j. \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{i,j} = \begin{cases} 1, & \text{if } z_i \ge z'_j. \\ 0, & \text{otherwise.} \end{cases}$$

$$\beta_{ilj} = \begin{cases} 1, & \text{If corner } l \text{ of box } i \text{ is supported by box } j. \\ 0, & \text{otherwise.} \end{cases}$$

$$g_i = \begin{cases} 1, & \text{If box } i \text{ stands on the ground level of the pallet, i.e. } z_i = Z. \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathrm{SH}_{ij} = \begin{cases} 0, & \text{If box } j \text{ is on a sufficient level to support box } i \text{, i.e. } z_i = z'_j. \\ 1, & \text{otherwise.} \end{cases}$$

$$\mathrm{p}_{ij} = \begin{cases} 0, & \text{If the projection on the } xy\text{-plane of boxes } i \text{ and } j \text{ have a non-empty intersection.} \\ 1, & \text{otherwise.} \end{cases}$$

$$s_{ij} = \begin{cases} 1, & \text{If box } j \text{ has the right conditions to support box } i \text{, i.e. } \mathrm{p}_{ij} = \mathrm{SH}_{ij} = 0. \\ 0, & \text{otherwise.} \end{cases}$$

$$\eta^1_{ij} = \begin{cases} 0, & \text{If } x_j \le x_i. \\ 1, & \text{otherwise.} \end{cases}$$

$$\eta^2_{ij} = \begin{cases} 0, & \text{If } y_j \le y_i. \\ 1, & \text{otherwise.} \end{cases}$$

$$\eta^3_{ij} = \begin{cases} 0, & \text{If } x'_i \le x'_j. \\ 1, & \text{otherwise.} \end{cases}$$

$$\eta^4_{ij} = \begin{cases} 0, & \text{If } y'_i \le y'_j. \\ 1, & \text{otherwise.} \end{cases}$$

Note that variables $a, b, c, \ldots, f$ indicate the boxes' relative positions, and are used to avoid overlaps between boxes. The same goes for $\eta^1, \ldots, \eta^4$ which are used in the constraints ensuring vertical stability.

2.3 Objective Function and Constraints

The problem is stated as follows:

$$\text{maximize } \textstyle\sum_{i \in n_R} l_i \cdot w_i \cdot h_i \cdot P_i. \tag{1}$$

subject to

$$x_i \geq X \cdot P_i \ \forall i \in n_R. \tag{2}$$

$$y_i \geq Y \cdot P_i \ \forall i \in n_R. \tag{3}$$

$$z_i \geq Z \cdot P_i \ \forall i \in n_R. \tag{4}$$

$$x'_i \leq X + L \ \forall i \in n_R. \tag{5}$$

$$y'_i \leq Y + W \ \forall i \in n_R. \tag{6}$$

$$z'_i \leq Z + H \ \forall i \in n_R. \tag{7}$$

$$x'_i = l_i + x_i \ \forall i \in n_R. \tag{8}$$

$$y'_i = w_i + y_i \ \forall i \in n_R. \tag{9}$$

$$z'_i = h_i + z_i \ \forall i \in n_R. \tag{10}$$

$$x'_i \leq x_j + (1 - a_{ij})M \ \forall i \in n_R, \ j > i. \tag{11}$$

$$x'_j \leq x_i + (1 - b_{ij})M \ \forall i \in n_R, \ j > i. \tag{12}$$

$$y'_i \leq y_j + (1 - c_{ij})M \ \forall i \in n_R, \ j > i. \tag{13}$$

$$y'_j \leq y_i + (1 - d_{ij})M \ \forall i \in n_R, \ j > i. \tag{14}$$

$$z'_i \leq z_j + (1 - e_{ij})M \ \forall i \in n_R, \ j > i \tag{15}$$

$$z'_j \leq z_i + (1 - f_{ij})M \ \forall i \in n_R, \ j > i. \tag{16}$$

$$x'_i \geq x_j + 1 - a_{ij}M \ \forall i \in n_R, \ j > i. \tag{17}$$

$$x'_j \geq x_i + 1 - b_{ij}M \ \forall i \in n_R, \ j > i. \tag{18}$$

$$y'_i \geq y_j + 1 - c_{ij}M \ \forall i \in n_R, \ j > i. \tag{19}$$

$$y'_j \geq y_i + 1 - d_{ij}M \ \forall i \in n_R, \ j > i. \tag{20}$$

$$z'_i \geq z_j + 1 - e_{ij}M \ \forall i \in n_R, \ j > i. \tag{21}$$

$$z'_j \geq z_i + 1 - f_{ij}M \ \forall i \in n_R, \ j > i. \tag{22}$$

$$1 \leq a_{ij} + b_{ij} + c_{ij} + d_{ij} + e_{ij} + f_{ij} \quad \forall i \in n_R, \ j > i. \tag{23}$$

$$P_j \geq P_{j+1} \ \forall j \in \mathrm{B}_i, \forall i \in \{1...\mathrm{Types}\}. \tag{24}$$

$$0 \leq x_i, y_i, z_i, x'_i, y'_i, z'_i \tag{25}$$

The objective function (1) maximizes utilized space on the pallet. Equations (2) to (7) ensure that all boxes placed on the pallet lie completely within the boundaries of the pallet. Note that X, Y, and Z need to be defined sufficiently large to make space for the boxes not placed on the pallet. Constraints (11) to (23) ensure that no boxes overlap each other. Constraint (24) is redundant in terms of reducing the feasible region of the problem but is added to improve computation time. It ensures that if a certain box is excluded from the solution, there will be no further searching for a solution including boxes of the same type that has not been placed yet. Stability constraints are defined as follows:

$$\sum_{j \in n_R} \sum_{l=1}^{4} \beta_{ilj} \geq 4(1 - g_i) \ \ \forall i \in n_R \tag{26}$$

$$z_i \leq Z + (1 - g_i) \cdot H \ \ \forall i \in n_R \tag{27}$$

$$v_{ij} \geq z'_j - z_i \ \ \forall i, j \in n_R \tag{28}$$

$$v_{ij} \geq z_i - z'_j \ \ \forall i, j \in n_R \tag{29}$$

$$\mathrm{SH}_{ij} \leq v_{ij} \ \ \forall i, j \in n_R \tag{30}$$

$$v_{ij} \leq \mathrm{SH}_{ij} \cdot H \ \ \forall i, j \in n_R \tag{31}$$

$$\mathrm{p}_{ij} \leq a_{ij} + b_{ij} + c_{ij} + d_{ij} \ \ \forall i \in n_R, j > i \tag{32}$$

$$2\mathrm{p}_{ij} \geq a_{ij} + b_{ij} + c_{ij} + d_{ij} \ \ \forall i \in n_R, j > i \tag{33}$$

$$\mathrm{p}_{ij} = \mathrm{p}_{ji} \ \ \forall i, j \in n_R \tag{34}$$

$$1 - s_{ij} \leq \mathrm{SH}_{ij} + \mathrm{p}_{ij} \ \ \forall i, j \in n_R \tag{35}$$

$$(1 - s_{ij}) \cdot 2 \geq \mathrm{SH}_{ij} + \mathrm{p}_{ij} \ \ \forall i, j \in n_R \tag{36}$$

$$\beta_{ilj} \leq s_{ij} \ \ \forall i, j \in n_R, \ \ \forall l \in 1..4 \tag{37}$$

$$\beta_{ilj} \leq P_i \ \ \forall i, j \in n_R, \ \ \forall l \in 1..4 \tag{38}$$

$$\beta_{ilj} \leq P_j \ \ \forall i, j \in n_R, \ \ \forall l \in 1..4 \tag{39}$$

$$\eta^1_{ij} + \eta^2_{ij} \leq 2 \cdot (1 - \beta_{i1j}) \ \ \forall i, j \in n_R \tag{40}$$

$$\eta^2_{ij} + \eta^3_{ij} \leq 2 \cdot (1 - \beta_{i2j}) \ \ \forall i, j \in n_R \tag{41}$$

$$\eta^3_{ij} + \eta^4_{ij} \leq 2 \cdot (1 - \beta_{i3j}) \ \ \forall i, j \in n_R \tag{42}$$

$$\eta^1_{ij} + \eta^4_{ij} \leq 2 \cdot (1 - \beta_{i4j}) \ \ \forall i, j \in n_R \tag{43}$$

$$x_j \leq x_i + \eta^1_{ij} \cdot L \ \ \forall i, j \in n_R \tag{44}$$

$$y_j \leq y_i + \eta^2_{ij} \cdot W \ \ \forall i, j \in n_R \tag{45}$$

$$x'_i \leq x'_j + \eta^3_{ij} \cdot L \ \ \forall i, j \in n_R \tag{46}$$

$$y'_i \leq y'_j + \eta^4_{ij} \cdot W \ \ \forall i, j \in n_R \tag{47}$$

By constraint (26), all boxes that are not placed on the bottom level must be supported at all four bottom corners. (27) ensures that g_i is set to zero as long as box i is not placed on the pallet floor. Constraints (28) and (29) are added to establish v_{ij}, which is a number that is at least as large as $|z_i - z'_j|$. This is used to define SH_{ij} i.e if $v_{ij} = 0$ then $\text{SH}_{ij} = 0$ (30), and if $v_{ij} > 0$ then $\text{SH}_{ij} = 1$ (31). Constraints (32) and (33) establish p_{ij} in an upper triangular matrix ($j > i$) which shows if the projections on the xy-plane of box i and j overlap, and (34) fills out the lower triangle of the symmetrical matrix, since all entries of it are needed in the constraints (35) and (36). (35) forces s_{ij} to be 1 if box j is on a sufficient level to support box i and there is an overlap in their projections on the xy-plane. (36) forces s_{ij} to equal 0 if not both these criteria are satisfied and constraint (37) ensures that β_{ilj} is 0 for all $l \in 1..4$ as long as s_{ij} is zero. (38) and (39) ensure that box i cannot be supported by j if they are not both placed on the pallet. The remaining constraints determine exactly which boxes support which corners of other boxes.

3 Computational Experiments

The purpose of the experiments is to see what happens with computation time when some parameters of the input is changed. The different experiments which this paper tests the model on are summarized In table 1:

Table 1 Experiments.

Experiment	What is tested	Total Number of Boxes
1	Ratio between pallet volume and box volume	10 and 20
2	Ratio between small and large boxes	16
3	Number of boxes to be stacked	10, 20, 30, and 40

The experiments are selected to represent various scenarios that will be encountered when solving the problem in real life. All experiments are performed on a computer with an Intel(R) Quad Core i7- 4700MQ processor, 2.4 GHZ CPU, 8 GB RAM, a 64 bit OS and solved in IBM's ILOG CPLEX v. 12.6.2. All datasets include two box types and are constructed using the same rules. The dimensions of boxes and the pallet are created in such a way that the pallet dimension is always divisible by the corresponding box dimension e.g. the pallet length divided by the box length is always integer. The argument for this simplification is that it would be illogical to create boxes with dimensions that does not allow for total utilization of the pallet. Do note that this is not a requirement for the model but it simplifies creating datasets with an on forehand known solution. An example of such a dataset is shown below:

Table 2 Example of Dataset.

Pallet Dimension	Box Type 1 Dimension	# Box Type 1	Box Type 2 Dimension	# Box Type 2
120 × 80 × 60	20 × 20 × 10	12	60 × 40 × 20	8

In order to get consistent results for each experiment, ten datasets are created for each problem with varying pallet and box sizes. This serves to eliminate model bias towards performing better on smaller/larger pallets or with certain types of boxes.

3.1 Experiment 1

In the first experiment it is tested what impact, if any, the ratio between the maximum volume allowed on the pallet and the total volume of the boxes to be stacked has on computation time. This means that the following ratio is changed:

$$\frac{\sum\limits_{i \in n_R} l_i \cdot w_i \cdot h_i}{L \cdot W \cdot H}. \tag{48}$$

The ratios tested are: 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, and 2, that is, the total volume of boxes to be stacked equals 25% of the allowed pallet volume, 50 %, 75 % etc. Figure 1 shows computation times for datasets with 10 boxes, and Figure 3 of those with 20 boxes. Figure 2 and 4 show their respective means.

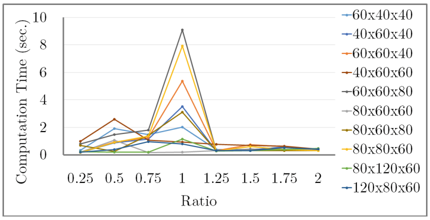


Fig. 1 Computation Times for 10-Box Problems.

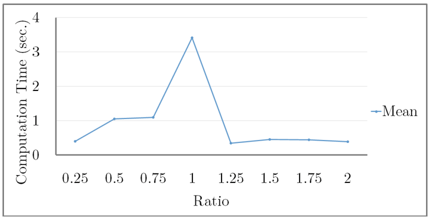


Fig. 2 Mean Computation Times for 10-Box Problems.

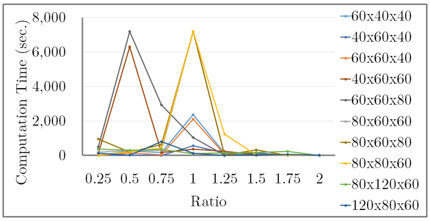


Fig. 3 Computation Times for 20-Box Problems.

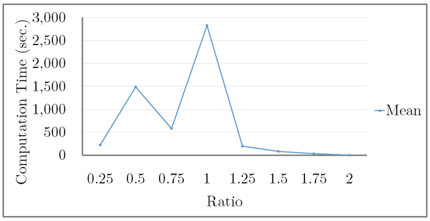


Fig. 4 Mean Computation Times for 20-Box Problems.

Note that the model is stopped at 7, 200 seconds. For the 10-box problems all instances were solved to optimality within ten seconds, but with 20 boxes four instances were not solved to optimality within two hours. In both experiments the most extensive computation times are found with a ratio of one, and lowest when the volume of the boxes exceeds the capacity of the pallet.
The overall conclusion is that on average the computation time is largest when the box-to-pallet ratio equals 1, smallest when it is greater than one, and when the ratio is smaller than one the stability constraints make it hard to find an optimal solution.

3.2 Experiment 2

In this experiment the ratio between the number of small and large boxes is tested. The box-to-pallet-volume ratio is restricted to be between 0.7 and 0.8 for the datasets used in this experiment. A problem with 12 large and 4 small boxes is constructed, another with 4 large and 12 small boxes, and lastly a problem with 8 large and 8 small boxes. Each problem is tested on 10 different datasets thus keeping the total number of boxes constant across experiments. The computation times for the three types of problems are plotted below.

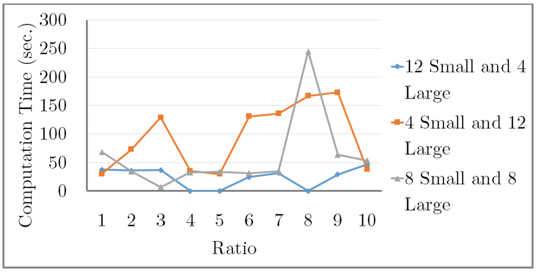


Fig. 5 Computation times for Experiment 2.

On average the computation time is longest with 12 large boxes. 8 out of 10 times this problem has a computation time larger than the one with 4 large boxes. Furthermore the fastest computation time is achieved in the problem with 4 large boxes and 12 small. The average computation time for each problem type is shown in the table below.

Table 3 Average Computation Time.

Problem	12 Small & 4 Large	4 Small & 12 Large	8 Small & 8 Large
Average Computation Time (sec.)	24.24	94.25	60.55

From Table 3 it can be seen that there is a significant difference in computation times and that the ratio between small and large boxes has an impact on the computation time: the more large boxes, the longer computation time.

3.3 Experiment 3

In this experiment it is tested how the number of boxes impacts the computation time. The dimensions of the two box types are fixed to 40 × 40 × 10 and 20 × 10 × 10 and the box-to-pallet-volume ratio lies between 0.75 and 0.9. A problem including 10 boxes to be packed is first solved and then the number increases by 10 until the problem is unsolvable within a reasonable time. Figure 6 shows the computation times.

As expected it can be concluded that the computation times increasingly grows when the number of boxes increases. The problem with 40 boxes is not solved within four hours. This indicates that this model is not useful for problems larger than 20-30 boxes when an optimal solution within a reasonable time is wanted.

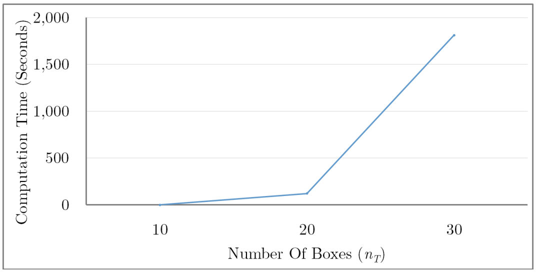


Fig. 6 Computation Times

4 Conclusions and Further Research

By experimental investigation of the DPLP it is established that having a box-to-pallet volume ratio equal to one has a large negative impact on the computation time compared to other ratios. This is interesting as for many practical cases of the single pallet DPLP, this ratio would typically lie close to 1. In cases where this ratio is far from 1 the problem described in this paper is less relevant. As such the results in this paper can be used to determine when the use of optimization is relevant and when meta-heuristics should be applied. If the volume of the pallet far exceeds the total volume of boxes to be packed it is not hard to fit all boxes on the pallet and when total box volume far exceeds pallet volume one may utilize more pallets, changing the problem formulation and nature. Therefore future research should involve changing the objective such as to minimize the packing height which is relevant in the first case. As for the second case it does not take much modification of the model to make it capable of solving multi-pallet problems. Furthermore, factors such as weight distribution, the horizontal stability of the pallet, and permitting the rotation of boxes will result in different search and solution spaces. Implementing these factors in the model might yield other results in the experiments constructed by this paper.

References

1. Paquay, C., Schyns, M., Limbourg, S.: A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. International Transactions in Operational Research (2014)
2. Chen, C., Lee, S.M., Shen, Q.: An analytical model for the container loading problem. European Journal of Operational Research **80**, 68–76 (1995)
3. Junqueira, L., Morabito, R., Yamashita, D.S., Yanasse, H.H.: Optimization models for the three-dimensional container loading problem with practical constraints. In: Modeling and Optimization in Space Engineering, pp. 271–293. Springer (2013)
4. Al-Shayea, A.M.: Solving the three-dimensional palet-paking problem using mixed 0–1 model. Journal of Service Science and Management **4**, 513 (2011)
5. Bischoff, E.E., Janetz, F., Ratcliff, M.: Loading pallets with non-identical items. European journal of operational research **84**, 681–692 (1995)
6. Wu, Y., Li, W., Goh, M., de Souza, R.: Three-dimensional bin packing problem with variable bin height. European Journal of Operational Research **202**, 347–355 (2010)
7. Tina, S., Sren, F., Jeppe Mulbjerg, G., Peter, N., Mukund, N.J., Madsen, O.: 3d pallet stacking with rigorous vertical stability. Submitted to the 13th International Conference on Distributed Computing and Artificial Intelligence 2016 to be held at Sevilla, Spain

GA-Based Scheduling for Transporting and Manufacturing Mobile Robots in FMS

Lam Nguyen, Quang-Vinh Dang and Izabela Nielsen

Abstract With a high degree of flexibility and automation, flexible manufacturing system (FMS) consisting of a number of automatic machine tools and mobile robots is capable of coping with customized product requirements. In addition to transporting, mobile robots participate in processing some value-added operations on some specific machines such as pre-assembly or quality inspection if required. This paper presents a genetic algorithm to deal with the problem of sequencing of operations, routing of mobile robots, and operations assignment for mobile robots in FMS with the aim to minimize the makespan. The performance of the algorithm is demonstrated by a generated numerical example.

Keywords Genetic algorithms · Mobile robots · FMS

1 Introduction

The current manufacturing systems have a tendency to become more and more flexible to adapt to the needs of product diversification. These systems are constructed of a number of automatic machines, material handling devices, e.g. automatic guided vehicles (AGVs) or mobile robots, and a central control computer. Likewise AGVs, mobile robots have the capability of moving around within their

L. Nguyen · Q.-V. Dang(✉)
Department of Global Production Engineering and Management,
Vietnamese-German University, Binh Duong, Vietnam
e-mail: {lam.nq,vinh.dq}@vgu.edu.vn

I. Nielsen
Department of Mechanical and Manufacturing Engineering,
Aalborg University, Aalborg, Denmark
e-mail: izabela@m-tech.aau.dk

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_59

working space to transport a variety of part types from one point to another and perform manipulation abilities without human intervention. In particular, these robots not only can transport part/component among machines but also execute various value-added activities on different machines (or workstations) thanks to their manipulation arm such as machine tending, pre-assembly, or quality inspection. Fig. 1 illustrates a typical layout of an FMS with five machines and two mobile robots and how the system operates. There are some operations transported by mobile robots and a few of them are processed by mobile robots on some specific machines. For instance, operation O_{i*j*} is transported from machine 4 to machine 5 by mobile robot R1 which then processes this operation on machine 5. After operation O_{i*j*} is finished, mobile robot R1 is only in charge of moving this task to machine 3 where operation $O_{i*,j*+1}$ will be executed. Subsequently, R1 delivers operation $O_{i,j+1}$ from machine 2 to machine 1 and then performs the processing of this operation on machine 1.

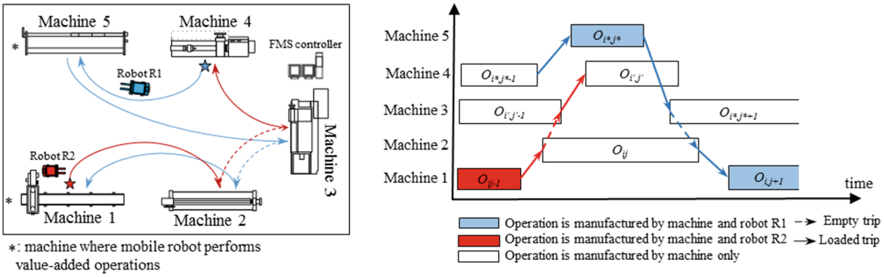


Fig. 1 Illustration of system operation

These operations require simultaneous accesses to resources, i.e. machines and mobile robots which can be considered multimodal processes [3,4] because the makespan, i.e. time to complete all tasks entirely depends on sequence of operations (*SO*), routes of mobile robots movement (*RR*), and mobile robot assignments (*RA*). In general the multimodal processes are denoted by $mP_k = (SO_k, RR_k, RA_k)$,

where: $SO_k = \{mo_{i1},..., mo_{ij}\}$ - mo_{ij} : operation *i* of task *j*,

$RR_k = \{0, 1, 0, 2,\ldots, mR_n\}$ - $mR_n \in R \mid R$: number of mobile robots ,

$RA_k = \{0, 0, 1, 2,\ldots, mR_n\}$- $mR_n \in R \mid R$: number of mobile robots .

For instance, considering a problem with 6 tasks (16 operations) and 2 mobile robots there exists many possible multimodal processes. Two of them, mP_1 and mP_2, are presented to illustrate operation sequences and mobile robot assignments for transporting and manufacturing operations. The makespan of mP_1 and mP_2 are respectively 215 and 232 (time units) as seen in the Gantt chart of Fig. 2.

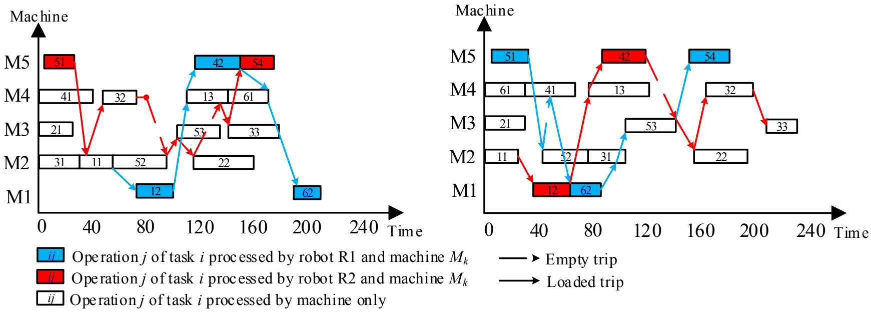


Fig. 2 Gantt chart for mP_1 and mP_2

$$mP_1 = (SO_1, RR_1, RA_1)$$
$$mP_2 = (SO_2, RR_2, RA_2)$$

with: $SO_1 = \{51, 21, 41, 31, 32, 11, 52, 12, 13, 53, 42, 22, 61, 33, 54, 62\}$
$RR_1 = \{0, 0, 0, 0, 2, 0, 2, 1, 1, 2, 1, 2, 0, 2, 2, 1\}$
$RA_1 = \{0, 0, 0, 1, 0, 2, 0, 2, 2, 0, 1, 1, 1, 2, 2, 2\}$
$SO_2 = \{51, 11, 21, 52, 61, 12, 41, 13, 42, 31, 62, 53, 54, 22, 32, 33\}$
$RR_2 = \{0, 0, 0, 0, 2, 0, 2, 1, 1, 2, 1, 2, 0, 2, 2, 1\}$
$RA_2 = \{0, 0, 0, 1, 0, 2, 0, 2, 2, 0, 1, 1, 1, 2, 2, 2\}$

In order to utilize the described network in an efficient manner, it is important to properly schedule operations on machines and concurrently schedule mobile robots for transporting these operations and manufacturing some of them if required. In other words, the problem consists of decisions on the sequences of operations, routes of mobile robots, and operation assignment for mobile robots. This study is executed under some assumptions: every first operation is available at a machine; each mobile robot can transport only one kind of parts at a time; the input and output buffer spaces are sufficient; traveling time and processing time are deterministic; loading and unloading time are included in the traveling time of loaded trips; such issues as traffic congestions, mobile robot collisions, machine failures or scraps are not considered in this paper.

This problem has been modeled in several respects comparable to the scheduling problems of operations and AGVs in the last three decades. In particular, Blazewicz et al. [2] propose a dynamic programming approach to schedule both machines and vehicles. Many solution algorithms have been then presented to solve the problem of operation sequencing and AGV assignment such as evolutionary algorithm by Ulusoy et al. [13], hybrid multi-objective genetic algorithm (GA) by Reddy and Rao [11], and adaptive GA by Jerald et al. [8]. Furthermore, the problem of simultaneous scheduling of operations and mobile robots has been considered by Nielsen et al. [10]. These mobile robots are in charge of carrying out a part/component from a machine to another, and then come back to its own workstation to perform a preemptive task. Dang and Nguyen [5] propose a genetic algorithm-based heuristic to deal with the preemptive scheduling of mobile robots with multiple transportation being taken into account. The main novelty of this paper lies in the fact that mobile robots are capable of transporting operations to destination machines and processing some of them there if required. In that context, our contribution is to develop a solution approach mimicking the genetic algorithm in order to find the best solutions for the considered problem, which is presented in the next section.

2 Solution Algorithm

The considered problem of sequencing of operations, routing of mobile robots, and operation assignment for mobile robots belongs to NP-hard class [6]. Hence in this paper a genetic algorithm is developed to convert the problem to the way that the best solutions could be found.

Encoding and Decoding Scheme

A feasible solution is considered as a chromosome representing a sequence of operation and mobile robot assignments. A gene of the chromosome is constructed by three parts: the first part is operation, the second one is mobile robot carrying out that operation, and the third one indicates mobile robot performing value-added activities on that operation. In case a gene includes the first operation of every task, the second part of a gene is zero (0). In addition, as an operation do not need the participation of mobile robot to perform a manufacturing activity, the third part of a gene is zero (0). A feasible chromosome for an exemplary scheduling problem is illustrated in Fig. 3.

Chromosome	11,0,0	21,0,0	12,2,1	31,0,0	41,0,1	32,2,0	42,1,1	13,1,0	22,1,0

Fig. 3 Illustration of a feasible chromosome

Chromosomes in the initial population are randomly generated. Each chromosome is built up gene by gene. The first part of a gene is assigned an eligible operation whose predecessors are assigned. If that operation needs to be done by mobile robot, one of the mobile robots from set *R* is selected randomly and then assigned to the third part of gene; otherwise, the value of this part must be zero (0). Moreover if that eligible operation is the first operation, zero (0) is assigned to the second part. Otherwise, one of the mobile robots is randomly chosen. The eligible set of operations is updated and the process continues as shown in Fig. 4.

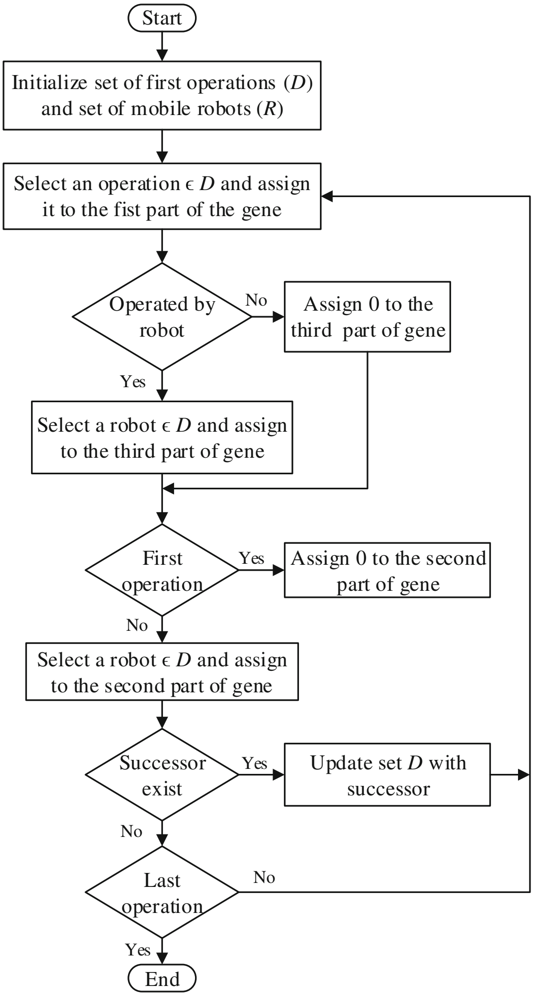

Start
Initialize set of first operations (D) and set of mobile robots (R)
Select an operation ϵ D and assign it to the fist part of the gene
Operated by robot
No
Assign 0 to the third part of gene
Yes
Select a robot ϵ D and assign to the third part of gene
First operation
Yes
Assign 0 to the second part of gene
No
Select a robot ϵ D and assign to the second part of gene
Successor exist
Yes
Update set D with successor
No
Last operation
No
Yes
End

Fig. 4 Procedure of the initialization

The pseudocode of decoding method is presented in Fig. 5 below:

```
Input: a feasible solution $k$, number of mobile robot $nr$
Output: fitness value $V_k$
  Begin
    // $tt(a,b)$: travelling time from point $a$ to point $b$
    $V_k \leftarrow 0$;
    For $i$= 1 to $l[k]$ // $l[k]$: length of solution $k$
      $tso$ [i] ← 0; // $tso[i]$: starting time of gene $i$
      $tco$ [$i$] ← 0; // $tco[i]$ : completion time of gene $i$
    End
    For $i$= 1 to $nr$
      $ld[i]$ ← 0; // $ld[i]$: last destination of robot $i$
      $lf$ [$i$] ← 0; // $lf[i]$: time finish last trip of robot $i$
    End
    For $i$= 1 to $l[k]$
      If $O_k[i]$ does not need to be transported and processed by robot then
                                            // $O_k[i]$: operation of gene $i^{th}$ of solution $k$
        $pm$ ← completion time of last operation on machine $mo$;
        $tso[i] \leftarrow pm$;
        $tco[i] \leftarrow tso[i] + tp[i]$;          // $tp[i]$: processing time of $O_k[i]$

       Elseif $O_k[i]$ processed by robot $r$ and available on machine then
                                              // $r$: index of robot
        $tso[i] \leftarrow lf[r] + tt(ld[r], mo)$;     // $mo$: machine processes $O_k[i]$
        $tco[i] \leftarrow tso[i] + tp[i]$;
        $ld[r] \leftarrow mo$;
        $lf[r] \leftarrow tco[i]$;

      Elseif $O_k[i]$ is transported by robot $r$ only then
        $pd$ ← completion time of predecessor;
        $mpd$ ← machine processes predecessor of $O_k[i]$;
        $pm$ ← completion time of last operation on machine $mo$;
        $tso[i] \leftarrow \mathbf{max}\{lf[r] + tt(ld[r], mpd) + tt(mpd, mo), pd + tt(mpd, mo), pm\}$;
        $tco[i] \leftarrow tso[i] + tp[i]$;
        $ld[r] \leftarrow mo$;
        $lf[r] \leftarrow tso[i]$;
      Elseif $O_k[i]$ is transported by robot $r$ and processed by robot $r'$ then
                                              // $r'$: index of robot
        $pd$ ← completion time of predecessor;
        $mpd$ ← machine processes predecessor of $O_k[i]$;
        $pm$ ← completion time of last operation on machine $mo$;
        $tso[i] \leftarrow \mathbf{max}\{lf[r] + tt(ld[r], mpd) + tt(mpd, mo), lf[r']+tt(ld[r'], mo), pd$
        $+tt(mpd, mo), pm\}$;
        $tco[i] \leftarrow tso[i] + tp[i]$;
        $ld[r] \leftarrow mo$;
        $ld[r'] \leftarrow mo$;
        $lf[r] \leftarrow tso[i]$;
        $lf[r'] \leftarrow tco[i]$;
    End
    Output $V_k$;
  End
```

Fig. 5 Pseudocode of fitness evaluation

Genetic Operators

Parent chromosomes are probabilistically selected by the Roulette-wheel method [9] based on the fitness value, and uniform crossover [1] operating with probability P_c is used to generate two offspring. With mixing ratio equal to 0.5, each offspring receives approximately 50 per cent genes from each parent. In other words, two parent chromosomes exchange their genes with probability of 0.5.

In order to explore the searching spaces, two mutation operators with probability P_m are employed to produce random changes in gene level. The first mutation selects two random positions on a chromosome and swaps these with respect to those positions. The second mutation operator generates two offspring by making random changes in mobile robots assigned for transporting and those assigned for manufacturing.

Chromosomes produced by the first mutation operator may be infeasible in terms of precedence constraints. Therefore, a repair operator is proposed to validate these chromosomes by exchanging location of operations belong to the same task so that a valid sequence is achieved. In the successive generations, ($\mu + \lambda$) selection [7] is employed to choose chromosomes for reproduction while remaining population diversity by fitness sharing method and keeping the best solution by elitist method [12], which guarantees a wide range of good chromosome can remain for the next generation.

3 Numerical Example

In this section, a generated problem is used to examine the performance of the proposed algorithm. 16 operations of 6 tasks are carried out on 5 machines. Two mobile robots are considered. A layout of this problem can be seen in Fig. 1. Task description, processing time, and traveling time of mobile robots can be found in Table 1 and Table 2.

Table 1 Task description

Task	Operation	Processing time (time unit)	Machine	Mobile robot
1	1	28	2	-
	2	30	1	R1, R2
	3	32	4	-
2	1	32	3	-
	2	42	2	-
3	1	38	2	-
	2	30	4	
	3	25	3	-
4	1	45	4	-
	2	35	5	R1, R2
5	1	20	5	R1, R2
	2	28	2	-
	3	30	3	-
	4	22	5	R1, R2
6	1	32	4	-
	2	25	1	R1, R2

Table 2 Travelling time of mobile robots (time unit)

From/To	M1	M2	M3	M4	M5
M1	0	8	8	12	13
M2	10	0	14	8	10
M3	8	12	0	10	12
M4	12	10	12	0	16
M5	10	10	14	10	0

For GA parameters, N_p (population size), P_c, and P_m are set to be 100, 0.4, and 0.1, respectively. The proposed algorithm terminates at the generation of 1000. The algorithm has been programmed in VB.NET and run on a PC having an Intel® Core i5, 2.2 GHz processor, and 4GB RAM. The best solution (multimodal process mP_{best}) is found as follows:

SO_{best}= {61, 11, 51, 21, 12, 31, 52, 41, 32, 42, 62, 22, 53, 13, 33, 54}
RR_{best}= {0, 0, 0, 0, 1, 0, 2, 0, 2, 2, 2, 1, 1, 1, 1, 1}
RA_{best} = {0, 0, 2, 0, 1, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1}
with the computation time of approximately 7 seconds.

Fig. 6 depicts a Gantt chart for the best solution whose makespan is 182 (time unit). The first operation of each task is available on machines, thus it does not need to be transported while the others have to be transported by mobile robots. For instance, mobile robot R2 performs operation 51 on machine 5 and subsequently delivers to machine 2 prior to transporting operation 32 from machine 2 to machine 4. Similarly, mobile robot R1 picks up operation 12 from machine 2 and delivers it to machine 1 where this operation is processed by robot R1. After operation 12 is completed, robot R1 is only in charge of moving it to machine 4 where operation 13 will be executed. Robot R1 continues performing transportation of some operations prior to delivering operation 54 to machine 5 and executing it there.

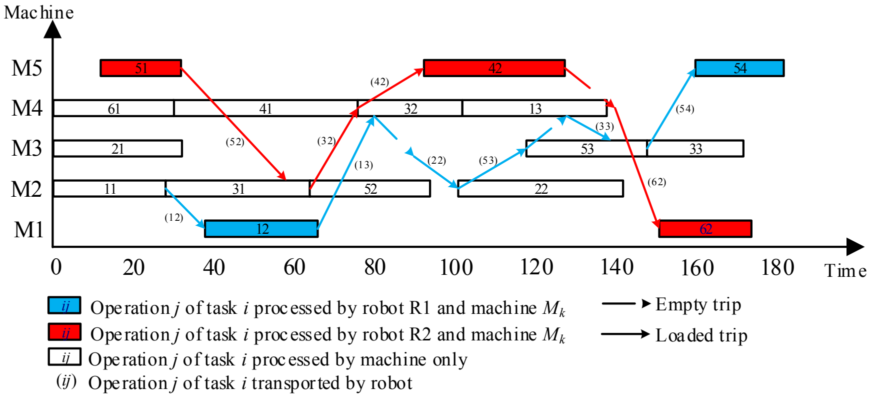


Fig. 6 Gantt chart for the best solution (multimodal process)

4 Conclusions

The joint problem of sequencing of operations, routing of mobile robots, and operation assignment for mobile robots is studied. The main novelty of this paper lies in the mobile robots' capabilities of transporting operations among machines and performing some valued-added activities on specific machines when required. With the aim of minimizing the time to complete all tasks, a genetic algorithm is proposed to find the best solutions for the described problem. A generated numerical example demonstrates the effectiveness of the proposed algorithm whose solutions are useful in the decision-making process. For further research, other competing approaches and comparative analysis of the achieved results are implemented in order to evaluate the performance of the proposed algorithm.

References

1. Abdelmaguid, T.F., Nassef, O.N., Kamal, B.A., Hassan, M.F.: A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. Int. J. Prod. Res. **42**, 267–281 (2004)
2. Blazewicz, J., Eiselt, H.A., Finke, G., Laporte, G., Weglartz, J.: Scheduling tasks and vehicles in a flexible manufacturing system. Int. J. Flex. Manuf. Sys. **4**, 5–16 (1991)
3. Bocewicz, G., Muszyński, W., Banaszak, Z.: Models of multimodal networks and transport processes. Bulletin of the Polish Academy of Sciences Technical Sciences **63**, 635–650 (2015)
4. Bocewicz, G., Nielsen, P., Banaszak, Z.A., Dang, Q.V.: Cyclic steady state refinement: multimodal processes perspective. In: Frick, J., Laugen, B.T. (eds.) APMS 2011, IFIP AICT, vol. 384, pp. 18–26. Springer, Heidelberg (2012)
5. Dang, Q.V., Nguyen, L.: A heuristic approach to schedule mobile robots in flexible manufacturing environments. Procedia CIRP **40**, 390–395 (2016)
6. Ganesharajah, T., Hall, N.G., Sriskandarajah, C.: Design and operational issues in AGV-served manufacturing systems. Ann. Oper. Res. **76**, 109–154 (1998)
7. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Kluwer Academic Publishers, Boston (1989)
8. Jerald, J., Asokan, P., Saravanan, R., Rani, A.D.C.: Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. Int. J. Adv. Manuf. Technol. **29**, 584–589 (2006)
9. Lin, L., Shinn, S.W., Gen, M., Hwang, H.: Network model and effective evolutionary approach for AGV dispatching in manufacturing system. J. Intell. Manuf. **17**, 465–477 (2006)
10. Nielsen, I., Dang, Q.V., Nielsen, P., Pawlewski, P.: Scheduling of mobile robots with preemptive tasks. In: Omatu et al., S. (eds). Distributed Computing and Artificial Intelligence, 11th International Conference, AISC, vol. 290, pp. 19–27. Springer, Switzerland (2014)
11. Reddy, B.S.P., Rao, C.S.P.: A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. Int. J. Adv. Manuf. Technol. **31**, 602–613 (2006)
12. Sareni, B., Krähenbühl, L.: Fitness sharing and niching methods revisited. IEEE Trans. Evol. Comput. **2**, 97–106 (1998)
13. Ulusoy, G., Sivrikaya-Şerifoğlu, F., Bilge, Ü: A genetic algorithm approach to the simultaneous scheduling of stations and automated guided vehicles. Comput. Oper. Res. **24**, 335–351 (1997)

Erratum to: Distributed Computing and Artificial Intelligence, 13th International Conference

Sigeru Omatu, Ali Selamat, Grzegorz Bocewicz, Pawel Sitek, Izabela Nielsen, Julián A. García-García and Javier Bajo

Erratum to:
S. Omatu et al. (eds.),
Distributed Computing and Artificial Intelligence,
13th International Conference,
Advances in Intelligent Systems and Computing,
DOI: 10.1007/978-3-319-40162-1

In the original version, the volume editor's name was misspelled in the copyright page. The correct name is Ali Selamat.

The online version of the updated original book can be found under
DOI: 10.1007/978-3-319-40162-1

S. Omatu et al. (eds.), *DCAI, 13th International Conference*,
Advances in Intelligent Systems and Computing 474,
DOI: 10.1007/978-3-319-40162-1_60

Author Index