Studies in Computational Intelligence 465

Kurosh Madani
António Dourado
Agostinho Rosa
Joaquim Filipe (Eds.)

# Computational Intelligence

Revised and Selected Papers of the International Joint Conference, IJCCI 2011, Paris, France, October 24–26, 2011

Springer

# Studies in Computational Intelligence 465

**Editor-in-Chief**

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/7092

Kurosh Madani, António Dourado,
Agostinho Rosa, and Joaquim Filipe (Eds.)

# Computational Intelligence

Revised and Selected Papers of the
International Joint Conference, IJCCI 2011,
Paris, France, October 24–26, 2011

Springer

*Editors*
Prof. Kurosh Madani
Images, Signals and Intelligence Systems Laboratory
University PARIS-EST Creteil (UPEC)
Paris 12
France

Prof. António Dourado
Departamento de Engenharia Informatica
Polo II - Pinhal de Marrocos
University of Coimbra
Coimbra
Portugal

Prof. Agostinho Rosa
Instituto Superior Tecnico IST
Systems and Robotics Institute
Evolutionary Systems and Biomedical Engineering Lab
Lisboa
Portugal

Prof. Joaquim Filipe
Polytechnic Institute of Setúbal / INSTICC
Setubal
Portugal

ISSN 1860-949X
e-ISSN 1860-9503
ISBN 978-3-642-35637-7
e-ISBN 978-3-642-35638-4
DOI 10.1007/978-3-642-35638-4
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012954154



Printed on acid-free paper

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

# Preface

Theoretical, applicative and technological challenges, emanating from nowadays' industrial, socioeconomic or environment needs, open every day new dilemmas to solved and new challenges to defeat. Computational Intelligence (Neural Computation, Fuzzy Computation and Evolutionary Computation) and related topics have shown their astounding potential in overcoming the above-mentioned needs. It is a fact and at the same time a great pleasure to notice that the ever-increasing interest of both confirmed and young researchers on this relatively juvenile science, upholds a reach multidisciplinary synergy between a large variety of scientific communities making conceivable a forthcoming emergence of viable solutions to these real-world complex challenges.

Since its first edition in 2009, the purpose of International Joint Conference on Computational Intelligence (IJCCI) is to bring together researchers, engineers and practitioners in computational technologies, especially those related to the areas of fuzzy computation, evolutionary computation and neural computation. IJCCI is composed of three co-located conferences, each one specialized in one of the aforementioned - knowledge areas. Namely:

- International Conference on Evolutionary Computation Theory and Applications (ECTA)
- International Conference on Fuzzy Computation Theory and Applications (FCTA)
- International Conference on Neural Computation Theory and Applications (NCTA)

Their aim is to provide major forums for scientists, engineers and practitioners interested in the study, analysis, design and application of these techniques to all fields of human activity.

In ECTA modeling and implementation of bioinspired systems namely on the evolutionary premises, both theoretically and in a broad range of application fields, is the central scope. Considered a subfield of computational intelligence focused on combinatorial optimization problems, evolutionary computation is associated with systems that use computational models of evolutionary processes as the key elements in design and implementation, i.e. computational techniques which are inspired by the evolution of biological life in the natural world. A number of evolutionary computational models have been proposed, including evolutionary algorithms, genetic algorithms, evolution strategies, evolutionary programming, swarm optimization and artificial life.

In FCTA, modeling and implementation of fuzzy systems, in a broad range of fields is the main concern. Fuzzy computation is a field that encompasses the theory and application of fuzzy sets and fuzzy logic to the solution of information processing, system analysis and decision problems. Bolstered by information technology developments, the extraordinary growth of fuzzy computation in recent years has led to major applications in fields ranging from medical diagnosis and automated learning to image understanding and systems control.

NCTA is focused on modeling and implementation of artificial neural networks computing architectures. Neural computation and artificial neural networks have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics, in problems of prediction, classification decision or control. Several architectures, learning strategies and algorithms have been introduced in this highly dynamic field in the last couple of decades.

The present book includes extended and revised versions of a set of selected papers from the Third International Joint Conference on Computational Intelligence (IJCCI 2011), held in Paris, France, from 24 to 26 October, 2011.

IJCCI 2011 received 283 paper submissions from 59 countries in all continents. To evaluate each submission, a double blind paper review was performed by the Program Committee. After a stringent selection process, 35 papers were accepted to be published and presented as full papers, i.e. completed work, 61 papers reflecting work-in-progress or position papers were accepted for short presentation, and another 57 contributions were accepted for poster presentation. These numbers, leading to a "full-paper" acceptance of about 12% and a total oral paper presentations acceptance ratio close to 34%, show the high quality forum for the present and next editions of this conference. This book includes revised and extended versions of a strict selection of the best papers presented at the conference.

Furthermore, IJCCI 2011 included six plenary keynote lectures given by Qiangfu Zhao, Witold Pedrycz, Didier Dubois, Marco A. Montes de Oca, Plamen Angelov and Michel Verleysen. We would like to express our appreciation to all of them and in particular to those who took the time to contribute with a paper to this book.

On behalf of the Conference Organizing Committee, we would like to thank all participants. First of all to the authors, whose quality work is the essence of the conference, and to the members of the Program Committee, who helped us with their expertise and diligence in reviewing the papers. As we all know, producing a post-conference book, within the high technical level exigency, requires the effort of many individuals. We wish to thank also all the members of our Organizing Committee, whose work and commitment were invaluable.

September 2012

Kurosh Madani
António Dourado
Agostinho Rosa
Joaquim Filipe

# Organization

## Conference Co-chairs

| | |
|---|---|
| Joaquim Filipe | Polytechnic Institute of Setúbal / INSTICC, Portugal |
| Janusz Kacprzyk | Systems Research Institute - Polish Academy of Sciences, Poland |

## Program Co-chairs

## ECTA

| | |
|---|---|
| Agostinho Rosa | IST, Technical University of Lisbon, Portugal |

## FCTA

| | |
|---|---|
| António Dourado | University of Coimbra, Portugal |

## NCTA

| | |
|---|---|
| Kurosh Madani | University of Paris-EST Créteil (UPEC), France |

## Organizing Committee

Sérgio Brissos, INSTICC, Portugal
Helder Coelhas, INSTICC, Portugal
Vera Coelho, INSTICC, Portugal
Andreia Costa, INSTICC, Portugal
Patrícia Duarte, INSTICC, Portugal
Bruno Encarnação, INSTICC, Portugal
Liliana Medina, INSTICC, Portugal
Carla Mota, INSTICC, Portugal
Raquel Pedrosa, INSTICC, Portugal
Vitor Pedrosa, INSTICC, Portugal
Daniel Pereira, INSTICC, Portugal
Cláudia Pinto, INSTICC, Portugal
José Varela, INSTICC, Portugal
Pedro Varela, INSTICC, Portugal

## ECTA Program Committee

Chang Wook Ahn, Korea, Republic of
Christos Ampatzis, Belgium
Thomas Baeck, The Netherlands
Pedro Ballester, UK
Michal Bidlo, Czech Republic
Tim Blackwell, UK
Maria J. Blesa, Spain
Christian Blum, Spain
Stefan Boettcher, USA
Indranil Bose, Hong Kong
J. Arturo Perez C., Mexico
David Cairns, UK
Chi Kin Chow, Hong Kong
Antonio Della Cioppa, Italy
Narely Cruz-Cortes, Mexico
Liliana Dobrica, Romania
Peter Duerr, Switzerland
Bruce Edmonds, UK
Fabio Fassetti, Italy
Carlos Fernandes, Spain
Stefka Fidanova, Bulgaria
Dalila Fontes, Portugal
Girolamo Fornarelli, Italy
Xavier Gandibleux, France
Ozlem Garibay, USA
Carlos Gershenson, Mexico
Rosario Girardi, Brazil
Garrison Greenwood, USA
Jörg Hähner, Germany
J. Ignacio Hidalgo, Spain
Jinglu Hu, Japan
William N.N. Hung, USA
Seiya Imoto, Japan
Christian Jacob, Canada
Colin Johnson, UK
Marta Kasprzak, Poland
Ed Keedwell, UK
Abdullah Konak, USA
Mario Köppen, Japan
Ondrej Krejcar, Czech Republic
Jiri Kubalik, Czech Republic
Antonio J. Fernández Leiva, Spain
Wenjian Luo, China
Penousal Machado, Portugal
Bob McKay, Korea, Republic of
Barry McMullin, Ireland
Jörn Mehnen, UK
Ambra Molesini, Italy
Sanaz Mostaghim, Germany
Luiza de Macedo Mourelle, Brazil
Schütze Oliver, Mexico
Pietro S. Oliveto, UK
Ender Özcan, UK
Grammatoula Papaioannou, UK
Gary Parker, USA
Petrica Pop, Romania
Aurora Pozo, Brazil
Joaquim Reis, Portugal
Mateen Rizki, USA
Katya Rodriguez, Mexico
Guenter Rudolph, Germany
Miguel A. Sanz-Bobi, Spain
Lukáš Sekanina, Czech Republic
Franciszek Seredynski, Poland
Alice Smith, USA
Giandomenico Spezzano, Italy
Giovanni Stracquadanio, USA
Emilia Tantar, Luxembourg
Jonathan Thompson, UK
Vito Trianni, Italy
Krzysztof Trojanowski, Poland
Athanasios Tsakonas, UK
Elio Tuci, UK
Massimiliano Vasile, UK
Neal Wagner, USA
Jingxuan Wei, New Zealand
Bart Wyns, Belgium
Xin-She Yang, UK
Gary Yen, USA
Shiu Yin Yuen, China

## ECTA Auxiliary Reviewers

Alexandru-Adrian Tantar, Luxembourg
Argyrios Zolotas, UK

## FCTA Program Committee

Shigeo Abe, Japan
Sansanee Auephanwiriyakul, Thailand
Ulrich Bodenhofer, Austria
Jinhai Cai, Australia
Daniel Antonio Callegari, Brazil
Heloisa Camargo, Brazil
Giovanna Castellano, Italy
Gregory Chavez, USA
France Cheong, Australia
Bijan Davvaz, Iran, Islamic Republic of
Kudret Demirli, Canada
Ioan Despi, Australia
Scott Dick, Canada
József Dombi, Hungary
Girolamo Fornarelli, Italy
Yoshikazu Fukuyama, Japan
Jonathan Garibaldi, UK
Alexander Gegov, UK
Brunella Gerla, Italy
Chang-Wook Han, Korea, Republic of
Lars Hildebrand, Germany
Chih-Cheng Hung, USA
Uzay Kaymak, The Netherlands
László T. Kóczy, Hungary
Donald H. Kraft, USA
Anne Laurent, France
Kang Li, UK
Tsung-Chih Lin, Taiwan
Ahmad Lotfi, UK
Francesco Marcelloni, Italy
Ludmil Mikhailov, UK
Hiroshi Nakajima, Japan
Yusuke Nojima, Japan
Raúl Pérez, Spain
Sanja Petrovic, UK
David Picado, Spain
Valentina Plekhanova, UK
Antonello Rizzi, Italy
Alessandra Russo, UK
Luciano Sanchez, Spain
Steven Schockaert, Belgium
Umberto Straccia, Italy
Dat Tran, Australia
Christian Wagner, UK
Thomas Whalen, USA
Dongrui Wu, USA
Chung-Hsing Yeh, Australia
Jianqiang Yi, China
Hans-Jürgen Zimmermann, Germany

## NCTA Program Committee

Shigeo Abe, Japan
Veronique Amarger, France
Vijayan Asari, USA
Arash Bahrammirzaee, Iran, Islamic Republic of
Ammar Belatreche, UK
Gilles Bernard, France
Daniel Berrar, Japan
Yevgeniy Bodyanskiy, Ukraine
Samia Bouchafa, France
Ivo Bukovsky, Czech Republic
María José Castro-Bleda, Spain
João Catalão, Portugal
Abdennasser Chebira, France
Ning Chen, Portugal
Amine Chohra, France
Seungjin Choi, Korea, Republic of
Catalina Cocianu, Romania
Madjid Fathi, Germany
Girolamo Fornarelli, Italy
Josep Freixas, Spain
Marcos Gestal, Spain

Vladimir Golovko, Belarus
Michèle Gouiffès, France
Dongfeng Han, USA
Tom Heskes, The Netherlands
Robert Hiromoto, USA
Gareth Howells, UK
Magnus Johnsson, Sweden
Juha Karhunen, Finland
Christel Kemke, Canada
Dalia Kriksciuniene, Lithuania
Adam Krzyzak, Canada
Edmund Lai, New Zealand
H.K. Lam, UK
Noel Lopes, Portugal
Jinhu Lu, China
Jinwen Ma, China
Hichem Maaref, France
Kurosh Madani, France
Jean-Jacques Mariage, France
Mitsuharu Matsumoto, Japan
Ali Minai, USA
Adnan Abou Nabout, Germany
Mourad Oussalah, UK
Eliano Pessa, Italy
Dominik Maximilián Ramík, France
Neil Rowe, USA
Christophe Sabourin, France
Giovanni Saggio, Italy
Abdel-badeeh Mohamed Salem, Egypt
Gerald Schaefer, UK
Alon Schclar, Israel
Christoph Schommer, Luxembourg
Moustapha Séne, Senegal
Catherine Stringfellow, USA
Shiliang Sun, China
Johan Suykens, Belgium
Norikazu Takahashi, Japan
Oscar Mauricio Reyes Torres, Germany
Carlos M. Travieso, Spain
Andrei Utkin, Portugal
Michel Verleysen, Belgium
Brijesh Verma, Australia
Eva Volna, Czech Republic
Shuai Wan, China
Shandong Wu, USA
Yingjie Yang, UK
Weiwei Yu, China
Cleber Zanchettin, Brazil
Huiyu Zhou, UK

## NCTA Auxiliary Reviewers

Kye-Hyeon Kim, Korea, Republic of
Yunjun Nam, Korea, Republic of
Jiho Yoo, Korea, Republic of
Jeong-Min Yun, Korea, Republic of

## Invited Speakers

| | |
|---|---|
| Qiangfu Zhao | University of Aizu, Japan |
| Witold Pedrycz | University of Alberta, Canada / Polish Academy of Sciences, Poland |
| Didier Dubois | Institut de Recherche en Informatique de Toulouse, France |
| Marco A. Montes de Oca | University of Delaware, USA |
| Plamen Angelov | Lancaster University, UK |
| Michel Verleysen | Université Catholique de Louvain, Belgium |

# Contents

## Part II: Fuzzy Computation Theory and Applications



## Part III: Neural Computation Theory and Applications

# Invited Papers

# Computational Awareness: Another Way towards Intelligence

Qiangfu Zhao

The University of Aizu, Aizu-Wakamatsu, Japan
qf-zhao@u-aizu.ac.jp

**Abstract.** Artificial intelligence (AI) has been a dream of researchers for decades. In 1982, Japan launched the 5th generation computer project, expecting to create AI in computers, but failed. Noting that logic approach alone is not enough, soft computing (e.g. neuro-computing, fuzzy logic and evolutionary computation) has attracted great attention since 1990s. After another 2 decades, however, we have not got any system that is as intelligent as a human, in the sense of "over-all performance". Instead of trying to create intelligence directly, we may try to create "awareness" first, and obtain intelligence "step-by-step". Briefly speaking, awareness is a mechanism for detecting any event which may or may not lead to complete understanding. Depending on the complexity of the events to detect, aware systems can be divided into many levels. Although low level aware systems may not be clever enough to provide understandable knowledge about an observation; they may provide important information for high level aware systems to make understandable decisions. In this paper we do not intend to provide a survey of existing results related to awareness computing. Rather, we will study this field from a new perspective, try to clarify some related terminologies, and propose some problems to solve for creating intelligence through computational awareness.

## 1 Introduction

Artificial intelligence (AI) has been a dream of researchers for decades. In 1982, Japan launched the 5th generation computer project, expecting to create AI in computers [1], but failed. Noting that logic approach alone is not enough, soft computing (e.g. neuro-computing, fuzzy logic and evolutionary computation) has attracted great attention since 1990s. After another 2 decades, however, we have not got any system that is as intelligent as a human, in the sense of "over-all performance". Instead of trying to create intelligence directly, we may try to create "awareness" first. Briefly speaking, awareness is a mechanism for detecting an event. The detected event itself may not make sense, but it may provide important information for further understanding. Thus, creating different levels of awareness in a computer may lead to intelligence step-by-step.

The term "awareness computing" has been used for more than two decades in the context of computer supported cooperative work (CSCW) , ubiquitous computing, social network, and so on [2]-[14]. So far, awareness computing has been considered by researchers as a process for acquiring and distributing context information related

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 3–14.
DOI: 10.1007/978-3-642-35638-4_1 

to *what is happening*, *what happened*, and *what is going to happen* in an environment under concern. The main purpose of awareness computing is to provide context information in a timely manner so that human users or computing machines can take actions or make decisions proactively before something (risk, danger, chance, etc.) really happens.

So far, many aware systems have been studied in the literature. The systems are often classified based on the event to be aware of. Examples include, context aware, situation aware, intention aware, preference aware, location aware, energy aware, risk aware, chance aware, and so on. This classification is not helpful for us to understand the fundamental properties of aware systems because it divides aware systems into so many categories and the boundaries between the categories are not clear.

In this paper, we classify aware systems based on the following two factors:

1. Is the system aware of some event(s)?
2. Does the system make some decision(s) based on awareness?

Based on these two factors, existing aware systems can be divided into 3 types, namely NanD, AnD, and AmD. Detailed discussion is given as follows.

### 1.1 Type-I: Nand (No Aware, No Decision) Systems

This is the simplest case in which the aware system just provides the context, and the human user must be aware of any useful information contained in the context, and make decisions based on the information. The "media space" developed in 1986 [15] is a NanD system. It can provide all kinds of background information for cooperative work. Users in different places can work together as if they are in the same room. In fact, most monitoring systems for traffic control, for nuclear power plant, for public facilities, and so on, are NanD systems. These systems are useful for routine cooperative work. In emergent cases, however, human users may fail to detect possible dangers (because of the limited computing power of the human brain) even if the background information is provided seamlessly.

### 1.2 Type-II: And (Aware, but No Decision) Systems

An AnD system is aware of the importance or urgency of different context patterns, so that critical information can be provided to the human user (or some other systems) in a more noticeable, comprehensible and/or visible way. Clearly, compared with NanD systems, AnD systems are more aware, and may enhance the awareness ability of human users significantly. The system may help human users to detect important clues for solving a problem, for detecting some danger, for seizing a chance, etc. Many decision supporting systems (DSS) developed (and used successfully in many areas) so far are AnD systems, although in many cases their developers did not intend to build an "aware system" at all.

### 1.3 Type-III: Amd (Aware and Make Decision) Systems

An AmD system is aware of the meaning of the context patterns, and can make corresponding decisions for the user. For example, in an intelligent environment (IE)

(e.g. smart home, smart office, smart nursing room, etc.)[16], the server may be aware of the contexts related to actions, locations, behaviors, and so on, of the human users; and can provide suitable services based on different context patterns. Examples of services may include: switching-on/off of light to keep the best lighting condition in a smart office; providing software/hardware resources to meet the requirement of a user in cloud-computing; sending a certain amount of electricity to a client in a smart grid power supply environment; and so on.

Generally speaking, AmD systems are more aware compared with AnD systems because it can also make decisions. However, many AmD systems are not intelligent (even if they are sometimes called intelligent systems for commercial purposes). This is mainly because in most cases the events to be aware of can be pre-defined, and the decisions can be made based on some simple manually definable rules. In fact, many context aware systems developed for mobile computing can be achieved simply by using some wireless sensors (known as smart sensors) connected to a server through a base-station (BS). These systems are not really intelligent because they are just "programmed".

With in rapid progress of information and communication technology (ICT), many aware systems have been developed for providing different services. Most aware systems are connected and related to each other directly or indirectly through internet and/or intranet. We human being, as the most intelligent swarm ever appeared in the planet earth, is constructing a global-scale awareness server (GSAS) that may provide *any service to anyone, anywhere and anytime*. Aware systems developed so far are nothing but sub-systems of the GSAS. Although GSAS is also an AmD (type-III) system, it is and will be much more intelligent than any existing aware systems.

GSAS is still expanding, and becoming more and more intelligent. The main driven force for making GSAS more and more intelligent is actually the needs of users. Note that individual users nowadays are equipped with different computing devices (desktop, laptop, handtop and palmtop devices), and many different group users have been and will be created through interaction of individual users. As the user number and user type increase, the interaction between different users becomes more and more complex. To meet the needs of all kinds of users, GSAS must be a "general problem solver" (GPS).

To obtain a GPS has been a dream of many AI researchers for decades [17]. We may think that this dream can be easily realized now because computing technology of today is much more powerful than that of 1950s. However, this is not true, because the main problem related to building a GPS is NP-complete, and increasing the computing power alone cannot solve the problem.

One heuristic method for building a GPS is divide-and-conquer (D&C). Briefly speaking, D&C first breaks down big problems into small ones, solves the small problems, and then puts the results together. Based on this concept, we can design many sub-systems for different problems first. Each sub-system is just a specialist in some restricted field. To solve big problems, solutions provided by the sub-systems must be integrated. There are two approaches for integration. One is "centralized" approach, and another is "decentralized" approach. In the former, there is a "governor" (server) to coordinate all sub-systems (clients). The governor should be able to divide any given problems, distribute the small problems to the sub-systems, and integrate the solutions obtained from the sub-systems. However, to solve large

scale complex problems, designing a powerful governor itself is extremely difficult, and D&C must be used recursively.

In the decentralized approach, all sub-systems are connected organically to form a network. This network as a whole can solve any given problems. One good example of such a network is our human brain, and this is one of reasons why neural network has attracted great attention (again) in the late 1980s. GSAS is another example. In GSAS, many sub-systems have been and will be added for different applications. The system as a whole will soon or late become a GPS.

Note that no one can construct GSAS by him/herself. GSAS is being constructed by the human swarm little-by-little and step-by-step like the army ant in Africa eating a big animal. All persons are divided into two classes, namely the users and the producers. Each user, individual or group, is an AmD sub-system in GSAS, and raises "problems" in the form of "requests". Each producer is an AmD sub-system, too, and provides "solutions" in the form of "services".

Constructing GSAS is a competitive and also cooperative task. The users and producers compete with each other. To win this competition, users cooperate with each other to form larger and larger social groups, and pose more and more difficult problems. On the other hand, producers may cooperate with each other to find new solutions. Both users and producers are driven by their desires to obtain more and more profits, and this is why they are contributing to the project enthusiastically. This "cold war" relation between users and producers can be considered as the true driven force for GSAS to be more and more intelligent, and finally become a GPS.

Note that GSAS can become a GPS only if the sub-systems are connected "organically". That is, GSAS may not become a GPS "automatically". To ensure that GSAS expands in the correct direction, some regulations are necessary to control the growth of the system. That is, we should have some future vision about the architecture of the system, so that GSAS will not go in a wrong direction.

In this paper, we try to study some fundamental problems related to awareness. We do not intend to provide a survey of existing results. Rather, we will study this field from a new perspective, try to clarify some related terminologies, investigate the basic architectures, and propose some problems to solve for creating intelligence through computational awareness.

## 2 Clarification of Terminologies

According to Wikipedia [18], "*Awareness is a relative concept. An animal may be partially aware, subconsciously aware, or acutely aware of an event. Awareness may be focused on an internal state, such as a visceral feeling, or on external events by way of sensory perception. Awareness provides the raw material from which animals develop qualia or subjective ideas about their experience.*"

In computational awareness, we may define awareness as a mechanism for obtaining information or materials which are useful for human users, for other systems, or for other parts of the same system, to make decisions. In general, awareness does not necessarily lead directly to understanding. Awareness may have many levels (see Fig. 1), and we believe that "understanding" and "intelligence" can be created by some high level awareness.

Several concepts have very close relationship with awareness. The first one is perception. According to Wikipedia [19], "*perception is the process of attaining awareness or understanding of the environment by organizing and interpreting sensory information*". This concept has different meanings in psychology, neuroscience, and philosophy. In any case, perception is used mainly for brain related activities. In natural languages, perception may also mean "understanding".

In computational awareness, we may define perception as a mechanism for obtaining or receiving sensory data. Some fundamental processing of the sensory data can be included in perception, but understanding will not be produced at this stage. Among many levels of awareness, perception is defined as the sensory level awareness, which is the lowest level and serves as an interface between the aware system and the outside world.

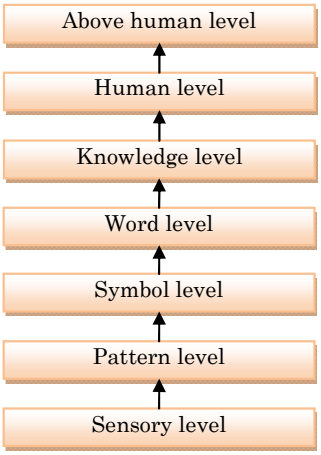



**Fig. 1.** Levels of awareness

Another related concept is cognition. Again, according to Wikipedia [20], "*cognition refers to mental processes. These processes include attention, remembering, producing and understanding language, solving problems, and making decisions*". Compared with perception, cognition is often used for understanding more abstract concepts and for solving problems based on logic thinking. In computational awareness, we can also follow this line, and consider *cognition as the human level awareness*, which is the highest level awareness we human being can achieve.

People are often confused about the relation between awareness and consciousness. "*Consciousness is a term that refers to the relationship between the mind and the world with which it interacts. It has been defined as: subjectivity, awareness, the ability to experience or to feel, wakefulness, having a sense of selfhood, and the executive control system of the mind*" (from Wikipedia [21]).

In computational awareness, we may define consciousness as the wakefulness of an aware system to its current situation. A system is wakeful if it, given a certain situation, knows what to do and why. Thus, *consciousness is a relatively high level awareness*. For example, we are aware of heart-beating but subconsciously.

Bees build their nests subconsciously, but human build their homes consciously. Usually, when we become conscious, we have already collected enough information for reasoning.

"Kansei" (or Ganshing in Chinese) is used by Japanese researchers to mean something that can be understood in mind, but cannot be described well in language. "*Kansei engineering is a method for translating human feelings and impressions into product parameters*" (Wikipedia [22]). Roughly speaking, Kansei is the human feeling that cannot be described logically. In this sense, *Kansei should be a pattern level awareness* (Fig. 1). The purpose of Kansei engineering is to transform Kansei to a high level awareness to make it more understandable.

In general, awareness has many levels. Each level produces materials or information for the upper level awareness to produce more complicated materials or information. In computational awareness, a*wareness above the word (not necessarily written) level can be defined as intelligence*. Many animals may use a limited number of spoken words, but they are not intelligent.

In each level, there are different awareness systems with different scales for achieving awareness of different concepts. The concepts can be expressed by or translated to some kind of symbols. Human users can understand the meanings of the symbols directly or indirectly through some kind of analysis, and can reason and solve problems based on these symbols.

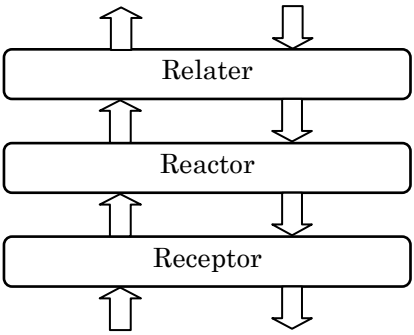



**Fig. 2.** Structure of an AU

## 3 Architecture of Awareness Systems

In this paper, we define an aware unit (AU) as an AmD (type-III) aware system. A conceptual structure of an AU is shown in Fig. 2. It can be mathematically described as follows:

$$y = R_3(R_2(R_1(x)))$$

where $R_1$ is a receptor, $R_2$ is a reactor, and $R_3$ is a relater (see Fig. 2). The input $x$ and the output $y$ are usually represented as real vectors. Each element of $x$ can come from a physical sensor, a software sensor, or a lower level AU. Each element of $y$ is a concept to be aware of or some information to be used by a higher level AU.

The purpose of the receptor $R_1$ is to receive data from outside, filter out irrelevant noises, enhance the signals, and normalize or standardize the inputs, so that inputs from different kinds of sensors can be treated in the same way. The purpose of the reactor $R_2$ is to take reaction to a given input, and extract/select important features. The purpose of the relater $R_3$ is to detect certain events, and make proper decisions based on features provided by $R_2$, and relate the detected events to other AUs. In fact, the receptor is a NanD system; and a receptor plus a reactor forms an AnD system.

Note that the data flow both forward and backward in an AU. An AU has two different modes, namely working mode and learning mode. In the working mode, the AU receives sensory inputs, and makes proper decisions. In the learning mode, the AU receives feedback from the higher level AUs, and sends feedback to lower level AUs. This happens also inside the AU itself. That is, the receptor receives feedback from the reactor; and the reactor receives feedback from the relater. System parameters can be adjusted based on the feedback.

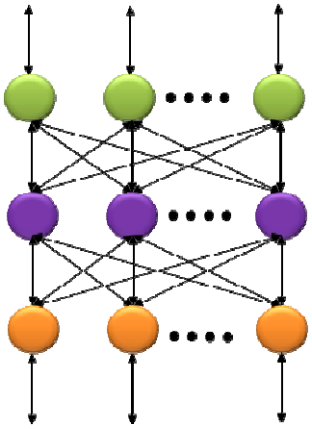


**Fig. 3.** A networked aware system

An AU itself can be used as an aware system; or it can be used as a sub-system and form a larger system with other AUs. Fig. 3 shows an example of a networked aware system, which is similar to a multilayer feedforward neural network (MFNN). All AUs in the input (bottom) layer realize the receptor of the whole system, AUs in the hidden (internal) layer together realize the reactor, and AUs in the output (top) layer realize the relater. This system again can be used as an AU to form larger systems.

An MFNN is also an aware system, in which a neuron is the smallest AU. Therefore, an MFNN is a natural model of aware systems. However, MFNN is NOT comprehensible, because the learned concepts (especially those in the hidden nodes) and the relation between concepts learned in different layers cannot be interpreted easily. Thus, in computational awareness, we should also study other models to obtain more comprehensible systems.

Note that most existing aware systems take the structure of Fig. 2. For example, in a typical context-aware system [6], the receptor may contain many sensors for collecting different information; the reactor may contain a context repository for

storing important contexts selected from the input data; and the relater may contain a production system for sending proper contexts to proper users, or giving proper commands to proper actuators.

One defect of most existing context-aware systems is that they are just "designed" or "programmed". When the contexts to be aware of are complex and dynamically changing, the system must be able to learn and become more and more aware autonomously. This is an important topic for further study in computational awareness.

Based on the proposed AU model, any aware system can be connected through internet or intranet with other aware systems to form a larger system for providing a large variety of contexts. The larger system, of course, may not be owned by a single company or organization. This is not important for the users, as long as they can get proper services with proper prices. This is actually the true concept of cloud computing (i.e., any user can get any service anywhere and anytime, without knowing where is the provider), and should be promoted further.

We may consider an aware system formed through internet as a virtual AU. In this virtual AU, the nodes are correlated through the relaters of the nodes. The virtual AUs can form a still higher level AU through internet. That is, internet provides a flexible way to form higher and higher level AUs. This poses many related problems (e.g. security, privacy, ownership of resources, etc.), and these will be important topics for research in computational awareness.

## 4 Basic Problems to Solve

Note that our aim is to create intelligence using computational awareness, and at the same time, make aware systems more intelligent. For this purpose, ad hoc approaches which are often used in the awareness computing community are not enough. We should understand the physical meaning of different problems first, and then propose different approaches for solving them. At the first glance, there are so many problems to solve. However, if we classify them properly, we may see that the problems actually belong to a very limited number of categories. Although there can be many ways for classifying computational awareness related problems, we just propose one example here as a starting point for further discussion.

### 4.1 The "For What" Problem: Awareness for What?

The first problem we must consider is the purpose of awareness. Suppose that a system is aware of a piece of important information (context or concept). The system may use this information to help the user to avoid some danger, to avoid wasting time or money in doing something, to get more opportunity for success, etc.; or the system may help the producer to maximize its profit, to avoid attacks from malicious users, to get a good business chance, etc.

To solve the "for what" problem, it is necessary to build a correlation map between the input (context or concept) and the output (possible goal). It is relatively easy to find the correlation map if the number of possible outputs is small and the outputs can be derived directly from the input. In practice, however, there can be many

unforeseen outputs (e.g. outputs not registered in the system, like the big Tsunami for destroying Fukushima nuclear power station), and the current input maybe the factor for many outputs to occur. In the former case, the aware system should be able to detect possible new outputs (novelty detection); and in the latter case, the system should be able to modify the correlation map dynamically, so that the scope of possible outputs can be narrowed when more information is acquired.

From the machine learning point of view, the "what for" problem can be solved by adding two functional modules in an aware system. The first one is a novelty detection module for detecting possible new outputs given some inputs; and the second is a reasoning module for predicting possible results given a sequence of inputs. For instance, support vector machine (SVM) can be used for the former, and Bayesian network (BN) can be used for the latter.

The "what for" problem can also be understood as follows. Suppose that the user has some goal (purpose or intention) when he/she uses an aware system. If the system is aware of the goal in an early stage, the system can provide services proactively, and the user can be more efficient in reaching the goal. In addition, if the user does not have a clear goal, the system can help him/her to formulate the goal. Goal or intention awareness can be achieved by asking user feedbacks, or the system may just record the history of the user, and guess the goal autonomously. Again, the system should have two function modules. One is to find possible new goals from the current situation; and the other is to modify the predicted goal based on a sequence of situations.

To be aware of the user goal, current situation alone is not enough. This is because different users may have different goals even under the same situation. Thus, goal awareness is closely related to user modeling. Specifically, the system should be aware of the physical and/or mental status of the user. This problem will be discussed next.

### 4.2 The "For Whom" Problem: Awareness for Whom?

User awareness is important to provide personalized services. For different users, they need different services even under the same situation. To be aware of the users, user modeling is important and has been studied extensively in the literature [23][24] [25]. However, existing results are still not enough. In fact, user modeling is a very difficult task because it is related to several factors, namely the human factor, the social factor, the context factor, the spatial-temporal factor, and so on. The problem is difficult even if we focus on the human factor alone. For example, human emotion is difficult to be aware of because even for the same person, his/her emotion can be different if the situation or time is different.

If the user is a group (e.g. all persons holding the stocks of Toyota Company, all iPhone users, and all people interested in computational awareness), it is more difficult to model the user. For example, it is difficult to predict Toyota stock price of next month; it is difficult to know what will happen next year for the market share of iPhone; and it is difficult to know what new results will be obtained next year in computational awareness. In ubiquitous computing, it is important to predict the collective behavior of different group users. Thus, user modeling or user awareness will continue to be a hot topic in the coming years.

### 4.3 The "Of What" Problem: Awareness of What?

So far, in the field of awareness computing, the event to be aware of is often specified by the user, and the system just searches related information for the user from the database or captures the information from related sensors (including software sensors like a search agent). In practice, however, it may be difficult to specify the event in advance. As an example, let us consider brainstorming. In brainstorming, when we see or hear something, we human being may be aware of some interesting information for producing good ideas. This kind of information cannot be pre-defined and must be captured in real time. As another example, let us consider the case when the system is aware of the goal of a user for doing something. The system may try to propose a plan for the user to reach the goal. The goal may be reached in several steps, and in each step, the system needs to dynamically define the event to be aware of. In this sense, "of what" is a lower level problem compared with the "for what" problem.

Solving the "of what" problem is also important for making an aware system more comprehensible. Let us consider a multi-level aware system. As in a multilayer neural network, even if we can define or be aware of the concepts in the last (output) level, it is usually difficult to define those in the hidden level(s). If we can, we will be able to describe the input-output relation of each hidden unit using a symbolized concept, and a reasoning process can be provided for any decision made by the system. Thus, solving the "of what" problem can make the system more comprehensible or understandable for human.

### 4.4 The "With What" Problem: Awareness with What?

Now suppose that we have already defined the event to be aware of. The next question is that what kind of inputs shall we collect? Even if the inputs are given, which inputs are the most informative? Without knowing the correct inputs, we can only design a system that uses all kinds of inputs, or part of the inputs. The former will not be efficient, and the latter will not be effective. Thus, an aware system should be able to learn to extract and select the most important features for making decisions.

The "with what" problem is partly related to the well-known feature selection problem that has been studied in the context of machine learning. However, existing results are not enough. In ubiquitous computing, since the working environment of an aware system changes constantly, we must consider the plasticity-stability problem seriously. That is, we cannot just select against features that are not important for the time being. We must consider the long-term performance of the system.

Another related problem is how to produce (extract) useful features. Remember that in an AU, it is the reactor that produces useful features for the relater to make decisions. The reactor not just select inputs and passes them to the relater, it also produces more useful features by combining existing inputs, linearly or non-linearly. This topic has been studied in the context of dimensionality reduction and feature extraction. The main purpose is to represent the input-output relation in a more compact way, so that the relater can make decisions more efficiently. If some *a priori* information is available to the reactor, the produced features can also enable the relater to make decision more effectively.

When we consider a multi-level aware system, the hidden level units as a whole can be considered as the "reactor" of the system. Thus, the "of what" problem is closely related to the "with what" problem. For the former, we are interested in how to symbolize the hidden units; and for the latter, we are interested in how to produce proper concepts using the hidden units. In a dynamically changing computing environment, both problems are subject to change, and there should be a mechanism for updating the produced concepts and the corresponding symbols, and at the same time for preserving the stability of the whole system.

## 5 Conclusions

In this paper, we proposed a new classification method of existing aware systems. Compared with the event-based classification, the proposed method is more general and more scientific, and can provide a unified way for studying different aware systems regardless where the system is applied. We then clarified several terminologies related to computational awareness. We think that standardization of the terminologies is necessary for us to study different aware systems in a common framework.

We also proposed a general model of aware unit (AU). This model can be used to describe any existing aware systems, and at the same time, can be used to construct different systems in the future. Starting from the lowest level AU (a physical sensor), we can construct any aware systems of any size with any functions. Here, we do not provide mathematic proof for this, because a multilayer neural network is a special case of the proposed model, and its computing power has already been proved [26].

Based on the AU model, we then provided several problems related to computational awareness. Although these problems have been studied in related fields to some extent, we think existing results are still not enough. To create intelligent using computational awareness, or to make aware systems more intelligence, we should reconsider these problems, and propose more efficient and effective algorithms.

Note that this paper is the first try for putting different awareness in one common framework. We think it is just a starting point for discussion and for further improvement. Through discussion we may someday create real intelligence, which is our final goal.

**Acknowledgements.** This paper is an extended version of the keynote speech delivered in IJCCI2011. I would like to express my great thank to Prof. Kurosh Madani for providing me a chance for delivering the keynote speech, and to INSTICC for its great support both during and after the conference.

## References

1. Start of the 5th Generation Computer Project for Knowledge-Based Information Processing, Computer Museum, Information Processing Society of Japan, http://museum.ipsj.or.jp/en/computer/other/0002.html
2. Markopoulos, P., Ruyter, B.D., Mackay, W.: Awareness Systems. Springer (2009)
3. Jajodia, S., Liu, P., Swarup, V., Wang, C.: Cyber Situational Awareness. Springer (2010)

4. Miraoui, M., Tadj, C., Amar, C.B.: Architectural survey of context-aware systems in pervasive computing environment. Ubiquitous Computing and Communication Journal 3(3), 1–9 (2008)
5. Schilit, B.N., Adams, N., Want, R.: Context-ware computing applications. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 1–7 (December 1994)
6. Schmohl, R., Baumgarten, U.: A generalized context-aware architecture in heterogeneous mobile computing environments. In: Proc. the 4th International Conference on Wireless and Mobile Communications, pp. 118–124 (2008)
7. Singh, A., Conway, M.: Survey of context aware frameworks – analysis and criticism. UNC-Chapel Hill, Information Technology Services Version 1 (2006)
8. Truong, H.L., Dustdar, S.: A survey on context-aware web service systems. Journal of Web Information Systems 5(1), 5–31 (2009)
9. Winograd, T.: Architectures for context. Human-Computer Interaction Journal 16(2) (December 2001)
10. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
11. Barrett, K., Power, R.: State of the art: context management. State of Art Surveys, Release 2 (May 2003)
12. Biegory, G., Cahill, V.: A framework for developing mobile, context-aware application. In: Proc. 2nd IEEE Annual Conference on Pervasive Computing and Communications (2004)
13. Chien, B.C., He, S.Y., Tsai, H.C., Hsueh, Y.K.: An extendible context-aware service system for mobile computing. Journal of Mobile Multimedia 6(1), 49–62 (2010)
14. Schmidt, K.: The problem with "awareness", Computer Supported Cooperative Work. The Journal of Collaborative Computing 11(3-4), 285–298 (2002)
15. Media Space, http://en.wikipedia.org/wiki/Media_space (edited in August 2010)
16. Cook, D.J., Das, S.K.: How smart are our environments? An updated look at the state of the art. Pervasive and Mobile Computing 3(2), 53–73 (2007)
17. Newell, A., Shaw, J.C., Simon, H.A.: Report on a general problem-solving program. In: Proceedings of the International Conference on Information Processing, pp. 256–264 (1959)
18. Awareness, Wikipedia, http://en.wikipedia.org/wiki/Awareness (edited in March 2009)
19. Perception, Wikipedia, http://en.wikipedia.org/wiki/Perception
20. Cognition, Wikipedia, http://en.wikipedia.org/wiki/Cognition (edited in June 2009)
21. Consciousness, Wikipedia, http://en.wikipedia.org/wiki/Consciousness
22. Kansei Engineering, Wikipedia, http://en.wikipedia.org/wiki/Kansei_engineering (edited in August 2008)
23. Santos, E.J., Nguyen, H.: Modeling Users for Adaptive Information Retrieval by Capturing User Intent. In: Chevalier, M., Julien, C., Soulé, C. (eds.) Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling, pp. 88–118. IGI Global (2009)
24. Chen, Y., Hou, H.L., Zhang, Y.-Q.: A personalized context-dependent Web search agent using Semantic Trees. In: Fuzzy Information Processing Society, Annual Meeting of the North American, pp. 1–4 (2008)
25. Zhongming, M.A., Pant, G., Liu Sheng, O.R.: Interest-based personalized search. ACM Transactions on Information Systems 25(1), Article 5, 1–38 (2007)
26. Hornik, K.: Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks 4(2), 251–257 (1991)

# Concepts and Design of Granular Models: Emerging Constructs of Computational Intelligence

Witold Pedrycz[1,2]

[1] Department of Electrical & Computer Engineering, University of Alberta, Edmonton, AB, Canada
[2] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
wpedrycz@ualberta.ca

**Abstract.** In spite of their striking diversity, numerous tasks and architectures of intelligent systems such as those permeating multivariable data analysis (e.g., time series, spatio-temporal, and spatial dependencies), decision-making processes along with their underlying models, recommender systems and others exhibit two evident commonalities. They promote (a) human centricity and (b) vigorously engage perceptions (rather than plain numeric entities) in the realization of the systems and their further usage. Information granules play a pivotal role in such settings. Granular Computing delivers a cohesive framework supporting a formation of information granules and facilitating their processing. We exploit an essential concept of Granular Computing: an optimal allocation of information granularity, which helps endow constructs of intelligent systems with a much-needed conceptual and modeling flexibility.

The study elaborates in detail on the three representative studies. In the first study being focused on the Analytic Hierarchy Process (AHP) used in decision-making, we show how an optimal allocation of granularity helps improve the quality of the solution and facilitate collaborative activities (e.g., consensus building) in models of group decision-making. The second study concerns a formation of granular logic descriptors on a basis of a family of logic descriptors Finally, the third study focuses on the formation of granular fuzzy neural networks – architectures aimed at the formation of granular logic mappings.

**Keywords:** Granular computing, Design of information granules, Human centricity, Principle of justifiable granularity, Decision-making, Optimal allocation of information granularity.

## 1 Introduction

Let us consider a system (process) for which constructed is a family of models. The system can be perceived from different points of view, observed over some time periods and analyzed at different levels of detail. Subsequently, the resulting models are built with various objectives in mind. They offer some particular, albeit useful views at the system. We are interested in forming a *holistic* model of the system by taking advantage of the individual sources of knowledge – models, which have been

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 15–29.
DOI: 10.1007/978-3-642-35638-4_2 

constructed so far. When doing this, we are obviously aware that the sources of knowledge exhibit diversity and hence this diversity has to be taken into consideration and carefully quantified. No matter what the local models may look like, it is legitimate to anticipate that the global model (say, the one formed at the higher level of hierarchy) is more general, abstract. Another point of interest is to engage the sources of knowledge in intensive and carefully orchestrated procedures of knowledge reconciliation and consensus building. Granularity of information [1]; , [2]; [3]; [4]; [6]; [18]; [17]; [19]; [20] becomes of paramount importance, both from the conceptual as well as algorithmic perspective, in the realization of granular fuzzy models. Subsequently, processing realized at the level of information granules gives rise to the discipline of Granular Computing [1]. We envision here a vast suite of formal approaches of fuzzy sets [19] rough sets [9]; [10]; [11], shadowed sets [13]; [15]; [16], probabilistic sets [7]; [8] and alike. Along with the conceptual setups, we also encounter a great deal of interesting and relevant ideas supporting processing of information granules. For instance, in the realm of rough sets we can refer to [9] From the algorithmic perspective, fuzzy clustering [6], rough clustering, and clustering are regarded as fundamental development frameworks in which information granules are constructed.

Computational Intelligence (CI) has been around for several decades and covered a broad spectrum of design and analysis of intelligent systems. What has not been fully posed on the agenda of CI deals with a spectrum of problems inherently involving information granules. Those entities are helpful in the development of distributed systems, making existing models more realistic and capable of quantifying time-variant phenomena as well capture the data. The objective of this study is to introduce conceptual and algorithmic underpinnings of Granular Computing overarching the domain of CI in the form of an optimal allocation of information granularity and show a way in which granular mappings and their diversity augment the commonly present constructs of CI such as e.g., neural networks. The essentials of the optimal allocation of information granularity are outlined along with the optimization problems and its associated objective functions (Section 2). A family of protocols of allocation of information granularity is covered in Section 3. In Section 4, we focus on the role of distribution of information granularity in the AHP models of decision-making making them granular. In Section 5, we discuss an idea of granular logic and discuss how it emerges as a result of a global view at a collection of local logic descriptors while Sections 6 and 7 are concerned with granular fuzzy neural networks. Conclusions of the study are covered in Section 8.

## 2 Optimal Allocation of Information Granularity

Information granularity is an important design asset. Information granularity allocated to the original numeric construct elevates a level of abstraction (generalizes) of the original construct developed at the numeric level. It helps the original numeric constructs cope with the nature of the data. A way in which such an asset is going to be distributed throughout the construct or a collection of constructs to make the

abstraction more efficient, is a subject to optimization. We start with a general formulation of the problem and then show selected realizations of the granular mappings.

Let us consider a certain multivariable mapping y = f(**x**, **a**) with **a** being an n-dimensional vector of parameters of the mapping. The mapping can be sought (realized) as a general construct. One may think of a fuzzy model, neural network, polynomial, differential equation, linear regression, etc. The granulation mechanism ***G*** is applied to **a.** It gives rise to its granular counterpart, **A**= ***G***(**a**). Subsequently, this mapping can be described formally as follows

$$Y = \boldsymbol{G}(f(\mathbf{x},\mathbf{a})) = f(\mathbf{x}, \boldsymbol{G}(\mathbf{a})) = f(\mathbf{x}, \mathbf{A}) \quad (1)$$

Given the diversity of the underlying constructs as well as a variety of ways information granules can be formalized, we arrive at a suite of interesting constructs including granular neural networks, say interval neural networks, fuzzy neural networks, probabilistic neural networks. Likewise we can talk about granular (fuzzy, rough, probabilistic…) regression, cognitive maps, fuzzy models, just to name several constructs.

There are a number of well-justified and convincing arguments behind elevating the level of abstraction of the existing constructs. Those include: an ability to realize various mechanisms of collaboration, quantification of variability of sources of knowledge considered, better modelling rapport with systems when dealing with nonstationary environments.

Information granularity supplied to form a granular construct is a design asset whose allocation throughout the mapping can be guided by certain optimization criteria. Let us discuss the underlying optimization problem in more detail. In addition to the mapping itself, we are provided with some experimental evidence in the form of input-output pairs ***D*** = ($\mathbf{x}_k$, $t_k$), k=1, 2,…, N. Given is a level of information granularity ε, ε $\subset$ [0,1]. We allocate the available level ε to the parameters of the mapping so that the some optimization criteria are satisfied while the allocation of granularity satisfies the following balance $n\varepsilon = \sum_{i=1}^{n} \varepsilon_i$ where $\varepsilon_i$ is a level of information granularity associated with the i-th parameter of the mapping. For further processing all the individual allocations are organized in a vector format $[\varepsilon_1\ \varepsilon_2 \ldots\ \varepsilon_n]^T$.

The first criterion is concerned with the coverage of the output data $t_k$ by the outputs produced by the granular mapping. For $\mathbf{x}_k$ we compute $Y_k = f(\mathbf{x}_k, G(\mathbf{a}))$ and determine a degree of inclusion of $t_k$ in information granule $Y_k$, namely $\text{incl}(t_k, Y_k) = t_k \subset Y_k$. Then we compute an average sum of the degrees of inclusion taken over all data, that is $Q = \frac{1}{N}\sum_{k=1}^{N} \text{incl}(t_k, Y_k)$ Depending upon the formalism of information granulation, the inclusion returns a Boolean value in case of intervals (sets) or a certain degree of inclusion in case of fuzzy sets. Alluding just to sets and fuzzy sets as the formal models of information granules, the corresponding expressions of the performance index are expressed as follows,

- for sets (intervals)

$$Q = \frac{\text{card}\{t_k \mid t_k \in Y_k\}}{N} \tag{2}$$

- for fuzzy sets

$$Q = \frac{\sum_{k=1}^{N} Y_k(t_k)}{N} \tag{3}$$

Here $Y_k(t_k)$ is a degree of membership of $t_k$ in the fuzzy set of the information granule Yk.

The second criterion of interest is focused on the specificity of $Y_k$ - we want it to be as high as possible. The specificity could be viewed as a decreasing function of the length of the interval in case of set –based information granulation. For instance, one can consider the inverse of the length of $Y_k$, for instance 1/length($Y_k$), exp(-length($Y_k$)), etc. In case of fuzzy sets, one can consider the specificity involving the membership grades. The length of the fuzzy set $Y_k$ is computed by integrating the lengths of the β-cuts, $\int_0^1 \text{length}(Y_k^{\beta}) \beta d\beta$.

More formally, the two-objective optimization problem is formulated as follows. Distribute (allocate) a given level of information granularity ε so that the following two criteria are maximized

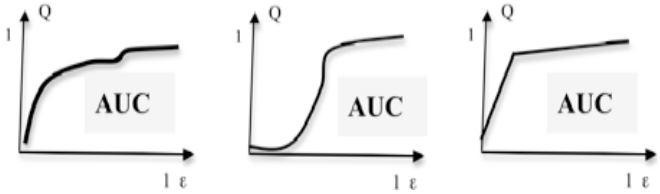
$$\begin{aligned} &\text{Maximize} && \frac{1}{N}\sum_{k=1}^{N} \text{incl}(t_k, Y_k) \\ &\text{Maximize} && g(\text{length}(Y_k)) \text{ (where g is a decreasing function of its argument)} \\ &\text{subject to} && n\varepsilon = \sum_{i=1}^{n} \varepsilon_i \end{aligned} \tag{4}$$

A simpler, optimization scenario involves a single coverage criterion. It can be regarded as an essential criterion considered in the problem

$$\begin{aligned} &\text{Maximize} && \frac{1}{N}\sum_{k=1}^{N} \text{incl}(t_k, Y_k) \\ &\text{subject to} && ne = \sum_{i=1}^{h} \varepsilon_i \end{aligned} \tag{5}$$

There is an interesting monotonicity property: higher values of ε lead to higher values of the maximized objective function, refer to Figure 1. There could be different patterns of changes of Q versus ε as illustrated in the same figure. Typically some clearly visible "knee" points are encountered on the curve beyond which the changes (increases) of Q become quite limited.

By taking into account the nature of the relationship shown in these figures, we can arrive at some global view at the relationship that is independent from a specific value of ε. This is accomplished by taking an area under curve (AUC) computed as AUC = $\int_0^1 Q(\varepsilon)d\varepsilon$. The higher the value of the AUC, the better the performance of the granular version of the mapping.




(a) (b) (c)

**Fig. 1.** Values of the coverage criterion Q regarded as a function of the assumed level of granularity ε. Shown are different relationships Q(ε) with some 'knee" points; a-c.

## 3 Protocols of Allocation of Information Granularity

An allocation of the available information granularity to the individual parameters of the mapping can be realized in several different ways depending how much diversity one would like to exploit in this allocation process. Here, we discuss several protocols of allocation of information granularity, specify their properties and show relationships between them. We assume that the parameter under discussion is denoted by "a". To focus our considerations, we consider that the values of the parameters are in the [0,1] interval. Furthermore in the following protocols we assume that the intervals generated here are included in the unit interval. The balance of the overall granularity is equal to nε with "n" being the number of the parameters of the mapping.

$P_1$: Uniform allocation of information granularity. This process is the simplest one and, in essence, does not call for any optimization. All parameters of the mapping are treated in the same way and become replaced by the interval of the same length. In terms of this generalization, we obtain an interval [a-ε/2, a+ε/2] positioned around the original numeric parameter. In essence, this allocation does not require any optimization.

$P_2$: Uniform allocation of information granularity with asymmetric position of intervals around the original parameter of the granular mapping. The allocation of this nature offers more flexibility through the asymmetric allocation of the interval.

We obtain an interval $[a-\gamma\varepsilon, a+(1-\gamma)\varepsilon]$ where $\gamma$ is an auxiliary parameter assuming values in [0,1] and controlling the position of the interval. If $\gamma=1/2$ then $P_1$ becomes a special case of $P_2$. There is only a single parameter to optimize.

$P_3$: Uniform allocation of information granularity with asymmetric position of intervals around the original connections of the network. Here each parameter is made granular however an asymmetric allocation of the corresponding interval varies from parameter to parameter. We have an interval $[a_i-\gamma_i\varepsilon, a_i+(1-\gamma_i)\varepsilon]$ associated with the i-th parameter $a_i$. The number of parameters to optimize is equal to 'n'. The balance of overall information granularity is retained.

$P_4$: Non-uniform allocation of information granularity with symmetrically distributed intervals of information granules. In this protocol, the numeric parameters are made granular by forming intervals distributed symmetrically around $a_i$ that is $[a_i-\varepsilon_i/2, a_i+\varepsilon_i/2]$. The length of the intervals ($\varepsilon_i$) could vary from parameter to parameter. The balance of information granularity requires that $n\varepsilon = \sum_{i=1}^{n} \varepsilon_i$

$P_5$: Non-uniform allocation of information granularity with asymmetrically distributed intervals of information granules. This protocol is a generalization of the previous one: we admit intervals that are asymmetrically distributed and of varying length. This leads to the interval of the i-th parameter described as $[a_i-\varepsilon_i^-, a_i+\varepsilon_i^+]$. Again we require a satisfaction of the balance of information granularity, which in this case reads as $n\varepsilon = \sum_{i=1}^{n} \varepsilon_i^- + \sum_{i=1}^{n} \varepsilon_i^+$.

$P_6$: An interesting point of reference, which is helpful in assessing a relative performance of the above methods, is to consider a random allocation of granularity. By doing this, one can quantify how the optimized and carefully thought out process of granularity allocation is superior over a purely random allocation process.

The assumption as to the [0,1] range of the parameters could be dropped. In this case, we consider a range of the parameter and include it in the above expressions. Here $\varepsilon \in [0,1]$ can be regarded as a fraction of the range of the corresponding parameter. For instance, in $P_1$ the original formula reads as [a-ε*range/2, a*range+ε/2].

The more sophisticated the protocol, the higher the coverage (and the AUC criterion) produced by running it. One has to emphasize that no matter whether we are considering swarm optimization or evolutionary techniques (say, genetic algorithms), the respective protocols call for a certain content of the particle or a chromosome. The length of the corresponding string depends upon the protocol, which becomes longer with the increased sophistication of the allocation process of information granularity.

Having considered all components that in essence constitute the environment of allocation of information granularity, we can bring them together to articulate a formal optimization process.

Assume that a certain numeric construct (mapping) has been provided. Given a certain protocol of allocation of information granularity P, determine such an allocation **I**, call it $\mathbf{I}_{opt}$, so that the value of the coverage index Q becomes maximized

$$\mathrm{Max}_{\mathbf{I}}\, Q \tag{6}$$

which is a function of ε or $\mathrm{Max}_{\mathbf{I}}$ AUC, which offers a global assessment of the protocol.

Alluding to the refinement of the protocols of allocation of information granularity, some inclusions (orderings) among $I_{opt}$ resulting from the use of the respective protocols are envisioned: $P_1 \supset P_2 \supset P_3\ P_3 \supset P_4$ where the relation Pi ⊃ Pj means that protocol $P_i$ produces weaker results than $P_j$.

The corresponding search spaces associated with the realization of the protocols (with the nested property outlined above) start exhibiting higher dimensionality.

We can think of fuzzy sets built around numeric values of the parameters where depending upon a certain the membership functions may exhibit symmetric or asymmetric character as well as come with various supports. In case of probabilistic information granules, one may talk about symmetric and asymmetric probability density functions with the modal values allocated to the numeric values of the parameters and standard deviations whose values vary from parameter to parameter. In total, we require a sum of the standard deviations to satisfy the predefined level of granularity that is $\sigma = \sum_{i=1}^{h} \sigma_i$.

## 4 Granular Analytic Hierarchy Process (AHP)

This model serves as a simple yet a convincing example in which the idea of granularity allocation can be used effectively in improving the quality of a solution both in case of a individual decision-making as well as its group version. Let us recall that the Analytic Hierarchy Process (AHP) is aimed at forming a vector of preferences for a finite set of n alternatives. These preferences are formed on a basis of a reciprocal matrix R, $R=[r_{ij}]$, i, j=1, 2, …, n whose entries are a result of pairwise comparisons of alternatives provided by a decision-maker. The quality of the result (reflecting the consistency of the judgment of the decision-maker) is expressed in terms of the following inconsistency index

$$v = \frac{\lambda_{\max} - n}{n-1} \tag{7}$$

where $\lambda_{max}$ is the largest eigenvalue associated with the reciprocal matrix. The larger the value of this index is, the more significant level of inconsistency is associated with the preferences collected in the reciprocal matrix.

We distinguish here two main categories of design scenarios: a single decision-maker is involved or we are concerned with a group decision-making where there is a collection of reciprocal matrices provided by each of the member of the group.

*A Single Decision-maker Scenario*. The results of pairwise comparisons usually exhibit a certain level of inconsistency. The inconsistency index presented above quantifies this effect. We generalize the numeric reciprocal matrix R by forming its

granular counterpart and allocating the admissible level of granularity to the individual entries of the matrix. Formally, the process can be schematically described in the following form:

$$R \xrightarrow{\varepsilon} \boldsymbol{G}(R) \tag{8}$$

where $\boldsymbol{G}(R)$ stands for the granular version of the reciprocal matrix. A certain predetermined level of information granularity e is distributed among elements of the reciprocal matrix R. More specifically, we look at the entries of the reciprocal matrix, which are below 1 and form information granules around those. Confining ourselves to intervals (for illustrative purposes), formed are intervals around the corresponding entries of the matrix whose total length satisfies the constraint $\sum_{ij} \varepsilon_{ij}=p\varepsilon$ where "p" stands for the number of elements of R assuming values below 1. Thus the original entry $r_{ij}$ is replaced by the interval whose lower and upper bound are expressed as $\max(1/9, r_{ij}-\varepsilon_{ij}(8/9))$ and $\min(1, r_{ij}+\varepsilon_{ij}(8/9))$. Here the number 9 reflects the largest length of the scale used in the realization of pairwise comparisons. For the reciprocal entry of the matrix, we compute the inverse of the lower and upper bound of the interval, round off the results to the closest integers (here we use the integers from 1 to 9) and map the results to the interval of the reciprocals. In this way an original numeric entry $r_{ij}$ and $1/r_{ij}$ are made granular. The same process is completed for the remaining entries of the reciprocal matrix.

As an illustration, let us show the calculations in case where $r_{ij}=1/3$ and $\varepsilon_{ij} = 0.10$. The numeric value is replaced by the bounds 0.24 and 0.42. The inverse produces the integers (after rounding off) being equal to 4 and 2. Mapping them again by computing the inverse produces the entry of the reciprocal matrix equal to [1/4, ½]. Summarizing, through an allocation of granularity, the original entries 1/3 and 3 were replaced by their granular (interval) counterparts of [1/4, ½] and [2, 4]. The resulting information granule depends upon the assumed level of granularity as well as the protocol of granularity allocation. For instance, for asymmetric allocation of granularity with $\varepsilon_{ij-}=0.1$ and $\varepsilon_{ij+}=0.2$, we arrive at the intervals [1/4, 1] and [1, 4], respectively.

The granular (interval-valued) reciprocal matrix $\boldsymbol{P}(R)$ manifest in a numeric fashion in a variety of ways. To realize such manifestation, we randomly pick up the numeric values from the corresponding intervals (maintaining the reciprocality condition, that is when a value of $r_{ij}$ has been selected from the range $[r_{ij-}, r_{ij+}]$ the value of $r_{ji}$ is computed as the inverse of the one being already selected). For the matrix obtained in this way computed is its inconsistency index. The overall process is repeated a number of times and determined is the average of the corresponding values of the inconsistency index of the matrices. Denote the average by $E(\nu)$ This average quantifies the quality of the granular reciprocal matrix being a result of allocation (distribution) of the level of information granularity ε. The goal of optimization is to minimize $E(\nu)$ by determining $\varepsilon_{ij}$ so that Min $\varepsilon_{ij}$ $E(\nu)$ subject to constraints $\sum_{i,j} \varepsilon_{ij}=p\varepsilon$.

*Group decision making.* In this situation, we are concerned with a group of reciprocal matrices R[1], R[2],…, R[c] along with the preferences (preference vectors), **e**[1], **e**[2],…, **e**[c] obtained by running the AHP for the corresponding reciprocal matrices. Furthermore the quality of preference vectors is quantified by the associated inconsistency index ν[i]. First, in the optimization problem, we bring all preferences close to each other and this goal is realized by adjusting the reciprocal matrices within the bounds offered by the admissible level of granularity provided to each decision-maker.

$$Q_1 = \sum_{i=1}^{c} (1-\nu_i) \| e[i] - \hat{\mathbf{e}} \|^2 \tag{9}$$

where $\hat{\mathbf{e}}$ stands for the vector of preferences which minimizes the weighted sum of differences ||.|| between $\boldsymbol{e}$[i] and $\hat{\mathbf{e}}$. Second, we increase the consistency of the reciprocal matrices and this improvement is realized at the level of individual decision-maker. The following performance index quantifies this aspect of collaboration

$$Q_2 = \sum_{i=1}^{c} \nu_i \tag{10}$$

These are the two objectives to be minimized. If we consider the scalar version of the optimization problem, it can arise in the following additive format $Q = gQ_1 + Q_2$ where $g \geq 0$. The overall optimization problem with constraint reads now as follows

$$\text{Min}_{R[1], R[2], \ldots, R[c] \in \boldsymbol{G}(R)} Q \tag{11}$$

subject to predetermined level of granularity ε where $\boldsymbol{G}$(R) stands for the granular version of the of the reciprocal matrix. We require that the overall balance of the predefined level of granularity given in advance ε is retained and allocated throughout all reciprocal matrices (Pedrycz and Song, 2011).

## 5 The Development of Granular Logic: A Holistic View at a Collection of Logic Descriptors

We consider a collection of logic descriptors describing some local relationships. We intend to view them at a global level to arrive at a generalized description in the form of a *granular* logic descriptor. Information granularity arises here as a result of inherent diversity within a family of individual descriptors. It also helps quantify the existing diversity as well as support some mechanisms of successive reconciliations of the sources of knowledge (logic descriptors).

Let us define a logic descriptor as a certain quantified logic expression coming in a conjunctive or disjunctive form. Given is a collection of "c" logic descriptors involving "n" variables in either a conjunctive form

$$L_{ii}: y[ii] = (w_1[ii] \text{or } x_1) \text{ and } (w_2[ii] \text{or } x_2) \text{ and } .. \text{ and } (w_n[ii] \text{or } x_n) \tag{12}$$

or a disjunctive form

$$L_{ii}: y[ii] = (w_1[ii] \text{and } x_1) \text{ or } (w_2[ii] \text{and } x_2) \text{ or } .. \text{ or } (w_n[ii] \text{and } x_n) \quad (13)$$

ii=1, 2,…, c. In the above logic descriptors, $x_1$, $x_2$,…, $x_n$ are the input variables assuming values in the unit interval and $w_j[ii]$ are the weights calibrating a level of contribution to the individual inputs, $\mathbf{w}[ii] = [w_1[ii]\ w_2[ii]\ldots\ w_n[ii]]^T$. Each logic descriptor (12)-(13) denoted here briefly as $L_1$, $L_2$, …, $L_c$ is a logic mapping from $[0,1]^n$ to [0,1]. We assume that all of them are disjunctive or conjunctive (if this does not hold, all are transformed to either of these formats). As the logic connectives are modeled by t-norms or t-conorms, the expressions shown above read as

$$L_{ii}: y[ii] = (w_1[ii] s\ x_1)\ t\ (w_2[ii] s\ x_2)\ t\ ..\ t\ (w_n[ii] s\ x_n) \quad (14)$$

or

$$L_{ii}: y[ii] = (w_1[ii] t\ x_1)\ s\ (w_2[ii] t\ x_2)\ s\ ..\ s\ (w_n[ii] t\ x_n) \quad (15)$$

We form a unified, holistic view at all of them by forming a certain granular abstraction of $\{L_1, L_2, \ldots L_c\}$, denoted here as GL(**x**), refer also to Figure 2.

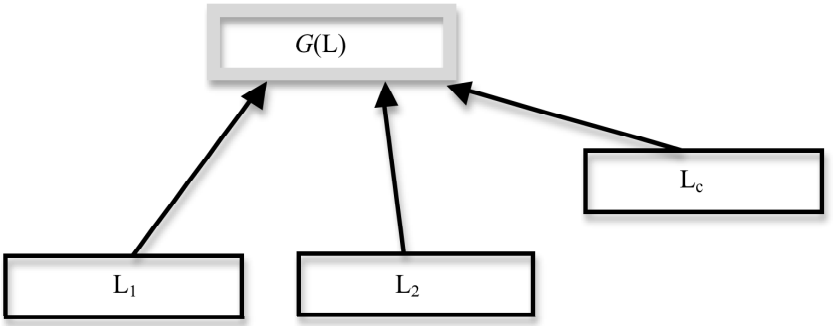



**Fig. 2.** From local logic descriptors to its global granular description *G*(L)

It reads as follows

$$L: y = (W_1 s\ x_1)\ t\ (W_2\ s\ x_2)\ t\ ..\ t\ (W_n\ s\ x_n) \quad (16)$$

or

$$L: y = (W_1 t\ x_1)\ s\ (W_2 t\ x_2)\ s\ ..\ s\ (W_n t\ x_n) \quad (17)$$

where $W_j$ is a granular weight.

The granular descriptor is developed in the following scenario. We select one of the logic descriptors, say $L_i$. If the corresponding data $\mathbf{D}_1$, $\mathbf{D}_2$, … $\mathbf{D}_c$ used to develop the logic descriptors are available, we form their union **D**, $\mathbf{D} = \bigcup_{i=1}^{c} \mathbf{D}_i$. These data are used to form the granular descriptor, $G(L_i)$ following the procedure discussed so far. The quality of this granular construct is quantified by the AUC value. If the data sets are not available (meaning that only the logic descriptors are given), one can form a new data set, **F**, using which $G(L_i)$ is developed and evaluated in terms of the AUC measure.

As noted, the granular structure is formed by starting with $L_i$. With this regard some optimization could be envisioned. All options are enumerated by choosing any of $L_1$, $L_2$, …, $L_c$ as a candidate for a granular logic descriptor and choosing the one coming with the highest AUC value, that is

$$i_{opt} = \arg\max_{i=1,2,\ldots,c} \mathrm{AUC}(G(L_i)) \tag{18}$$

## 6 An Architectures of the Fuzzy Logic Networks

The logic neurons (the constructs discussed above) can serve as building blocks of more comprehensive and functionally appealing architectures. The typical logic network that is at the center of logic processing originates from the two-valued logic and comes in the form of the fundamental Shannon theorem of decomposition of Boolean functions. Let us recall that any Boolean function $\{0,1\}^n \rightarrow \{0,1\}$ can be represented as a logic sum of its corresponding miniterms or a logic product of maxterms. By a minterm of "n" logic variables $x_1$, $x_2$, …, $x_n$ we mean a logic product involving all these variables either in direct or complemented form. Having "n" variables we end up with $2^n$ minterms starting from the one involving all complemented variables and ending up at the logic product with all direct variables. Likewise by a maxterm we mean a logic sum of all variables or their complements. Now in virtue of the decomposition theorem, we note that the first representation scheme involves a two-layer network where the first layer consists of AND gates whose outputs are combined in a single OR gate. The converse topology occurs for the second decomposition mode: there is a single layer of OR gates followed by a single AND gate aggregating *or*-wise all partial results.

The proposed network (referred here as a logic processor) generalizes this concept as shown in Figure 3. The OR-AND mode of the logic processor comes with the two types of aggregative neurons being swapped between the layers. Here the first (hidden) layer is composed of the OR neuron and is followed by the output realized by means of the AND neuron. The inputs and outputs are the levels of activation of information granules expressed in the input and output spaces.

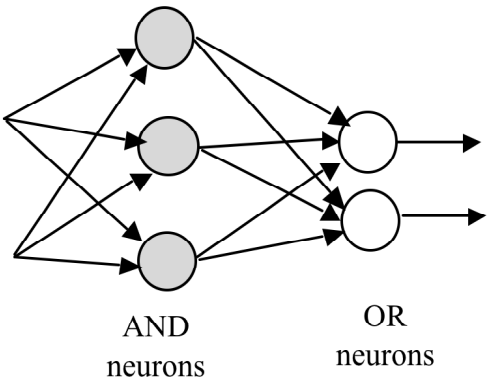



**Fig. 3.** A topology of the logic processor (N) in its AND-OR mode of realization

The logic neurons generalize digital gates by bringing essential learning capabilities and expanding the construct from its Boolean version to the multivalued alternative. The design of the network (viz. any fuzzy function) is realized through learning. If we confine ourselves to Boolean {0,1} values, the network's learning becomes an alternative to a standard digital design, especially a minimization of logic functions. The logic processor translates into a compound logic statement (for the time being we skip the connections of the neurons to emphasize the underlying logic content of the statement).

- if ($input_1$ *and*… *and* $input_n$) *or* ($input_d$ *and* …*and* $input_n$) then truth value of $B_j$

where the truth value of $B_j$ can be also regarded as a level of "satisfaction" (activation) of the information granule $B_j$. Given the number of inputs and the number of outputs equal to "n" and "m", the logic processor generates a mapping from $[0,1]^n$ to $[0,1]^m$ thus forming a collection of "m" n-input fuzzy functions.

## 7 Granular Fuzzy Logic Networks

Following the general scheme of the granular mapping, the numeric values of the connections are generalized (abstracted) to the granular connections in the form of some intervals being included in the unit interval. The emergence of the granular (interval) connections is legitimate. Again we would like to stress a role of information granularity being viewed as an important design asset, which needs to have prudently exploited. The essence of the granulation of the fuzzy logic network is visualized in Figure 4.

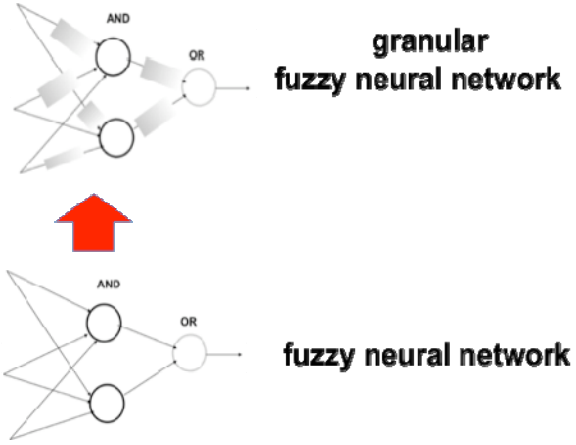



**Fig. 4.** From fuzzy neural network to its granular abstraction (generalization); small rectangular shapes emphasize the interval-valued character of the granular connections

As the connections of the logic neurons are now granular (represented in the form of intervals), the output of network becomes granular (interval) as well. To emphasize that, let us look at the OR neuron described by (17) where the connections are made granular, that is $G(w_{ij}) = [w_{ij-}, w_{ij+}]$. We have the following expression

$$Y_i = [y_{i-}, y_{i+}] = \overset{n}{\underset{j=1}{S}}(G(w_{ij})\ tu_j) \tag{19}$$

which, in virtue of the monotonicity of t-norms and t-conorms, results in the bounds of $Y_i$ to be equal to

$$Y_i = [\overset{n}{\underset{j=1}{S}}(G(w_{ij-})\ tu_j), \overset{n}{\underset{j=1}{S}}(G(w_{ij+})\ tu_j)] \tag{20}$$

The quality of the granular fuzzy neural network, assuming that a certain level of information granularity ε has resulted in the corresponding granular connections, can be evaluated in several ways.

Intuitively, as the connections are granular (interval-valued), the output produced by the network is also of interval-valued nature. The data used in the formation of the granular mapping is in the form of input-output pairs $\mathbf{D}' = (\mathbf{x}_k, \mathbf{target}_k)$. Ideally, one would anticipate that the outputs of the granular network should include the original data **D**'. Consider $\mathbf{x}_k \in \mathbf{D}'$ with the cardinality of **D**' equal to N'. Each of the outputs of the granular neural network comes in the form of the interval $Y_{kj} = [y_{j-}, y_{j+}] = G(N(\mathbf{x}_k))_j$, j=1, 2,…, m. The quality of the granular network can be assessed by counting how many times the inclusion relationship $target_{kj} \in G(N(\mathbf{x}_k))_j$ holds. In other words, the performance index we discussed so far, is expressed in the following form

$$Q = \frac{\overset{m}{\underset{j=1}{\sum}}\ \overset{N'}{\underset{k=1}{\sum}}\{card((k,j)\,|\,target_{kj} \in G(N(\mathbf{x}_k)_j\}}{N'*m} \tag{21}$$

Ideally, we could expect that this ratio is equal to 1. Of course Θ becomes a nondecreasing function of ε , Q(ε), so less specific information granules (higher values of ε) produce better coverage of the data but at an expense of the obtained results being less specific. As before, one can compute the AUC as it is independent from any specific value of the level of information of granularity.

Along with the coverage criterion, we can look at the quality of the information granule of the output formed by the granular logic network, that is a length L of the interval $L(G(N(\mathbf{x}_k))$ or its average value,

$$L = \frac{1}{M}\overset{M}{\underset{k=1}{\sum}} L(G(N(\mathbf{x}_k)) \tag{22}$$

Note that the criteria (21) and (22) are in conflict: while high values of (21) are preferred, lower values of (22) are advisable. If the two criteria are going to be

considered at a time then a formation of a Pareto front is a way to proceed in the optimization process.

In what we follows, we discuss some more advanced ways of allocating information granularity to the individual connections of the network (not all connections need to be granulated to the same extent), so that the performance indices (21) and (22) can be optimized (maximized and minimized, respectively) or their aggregate could be optimized.

## 8 Conclusions

In Granular Computing, we strive to build a coherent and algorithmically sound processing platform. The mechanism of optimal allocation of information granularity provide a way of forming information granules and exploiting information granularity as an important design asset in a variety of models. In this context, we highlight an important role of information granules as a vehicle through which we can achieve higher consistency of the models (along with a quantification of this feature) and facilitate various mechanisms of collaboration (as exemplified in the group decision-making realized via the AHP model). It is worth noting that they are independent from any specific formal way of representing information granules sets, fuzzy sets, rough sets, etc.) and in this manner, the discussed setup is of a general character.

The idea of optimal allocation (distribution) of information granularity calls for more advanced techniques of optimization (that go far beyond gradient-based techniques). In particular, one can anticipate the usage of evolutionary of swarm optimization methods. In this sense, we start witnessing here yet another example of an important synergy of technologies of Computational Intelligence. The granular constructs open a new avenue of Granular Computational Intelligence in which information granularity starts playing a visible role in the design of collaborative intelligent systems.

## References

1. Bargiela, W.P.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
2. Bargiela, W.P. (ed.): Human-Centric Information Processing Through Granular Modelling. Springer, Heidelberg (2009)
3. Bargiela, W.P.: Granular mappings. IEEE Transactions on Systems, Man, and Cybernetics-part A 35(2), 292–297 (2005)
4. Bargiela, W.P.: A model of granular data: a design problem with the Tchebyschev FCM. Soft Computing 9, 155–163 (2005)
5. Bargiela, W.P.: Toward a theory of Granular Computing for human-centered information processing. IEEE Transactions on Fuzzy Systems 16(2), 320–330 (2008)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, N. York (1981)
7. Hirota, K.: Concepts of probabilistic sets. Fuzzy Sets and Systems 5(1), 31–46 (1981)

8. Hirota, K., Pedrycz, W.: Characterization of fuzzy clustering algorithms in terms of entropy of probabilistic sets. Pattern Recognition Letters 2(4), 213–216 (1984)
9. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory. Kluwer Academic Publishers, Dordrecht (1991)
10. Pawlak, Z.: Rough sets and fuzzy sets. Fuzzy Sets and Systems 17(1), 99–102 (1985)
11. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
12. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. John Wiley, Hoboken (2007)
13. Pedrycz, W.: Shadowed sets: representing and processing fuzzy sets. IEEE Trans. on Systems, Man, and Cybernetics, Part B 28, 103–109 (1998)
14. Pedrycz, W., Song, M.: Analytic Hierarchy Process (AHP) in group Decision Making and Its Optimization with an Allocation of Information Granularity. IEEE Trans. on Fuzzy Systems (2011) (to appear)
15. Pedrycz, W.: Shadowed sets: bridging fuzzy and rough sets. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization. A New Trend in Decision-Making, pp. 179–199. Springer, Singapore (1999)
16. Pedrycz, W.: Interpretation of clusters in the framework of shadowed sets. Pattern Recognition Letters 26(15), 2439–2449 (2005)
17. Pedrycz, W., Hirota, K.: A consensus-driven clustering. Pattern Recognition Letters 29, 1333–1343 (2008)
18. Pedrycz, W., Rai, P.: Collaborative clustering with the use of Fuzzy C-Means and its quantification. Fuzzy Sets and Systems 159(18), 2399–2427 (2008)
19. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–117 (1997)
20. Zadeh, L.A.: From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions. IEEE Trans. on Circuits and Systems 45, 105–119 (1999)

# Incremental Social Learning in Swarm Intelligence Algorithms for Continuous Optimization

Marco A. Montes de Oca

Department of Mathematical Sciences, University of Delaware, Newark, DE, U.S.A.
mmontes@math.udel.edu

**Abstract.** Swarm intelligence is the collective problem-solving behavior of groups of animals and artificial agents. Often, swarm intelligence is the result of self-organization, which emerges from the agents' local interactions with one another and with their environment. Such local interactions can be positive, negative, or neutral. Positive interactions help a swarm of agents solve a problem. Negative interactions are those that block or hinder the agents' task-performing behavior. Neutral interactions do not affect the swarm's performance. Reducing the effects of negative interactions is one of the main tasks of a designer of effective swarm intelligence systems. Traditionally, this has been done through the complexification of the behavior and/or the characteristics of the agents that comprise the system, which limits scalability and increases the difficulty of the design task. In collaboration with colleagues, I have proposed a framework, called incremental social learning (ISL), as a means to reduce the effects of negative interactions without complexifying the agents' behavior or characteristics. In this paper, I describe the ISL framework and three instantiations of it, which demonstrate the framework's effectiveness. The swarm intelligence systems used as case studies are the particle swarm optimization algorithm, ant colony optimization algorithm for continuous domains, and the artificial bee colony optimization algorithm.

## 1 Introduction

Some animals form large groups that behave so coherently and purposefully that they truly seem to be superorganisms with a mind of their own [5]. These groups are often called *swarms* because the individuals that comprise them are usually of the same kind and are so numerous that they resemble true insect swarms. If the behavior of a swarm allows it to solve problems beyond the capabilities of any of its members, then we say that the swarm exhibits *swarm intelligence* [3]. One of the best known examples of swarm intelligence is the ability of ant colonies to discover the shortest path between their nest and a food source [11]. The members of a swarm usually cannot perceive or interact with all the other members of the swarm at the same time. Instead, swarm members interact with one another and with their environment only locally. As a result, a swarm member cannot possibly supervise or dictate the actions of all the other swarm members. This restriction implies that swarm intelligence is often the result of self-organization, which is a process through which patterns at the collective level of a system emerge as a result of local interactions among its lower level components [4]. Other mechanisms through which swarm intelligence may be obtained are leadership, blueprints, recipes, templates, or threshold-based responses [4, 9].

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 31–45.
DOI: 10.1007/978-3-642-35638-4_3 

Through the study of natural swarm intelligence systems, scientists have identified a number of principles and mechanisms that make swarm intelligence possible [9]. The existence of these principles and mechanisms makes the design of artificial swarm intelligence systems possible because we can make robots or software agents use the same or similar rules to the ones animals use. The first efforts toward the development of artificial swarm intelligence systems began in the 1990s with pioneering works in robotics, data mining, and optimization [6]. In this paper, I focus on swarm intelligence systems for optimization, which have been very successful in practice [31, 35].

A swarm intelligence algorithm for optimization consists of a set of agents, called swarm, or colony, that either generates candidate solutions or represents the actual set of candidate solutions. For example, in particle swarm optimization (PSO) algorithms [18], the swarm is composed of "particles" whose positions in the search space represent candidate solutions (see Section 4.1). In ant colony optimization (ACO) algorithms [7], the colony is made of "ants" that generate solutions in an incremental way guided by "pheromones" (see Section 4.2). In any case, the size of the swarm or colony is a parameter that determines the number of candidate solutions generated at each iteration of the algorithm: the larger the swarm, the more candidate solutions are generated and tried per iteration. The effect of the swarm size on the algorithms' performance depends on the amount of time allocated to the optimization task [7, 27]: If a long time is available, large swarms usually return better results than small swarms. On the contrary, if only a short amount time is allocated, small swarms return better results than large swarms. In Section 2, I provide an explanation of this phenomenon in terms of positive and negative interactions among agents. For the moment, it is enough to say that the discovery of good solutions to an optimization problem typically occurs when swarms are near a convergence state. Thus, since small swarms reach a convergence state sooner than large swarms, it follows that small swarms discover good solutions before large swarms. However, small swarms converge before the allocated time runs out, which causes search stagnation.

In the context of optimization, the incremental social learning (ISL) framework [26, 29, 24] exploits the faster convergence of small swarms while avoiding search stagnation. This is accomplished by varying the population size over time. An optimization algorithm instantiating the ISL framework starts with a small population in order to find good quality solutions early in the optimization process. As time moves forward, new individuals are added to the population in order to avoid search stagnation. The newly added individuals are not generated at random. They are initialized using information already present in the population through a process that simulates social learning, that is, the transmission of knowledge from one individual to another. These two elements, an incremental deployment of individuals and the social learning-based initialization of new individuals, are the core of the ISL framework. The actual implementation of these elements may vary from system to system but the goals of each element remain the same. ISL has not only been used in optimization but also in swarm robotics [28, 24]. In both cases, an important improvement of the system's performance has been obtained. With this paper, I hope to spark interest in the application and theoretical study of the ISL framework.

The rest of the paper is structured as follows. In Section 2, I explain the three kinds of interactions that occur in multiagent systems, including swarm intelligence systems. This explanation motivates the introduction of the ISL framework, which is described in detail in Section 3. In Section 4, I describe the three case studies that are used to show the effectiveness of the ISL framework in the context of optimization. I close this paper with some conclusions in Section 5.

## 2 Interactions in Multiagent and Swarm Intelligence Systems

In multiagent systems, including swarm intelligence systems, individual agents interact with one another and with their environment in order to perform a task. It is possible to classify all inter-agent and agent-environment interactions as "positive", "negative", or "neutral" based on whether they help the system achieve its goals or not [10]. Interactions that facilitate the accomplishment of the agents' assigned task are called positive. For example, a positive interaction would be one in which agents cooperate to perform a task that agents could not if they acted individually (see e.g., [19]). Negative interactions, also called *interference* [23], *friction* [10], or *repulsive and competitive interactions* [14], are those that block or hinder the ability of the system's constituent agents to perform the assigned task. Since negative interactions are an obstacle toward the efficient completion of a task, they decrease the performance of the system. For instance, in swarm intelligence algorithms for data clustering [13], agents can undo the actions of other agents, which increases the time needed to find a satisfactory final clustering. An interaction that does not benefit or harm progress toward the completion of a task is called neutral. An example of a neutral interaction could be a message exchange between two agents that just confirms information they already have and thus do not have to change plans.

Three difficulties arise when trying to directly measure the effects of interactions in a multiagent system. First, in many systems agent interactions are not predictable, that is, it is impossible to know in advance whether any two agents will interact and whether they will do so positively, negatively, or neutrally. Consequently, one can determine whether the effects of an interaction are beneficial or not only after the interaction has occurred. Second, an interaction may be positive, negative or neutral, depending on the time scale used to measure its effect. For example, an interaction that involves two robots performing collision avoidance can be labeled as a negative interaction in the short term because time is spent unproductively. However, if the time horizon of the task the robots are performing is significantly longer than the time frame of a collision avoidance maneuver, then the overall effect of such an interaction may be negligible. In this case, such interaction may be labeled as neutral. Third, the nature of the interactions themselves poses a challenge. In some systems, agents interact directly on a one-to-one or one-to-many basis. In other systems, agents interact stigmergically [12], that is, indirectly through the environment. Stigmergy makes the classification of interactions difficult because there may be extended periods of time between the moment an agent acts and the moment another agent (or even the acting agent itself) is affected by those actions. With these difficulties, the only practical way to measure the effects of interactions is to do it indirectly through the observation of the system's performance.

This is the approach my colleagues and I have taken to measure the effects of ISL (see Section 4).

Swarm intelligence systems are special kinds of multiagent systems. In contrast with traditional multiagent systems in which agents usually play a very specific role, swarm intelligence systems are usually composed of identical individuals. This feature has profound effects on the difficulty of the design task. The main problem is that the designer of a swarm intelligence system has to devise individual-level behaviors that foster positive interactions and, at the same time, minimize the number of negative interactions. Unfortunately, it is not always possible to achieve both goals simultaneously. For example, Kennedy and Eberhart [18], the designers of the first PSO algorithm, pondered different candidate particle interaction rules before proposing the rules that we now know (see Section 4.1). Their ultimate goal was to design rules that promoted positive interactions between particles. In the final design, particles cooperate, that is, they engage in positive interactions, by exchanging information with one another about the best solution to an optimization problem that each particle finds during its lifetime. It is hoped that this information exchange helps the algorithm improve the quality of the solutions by making particles move toward promising regions in the search space. At the same time, however, such an exchange of information may make particles evaluate regions of the search space that may in fact not contain the optimal solution or improve their current best estimate. When this happens, objective function evaluations are spent unproductively. The trade-off between solution quality and speed that many optimization algorithms exhibit is the result if these opposite-effect processes. As I said earlier, it is not possible to know in advance which particle interactions will be positive, or negative and thus a balance between these two kinds of interactions is always sought, usually through appropriate parameter settings [21].

Despite the aforementioned difficulties, swarm intelligence systems often exhibit the following two properties that make the management of negative interactions possible:

1. The number of negative interactions increases with the number of agents in the system. This effect is the result of the increased number of interactions within the system. The larger the number of agents that comprise the system, the more frequently negative interactions occur.
2. The number of negative interactions tends to decrease over time. At one extreme of the spectrum, one can find a system in which interactions between agents are completely random or not purposeful. In such a case, it is expected that agents cannot coordinate and thus, cannot perform useful work. As a result, the number of negative interactions remains constant over time. At the other extreme of the spectrum, one finds well-behaved systems consisting of a number of agents whose interaction rules are designed in order to make agents coordinate with each other. Initially, it is expected that many negative interactions occur because agents would not have enough knowledge about their current environment. However, over time, the behavioral rules of these agents would exploit any gained knowledge in order to make progress toward the completion of the assigned task. Thus, in cases like these, the number of negative interactions decreases over time.

The incremental social learning framework, which will be described next, exploits the two aforementioned properties in order to control, up to a certain extent, the number of negative interactions in a swarm intelligence system.

## 3 Incremental Social Learning

A framework called incremental social learning (ISL) was proposed by the author and colleagues [26,29,24] to reduce the effects of negative interactions in swarm intelligence systems. As a framework, ISL offers a conceptual algorithmic structure that does not prescribe a specific implementation of the ideas on which it relies. Each instantiation of ISL will benefit from knowledge about the specific application domain, and therefore, specific properties of the framework should be analyzed in an application-dependent context.

The ISL framework consists of two elements that exploit the two properties mentioned in Section 2. The first element of the framework directly reduces the number of negative interactions within a system by manipulating the number of agents. The strategy for controlling the size of the agent population exploits the second property, that is, that the number of negative interactions tends to decrease over time. Under the control of ISL, a system starts with a small population. Over time, the population grows at a rate determined by a user-defined agent addition criterion. Two phenomena with opposite effects occur while the system is under the control of the ISL framework. On the one hand, the number of negative interactions increases as a result of adding new agents to the swarm (first property described in Section 2). On the other hand, the number of negative interactions decreases because the system naturally tends toward a state in which fewer negative interactions occur (second property described in Section 2). The second element of the framework is social learning. This element is present before a new agent freely interacts with its peers. Social learning is used so that the new agent does not disrupt the system's operation due to its lack of knowledge about the environment or the task. Leadership, a swarm intelligence mechanism [4,9], is present in the framework in the process of selecting a subset of agents from which the new agent learns. The best strategy to select such a set depends on the specific application. However, even in the case in which a random agent is chosen as a "model" to learn from, knowledge transfer occurs because the selected agent will have more experience than the new agent that is about to be added.

The two elements that compose ISL are executed iteratively as shown in Algorithm 1. In a typical implementation of the ISL framework, an initial population of agents is created and initialized (line 4). The size of the initial population depends on the specific application. In any case, the size of this initial population should be small in order to reduce interference to the lowest possible level. A loop allows the interspersed execution of the underlying system and the creation and initialization of new agents (line 7). This loop is executed until some user-specified stopping criteria are met. Stopping criteria can be specific to the application or related to the ISL framework. For example, the framework may stop when the task assigned to the swarm intelligence system is completed or when a maximum number of agents are reached. While executing the main loop, agent addition criteria, which are also supplied by the user, are repeatedly evaluated (line 8). The criteria can range from a predefined schedule to conditions based on

**Algorithm 1.** Incremental social learning framework

**Input:** Agent addition criteria, stopping criteria
1: /* Initialization */
2: $t \leftarrow 0$
3: Initialize environment $\mathbf{E}^t$
4: Initialize population of agents $\mathbf{X}^t$
5:
6: /* Main loop */
7: **while** Stopping criteria not met **do**
8: **if** Agent addition criteria is not met **then**
9: default($\mathbf{X}^t, \mathbf{E}^t$) /* Default system */
10: **else**
11: Create new agent $a_{new}$
12: slearn($a_{new}, \mathbf{X}^t$) /* Social learning */
13: $\mathbf{X}^{t+1} \leftarrow \mathbf{X}^t \cup \{a_{new}\}$
14: **end if**
15: $\mathbf{E}^{t+1} \leftarrow$ update($\mathbf{E}^t$) /* Update environment */
16: $t \leftarrow t + 1$
17: **end while**

statistics of the system's progress. If the agent addition criteria are not met, the set of agents work normally, that is, the underlying swarm intelligence system is executed. In line 9, such an event is denoted by a call to the procedure default($\mathbf{X}^t, \mathbf{E}^t$). If the agent addition criteria are satisfied, a new agent is created (line 11). In contrast to a default initialization such as the one in line 4, this new agent is initialized with information extracted from a subset of the currently active population (line 12). Such an initialization is denoted by a call to the procedure slearn($a_{new}, \mathbf{X}^t$). This procedure is responsible for the selection of the agents from which the new agent will learn, and for the actual implementation of the social learning mechanism. Once the new agent is properly initialized, it becomes part of the system (line 13). In line 15, we explicitly update the environment. However, in a real implementation, the environment may be continuously updated as a result of the system's operation.

In most swarm intelligence systems, the population of agents is large and homogeneous, that is, it is composed of agents that follow exactly the same behavioral rules. Thus, any knowledge acquired by an agent is likely to be useful for another one. The social learning mechanism used in an instantiation of the ISL framework should allow the transfer of knowledge from one agent to the other. In some cases, it is possible to have access to the full state of the agent that serves as a "model" to be imitated, and thus, the social learning mechanism is simple. In other cases, access to the model agent's state may be limited and a more sophisticated mechanism is required. In most cases, the result of the social learning mechanism will not be simply a copy of the model agent's state, but a biased initialization toward it. Copying is not always a good idea because what may work very well for an agent in a system composed of $n$ agents may not work well in a system of $n + 1$ agents.

## 4 Case Studies

In this section, I will briefly describe the three case studies that colleagues and I used in order to measure the effectiveness of ISL in the context of optimization. The swarm intelligence algorithms used were the particle swarm optimization (PSO) algorithm [18], the ant colony optimization algorithm for continuous domains ($\text{ACO}_{\mathbb{R}}$) [33], and the artificial bee colony (ABC) algorithm [17].

### 4.1 Case Study 1: Particle Swarm Optimization

**The Basic Algorithm.** In PSO algorithms [18], very simple agents, called *particles*, form a *swarm* and move in an optimization problem's search space. Each particle's position represents a candidate solution to the optimization problem. The position and velocity of the $i$-th particle along the $j$-th coordinate of the problem's search space at iteration $t$ are represented by $x_{i,j}^t$ and $v_{i,j}^t$, respectively. The core of the PSO algorithm is the set of rules that are used to update these two quantities. These rules are:

$$v_{i,j}^{t+1} = wv_{i,j}^t + \text{U}(0, \varphi_1)(p_{i,j}^t - x_{i,j}^t) + \text{U}(0, \varphi_2)(l_{i,j}^t - x_{i,j}^t)\,, \tag{1}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}\,, \tag{2}$$

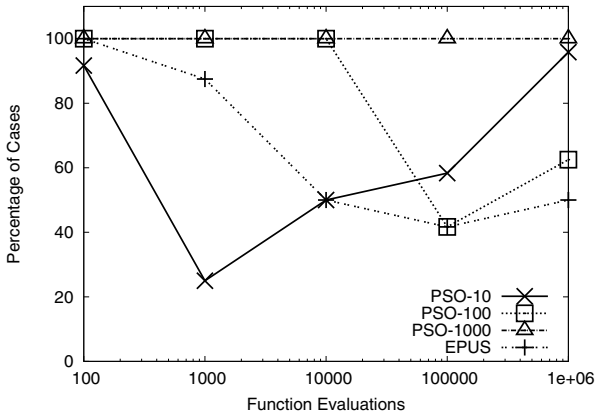
where $w$, $\varphi_1$ and $\varphi_2$ are parameters of the algorithm, $\text{U}(a, b)$ represents a call to a random number generator that returns a uniformly distributed random number in the range $[a, b)$, $p_{i,j}^t$ represents the $j$-th component of the best solution ever visited by the $i$-th particle, and $l_{i,j}^t$ represents the $j$-th component of the best solution ever visited by a subset of the swarm referred to as the $i$-th particle's *neighborhood*. The definition of each particle's neighborhood is usually parametric, fixed, and set before the algorithm is run.

**Integration with ISL.** The ISL framework can be instantiated in different ways in the context of PSO algorithms. Here, I present the most basic variant, which was first described in [26] and benchmarked in [29]. A more sophisticated variant that exhibits a much better performance is presented in [25].

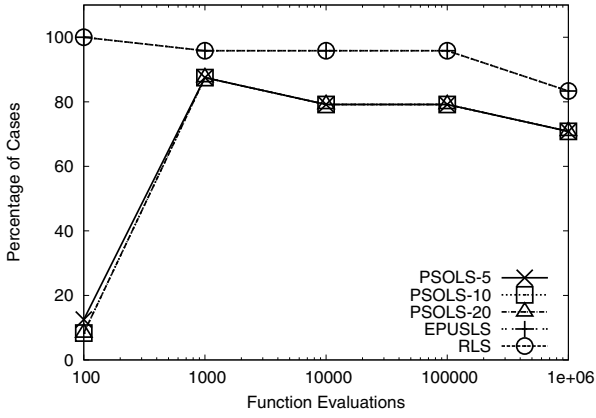
The most basic instantiation of the ISL framework in the context of PSO algorithms is a PSO algorithm with a growing population size called incremental particle swarm optimizer (IPSO). In IPSO, every time a new particle is added, it is initialized using the following rule:

$$x'_{new,j} = x_{new,j} + U(p_{model,j} - x_{new,j}), \tag{3}$$

where $x'_{new,j}$ is the new particle's updated position, $x_{new,j}$ is the new particle's original random position, $p_{model,j}$ is the model particle's previous best position, and $U$ is a uniformly distributed random number in the range $[0, 1)$. This rule moves a new particle from an initial randomly generated position in the problem's search space to one that is closer to the position of a model particle. Once the rule is applied for each dimension, the new particle's previous best position is initialized to the point $\boldsymbol{x}'_{new}$ and its velocity is set to zero. The random number $U$ is the same for all dimensions in order to ensure




a) PSO – No local search




b) IPSOLS – With local search

**Fig. 1.** Percentage of test cases in which the performance of IPSO (a) and IPSOLS (b) is better or no worse (according to a Wilcoxon test at a significance level of 0.05) than the performance of other comparable algorithms

that the new particle's updated previous best position will lie somewhere along the direct attraction vector $\boldsymbol{p}_{model} - \boldsymbol{x}_{new}$. Finally, the new particle's neighborhood, that is, the set of particles from which it will receive information in subsequent iterations, is generated at random using the same parameters used to generate the rest of the particles' neighborhoods.

A better performing variant of IPSO, called IPSOLS, uses a local search procedure. In the context of the ISL framework, a call to a local search procedure may be interpreted as a particle's "individual learning" ability since it allows a particle to improve its solution in the absence of any social influence. In experiments with IPSOLS and other algorithms, we used Powell's conjugate directions set method [32] as local search.

**Results.** A condensed view of the results obtained with IPSO and IPSOLS in [29] is shown in Fig. 1. The figure shows the percentage of test cases (a total of 24 cases: 12 benchmark functions and two neighborhood types) in which the performance of IPSO and IPSOLS is better than or indistinguishable from the performance of reference algorithms (in a statistical sense). The reference algorithms are a PSO algorithm with three different population sizes and a PSO algorithm with a sophisticated mechanism for changing the population size over time (EPUS) [16]. The algorithms used in the comparison with IPSOLS are also a PSO algorithm with local search and three different population sizes, EPUS with local search, and a randomly restarted local search. Details about the experimental setup can be found in [29].

Using IPSO is advantageous when the optimal population size for a particular budget in terms function evaluations is not known in advance. IPSO is advantageous in these cases because a specific population size will produce acceptable results only for runs of a particular length. For example, in our experiments, a PSO algorithm with 10 particles returned good results only for runs of 1000 function evaluations. However, if more time is available, 10 particles return poor results in comparison with larger swarms. In contrast, IPSO has a competitive performance for runs of different length as can be seen in Fig. 1. In any case, the absolute quality of the results obtained with the use of a local search procedure is much better than without local search. Thus, the results in subfigure (b) are more interesting. Here, IPSOLS's performance is clearly better than that of the other algorithms. The reason is that the combination of ISL, PSO and a local search procedure makes particles in IPSOLS move from one local optimum to another [30], producing high quality solutions in a few iterations of the algorithm.

### 4.2 Case Study 2: Ant Colony Optimization

**The Basic Algorithm.** ACO$_{\mathbb{R}}$ [33] maintains a solution archive of size $k$ that is used to keep track of the most promising solutions and their distribution over the search space. Initially, the solution archive is filled with randomly generated solutions. The archive is then updated as follows. At each iteration, $m$ new solutions are generated and from the $k+m$ solutions that become available, only the best $k$ solutions are kept. The mechanism responsible for the generation of new solutions samples values around the solutions $s_i$ with $i \in \{1,\ldots,k\}$ in the archive. This is done on a coordinate-per-coordinate basis using Gaussian kernels defined as sums of weighted Gaussian functions. The Gaussian kernel for coordinate $j$ is

$$G_j(x) = \sum_{i=1}^{k} \omega_i \frac{1}{\sigma_{ij}\sqrt{2\pi}} e^{-\frac{(x-\mu_{ij})^2}{2{\sigma_{ij}}^2}}, \tag{4}$$

where $j \in \{1,\ldots,D\}$ and $D$ is the problem's dimensionality. The mean and variance of these Gaussian functions are set as follows: $\mu_{ij} = s_{ij}$, and

$$\sigma_{ij} = \xi \sum_{r=1}^{k} \frac{|s_{rj} - s_{ij}|}{k-1}, \tag{5}$$

which is the average distance between the $j$-th component of the solution $s_i$ and the $j$-th component of the other solutions in the archive, multiplied by a parameter $\xi$.

The weight $\omega_i$ associated with solution $s_i$ depends on its quality, represented by its ranking in the archive, rank$(i)$ (the best solution is ranked first and the worst solution is ranked last). This weight is calculated using also a Gaussian function:

$$\omega_i = \frac{1}{qk\sqrt{2\pi}} e^{\frac{-(\text{rank}(i)-1)^2}{2q^2k^2}}, \tag{6}$$

where $q$ is a parameter of the algorithm. During the solution generation process, each coordinate is treated independently. For generating the $j$-th component of a new solution, the algorithm chooses first an archive solution with a probability proportional to its weight. Then, the algorithm generates a normally-distributed random number with mean and variance equal to $\mu_{ij}$ and $\sigma_{ij}$ as defined above. This number is the $j$-th component of the new solution. This process repeated $m$ times for each dimension $j \in \{1, ..., D\}$ in order to generate $m$ new candidate solutions.

**Integration with ISL.** The instantiation of the ISL framework with the ACO$_{\mathbb{R}}$ algorithm requires increasing the number of solutions handled per iteration and the biased initialization of new solutions. The resulting algorithm is called IACO$_{\mathbb{R}}$ if no local search is used, and IACO$_{\mathbb{R}}$-LS if it is. These algorithms were first proposed in [20].

In IACO$_{\mathbb{R}}$ the initial size of the solution archive is small. As the optimization process proceeds, new solutions are added to the solution archive at a rate determined by a user-specified criterion. New solutions are initialized using information from a subset of the solutions in the archive (usually the best solution). The rule used to bias the initialization of new solutions is the same as in IPSO (see Eq. 3).

IACO$_{\mathbb{R}}$ differs from the original ACO$_{\mathbb{R}}$ algorithm in the way the solution archive is updated. In IACO$_{\mathbb{R}}$, once a guiding solution is selected, and a new one is generated (in exactly the same way as in ACO$_{\mathbb{R}}$), they are compared. If the newly generated solution is better than the guiding solution, it replaces it in the archive. In contrast, in ACO$_{\mathbb{R}}$ all solutions, new and old, compete at the same time for a slot in the solution archive. Another difference is the mechanism for selecting the guiding solution in the archive. In IACO$_{\mathbb{R}}$, the best solution in the archive is used as guiding solution with probability $p$. With a probability $1 - p$, all the solutions in the archive are used to generate new solutions. Finally, IACO$_{\mathbb{R}}$ is restarted (keeping the best-so-far solution) if the best solution is improved less than a certain threshold for a number of consecutive iterations.

As in the PSO case, the quality of the solutions found with IACO$_{\mathbb{R}}$ typically improve if a local search method is used. In our experiments, we measured the performance of IACO$_{\mathbb{R}}$-LS with Powell's conjugate directions set [32] and Lin-Yu Tseng's mtsls1 [36] methods as local search procedures.

**Results.** To benchmark IACO$_{\mathbb{R}}$-LS, colleagues and I followed the protocol proposed by Lozano *et al.* for a special issue on large-scale optimization in the Soft Computing Journal [22]. We compared the results obtained by IACO$_{\mathbb{R}}$-LS with those obtained with IPSOLS (the version described in [25]) and other 15 algorithms. The results are shown in Fig. 2.

IACO$_{\mathbb{R}}$-LS using mtsls1 as a local search is among the best performing algorithms. In at least eight benchmark functions, IACO$_{\mathbb{R}}$-mtsls1 found an average solution quality at least equal to $10^{-14}$. Some of these functions are the well-known Rosenbrock and Rastrigin functions. These results are thus remarkable considering the fact that these

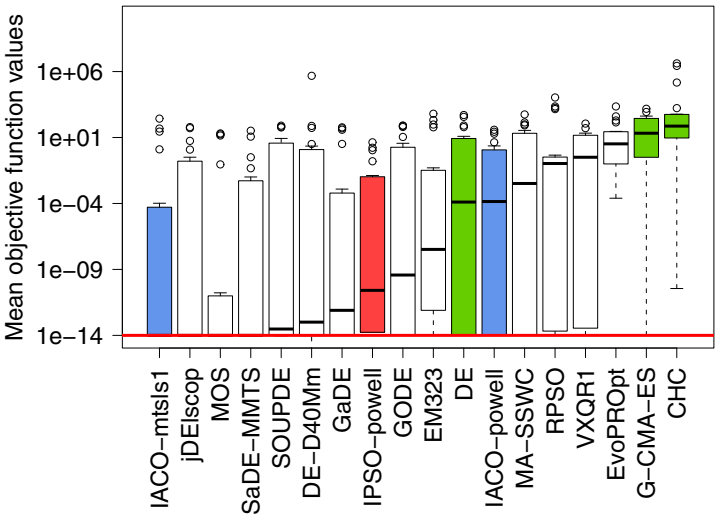



**Fig. 2.** Comparison of IACO$_{\mathbb{R}}$-LS with other algorithms: Distribution of average values (over 25 runs) obtained across 19 functions in 100 dimensions after up to 500,000 function evaluations. Values at or below the threshold $10^{-14}$ are considered "zero" for numerical reasons. DE [34], G-CMA-ES [1], and CHC [8] were proposed by Lozano *et al.* [22] as reference algorithms. The benchmark functions used are described in [15].

functions cannot usually be solved to such a precision level. An interesting result of this comparison comes from the fact that G-CMA-ES [1], which many still consider a state-of-the-art optimization algorithm, is among the worst performing algorithm. This result does not mean that G-CMA-ES is not a good algorithm, but that it does not scale well with the problem's size. Therefore, for large-scale problems, IACO$_{\mathbb{R}}$-mtsls1 can be considered a representative algorithm of the state of the art.

### 4.3 Case Study 3: Artificial Bee Colony Optimization

**The Basic Algorithm.** The design of the artificial bee colony (ABC) algorithm [17] is inspired by the foraging behavior of honeybee swarms, in particular, the recruitment of honeybees to good food sources. The first step of this algorithm is to randomly place a number $SN$ of candidate solutions, called *food sources*, in the problem's search space. The algorithm's goal is to discover better food sources (improve the quality of candidate solutions). This is done as follows: First, simple agents called *employed bees* select uniformly at random a food source and explore another location using the following rule:

$$v_{i,j} = x_{i,j} + \mathrm{U}(-1,1)(x_{i,j} - x_{k,j}),\ i \neq k\,, \tag{7}$$

where $i,\ k \in \{1,2,\ldots,SN\}$, $j \in \{1,2,\ldots,D\}$, $x_{ij}$ and $x_{kj}$ are the position of the reference food source $i$ and a randomly selected food source $k$ in dimension $j$, respectively. The better food source between the new and the reference food sources is kept by the algorithm. The next step is performed by another kind of agent called *onlooker*

*bee*, which looks for better food sources around other food sources based on their quality. This is done by first selecting a reference food source with a probability based on its quality so that better food sources are more attractive. This step is responsible for the intensification behavior of the algorithm since information about good solutions is exploited. The third step is performed by so-called *scout bees*. In this step, a number of food sources that have not been improved for a predetermined number of iterations (controlled by a parameter *limit*), are detected and abandoned. Then, scout bees search for a new food source randomly in the whole search space.

**Integration with ISL.** IABC and IABC-LS were proposed in [2]. From these two algorithms, IABC-LS is the better performing. In ABC-LS, the number of food sources increases over time according to a predefined schedule. Initially, only a few sources are used. New food sources are placed using Eq. 3. Scout bees in IABC-LS use a similar rule when exploring the search space. This rule is

$$x'_{\text{new},j} = x_{\text{best},j} + R_{\text{factor}}(x_{\text{best},j} - x_{\text{new},j}) \,, \tag{8}$$

where $R_{\text{factor}}$ is a parameter that controls how close to the best-so-far food source the new food source will be. IABC-LS also differs from the original ABC algorithm in the way employed bees select the food source around which they explore. In IABC-LS, employed bees search around the best food source instead of around a randomly chosen one in order to enhance the search intensification. IABC-LS is a hybrid algorithm that calls a local search procedure at each iteration. The best-so-far food source location is usually used as the initial solution from which the local search is called. The result

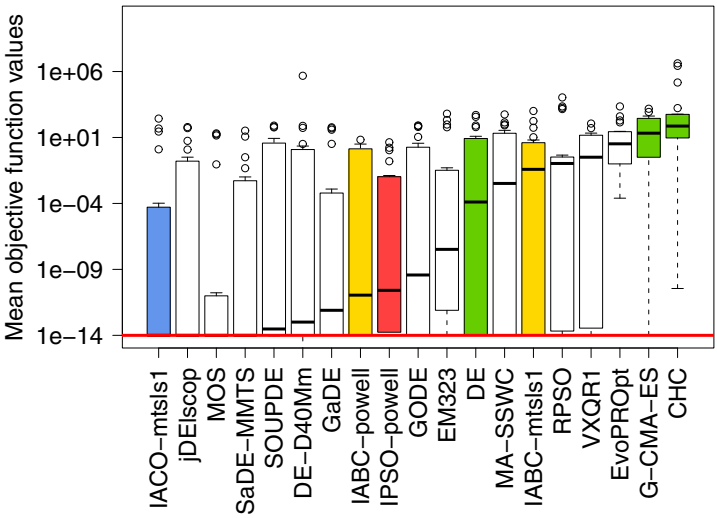



**Fig. 3.** Comparison of IABC-LS with other algorithms: Distribution of average values (over 25 runs) obtained across 19 functions in 100 dimensions after up to 500,000 function evaluations. Values at or below the threshold $10^{-14}$ are considered "zero" for numerical reasons. DE [34], G-CMA-ES [1], and CHC [8] were proposed by Lozano *et al.* [22] as reference algorithms. The benchmark functions used are described in [15].

of the local search replaces the best-so-far solution if there is an improvement on the initial solution. To fight stagnation, the local search procedure may be applied from a randomly chosen solution if the best-so-far solution cannot be improved any further.

**Results.** To measure the performance of IABC-LS, the same protocol used to benchmark IACO$_{\mathbb{R}}$-LS was used. The results are shown in Fig. 3. IABC-LS using Powell's conjugate directions set method as the local search component exhibits practically the same performance as IPSO-LS with the same local search. IABC-LS with mtsls1 as the local search method does not perform as well. These results together with the results obtained with IACO$_{\mathbb{R}}$-LS, suggest that the observed performance does not depend only on the local search method used, but on the interaction between the the incremental algorithm and the local search used. In any case, the instantiation of the ISL framework in the context of ABC algorithms also improves the performance of the original algorithm. ISL transformed an algorithm not known for being state of the art (ABC) into a highly competitive algorithm.

## 5 Conclusions

Engineered swarm intelligence systems are composed of agents that interact with one another and with their environment in order to accomplish a certain task. Usually, these systems are composed of agents that use the same behavioral rules; therefore, these rules must allow agents to engage in positive interactions (those that help the system accomplish the assigned task) and avoid negative interactions (those that block or hinder the agents' task-performing behavior). Typically, it is impossible to predict when any two agents will interact or whether they will do so positively. As a consequence, designers often complexify the behavioral rules of the agents, or the agents' characteristics. Both of these strategies limit the systems' scalability potential and make the design task more challenging.

The incremental social learning (ISL) framework was proposed to reduce the effects of negative interactions in swarm intelligence systems without requiring the complexification of the agents' behavioral rules or characteristics. Three case studies in the context of optimization were carried out in order to assess the effectiveness of the ISL framework. The three algorithms that served this purpose were particle swarm optimization, ant colony optimization for continuous domains ACO$_{\mathbb{R}}$, and artificial bee colony optimization. In each of these cases, the ISL framework improved the performance of the underlying algorithms, a sign of the reduced effect of negative interactions. The instantiation of the ISL framework with ACO$_{\mathbb{R}}$ resulted in a new state-of-the-art optimization algorithm for problems whose dimensionality makes them unsuitable to be dealt with other high performance algorithms such as G-CMA-ES.

**Acknowledgements.** I thank Agostinho Rosa and Joaquim Filipe for inviting me as a keynote speaker to the International Joint Conference on Computational Intelligence (IJCCI–2011), held on October 24–26, 2011 in Paris, France. I carried out the work reported in this paper in collaboration with Tianjun Liao, Doğan Aydın, Ken van den Enden, Thomas Stützle, and Marco Dorigo, while I was a doctoral student at the *Université libre de Bruxelles* in Brussels, Belgium.

## References


1. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005), pp. 1769–1776. IEEE Press, Piscataway (2005)
2. Aydın, D., Liao, T., Montes de Oca, M.A., Stützle, T.: Improving performance via population growth and local search: The case of the artificial bee colony algorithm. In: Proceedings of the International Conference on Artificial Evolution, EA 2011 (2011) (to appear)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press, New York (1999)
4. Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in Biological Systems. Princeton University Press, Princeton (2001)
5. Couzin, I.D.: Collective minds. Nature 445(7129), 715 (2007)
6. Dorigo, M., Birattari, M.: Swarm intelligence. Scholarpedia 2(9), 1462 (2007), http://dx.doi.org/10.4249/scholarpedia.1462
7. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Books. MIT Press, Cambridge (2004)
8. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Whitley, D.L. (ed.) Foundation of Genetic Algorithms 2, pp. 187–202. Morgan Kaufmann, San Mateo (1993)
9. Garnier, S., Gautrais, J., Theraulaz, G.: The biological principles of swarm intelligence. Swarm Intelligence 1(1), 3–31 (2007)
10. Gershenson, C.: Design and control of self-organizing systems. Ph.D. thesis, Vrije Universiteit Brussel, Brussels, Belgium (2007)
11. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the argentine ant. Naturwissenschaften 76(12), 579–581 (1989)
12. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes Sociaux 6(1), 41–80 (1959)
13. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intelligence 1(2), 95–113 (2007)
14. Helbing, D., Vicsek, T.: Optimal self-organization. New Journal of Physics 1, 13.1–13.17 (1999)
15. Herrera, F., Lozano, M., Molina, D.: Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems (2010), http://sci2s.ugr.es/eamhco/updated-functions1-19.pdf (last accessed: July 2010)
16. Hsieh, S.T., Sun, T.Y., Liu, C.C., Tsai, S.J.: Efficient population utilization strategy for particle swarm optimizer. IEEE Transactions on Systems, Man, and Cybernetics 39(2), 444–456 (2009)
17. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39(3), 459–471 (2007)
18. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, Piscataway (1995)
19. Kube, C.R., Bonabeau, E.: Cooperative transport by ants and robots. Robotics and Autonomous Systems 30(1-2), 85–101 (2000)

20. Liao, T., Montes de Oca, M.A., Aydın, D., Stützle, T., Dorigo, M.: An incremental ant colony algorithm with local search for continuous optimization. In: Krasnogor, N., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011), pp. 125–132. ACM Press, New York (2011)
21. Lobo, F.G., Lima, C.F.: Adaptive Population Sizing Schemes in Genetic Algorithms. In: Parameter Setting in Evolutionary Algorithms. SCI, vol. 54, pp. 185–204. Springer, Heidelberg (2007)
22. Lozano, M., Molina, D., Herrera, F.: Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. Soft Computing 15(11), 2085–2087 (2011)
23. Matarić, M.J.: Learning social behavior. Robotics and Autonomous Systems 20(2-4), 191–204 (1997)
24. Montes de Oca, M.A.: Incremental social learning in swarm intelligence systems. Ph.D. thesis, Université Libre de Bruxelles, Brussels, Belgium (2011)
25. Montes de Oca, M.A., Aydın, D., Stützle, T.: An incremental particle swarm for large-scale optimization problems: An example of tuning-in-the-loop (re)design of optimization algorithms. Soft Computing 15(11), 2233–2255 (2011)
26. Montes de Oca, M.A., Stützle, T.: Towards incremental social learning in optimization and multiagent systems. In: Rand, W., et al. (eds.) Workshop on Evolutionary Computation and Multiagent Systems Simulation of the Genetic and Evolutionary Computation Conference (GECCO 2008), pp. 1939–1944. ACM Press, New York (2008)
27. Montes de Oca, M.A., Stützle, T., Birattari, M., Dorigo, M.: Frankenstein's PSO: A composite particle swarm optimization algorithm. IEEE Transactions on Evolutionary Computation 13(5), 1120–1132 (2009)
28. Montes de Oca, M.A., Stützle, T., Birattari, M., Dorigo, M.: Incremental social learning applied to a decentralized decision-making mechanism: Collective learning made faster. In: Gupta, I., Hassas, S., Rolia, J. (eds.) Proceedings of the Fourth IEEE Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010), pp. 243–252. IEEE Computer Society Press, Los Alamitos (2010)
29. Montes de Oca, M.A., Stützle, T., Van den Enden, K., Dorigo, M.: Incremental social learning in particle swarms. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 41(2), 368–384 (2011)
30. Montes de Oca, M.A., Van den Enden, K., Stützle, T.: Incremental Particle Swarm-Guided Local Search for Continuous Optimization. In: Blesa, M.J., Blum, C., Cotta, C., Fernández, A.J., Gallardo, J.E., Roli, A., Sampels, M. (eds.) HM 2008. LNCS, vol. 5296, pp. 72–86. Springer, Heidelberg (2008)
31. Poli, R.: Analysis of the publications on the applications of particle swarm optimisation. Journal of Artificial Evolution and Applications, Article ID 685175, 10 pages (2008)
32. Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7(2), 155–162 (1964)
33. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. European Journal of Operational Research 185(3), 1155–1173 (2008)
34. Storn, R.M., Price, K.V.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
35. Stützle, T., López-Ibáñez, M., Dorigo, M.: A concise overview of applications of ant colony optimization. In: Cochran, J.J., et al. (eds.) Wiley Encyclopedia of Operations Research and Management Science, vol. 2, pp. 896–911. John Wiley & Sons, Ltd., New York (2011)
36. Tseng, L., Chen, C.: Multiple trajectory search for large scale global optimization. In: Proceeding of the IEEE 2008 Congress on Evolutionary Computation (CEC 2008), pp. 3052–3059. IEEE Press, Piscataway (2008)

# Part I
# Evolutionary Computation Theory and Applications

# Solution of a Modified Balanced Academic Curriculum Problem Using Evolutionary Strategies

Lorna V. Rosas-Tellez[1], Vittorio Zanella-Palacios[1], and Jose L. Martínez-Flores[2]

[1] Universidad Popular Autónoma del Estado de Puebla,
Information Technologies Department,
13 Poniente 1927 Col. Santiago, Puebla Pue. México
[2] Universidad Popular Autónoma del Estado de Puebla,
Interdisciplinary Center for Postgraduate Studies, Research, and Consulting,
21 sur 1103 Col. Santiago, Puebla Pue. México
{lornaveronica.rosas,vittorio.zanella,
joseluis.martinez01}@upaep.mx

**Abstract.** The Balanced Academic Curriculum Problem (BACP) is a constraint satisfaction problem classified as NP- Hard, this problem consists in the allocation of courses in the periods that are part of a curriculum such that the prerequisites are satisfied and the load of courses is balanced for the students. In this paper is presented the solution for a modified BACP where the loads may be the same or different for each one of the periods and is allowed to have some courses in a specific period. This problem is modeled as an integer programming problem and is proposed the use of evolutionary strategies for its solution because was not possible to find solutions for all the instances of this modified problem with formal methods.

**Keywords:** Optimization, Evolutionary strategies, Balanced academic curriculum problem.

## 1 Introduction

A curriculum is formed by a set of courses and these courses have assigned a number of credits that represent the effort in hours per week that the student requires to follow the courses successfully. For parents or tutors and for the institution represents the economic cost of this course. The academic load is the sum of the credits of all the courses in a given period.

Therefore, the correct planning of the curriculum results in benefit of the all the involved: For the institutions, favors the departmentalization and the resulting cost savings, for the students, one good load distribution represents the academic effort that they require invest, for the parents or tutors, a good distribution of the credits allow planning financial efforts.

Balanced Academic Curriculum Problem (BACP) consists in the allocation of courses in the periods that are part of a curriculum such that the prerequisites are satisfied and the credits load is balanced. The BACP belongs to the class of problems

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 49–58.
DOI: 10.1007/978-3-642-35638-4_4 

CSP (Constraint Satisfaction Problems), and this is a decisional optimization problem classified as NP-Hard [1].

The BACP problem was introduced by Castro and Manzano [2] with three test cases called BACP8, BACP10 and BACP12 included in CSPLib [3] and these have been used to test models proposed by other researchers.

The model proposed in [2] uses the following integer programming model:

**Parameters**

$m$ : Number of courses
$n$ : Number of periods
$\alpha_i$ : Number of credits of course $i$; $\forall i = 1..m$
$\beta$ : Minimum academic load allowed per period
$\gamma$ : Maximum academic load allowed per period
$\delta$ : Minimum amount of courses per period
$\varepsilon$ : Maximum amount of courses per period

**Decision Variables**

$$x_{ij} = \begin{cases} 1 \text{ if course } i \text{ is assigned to period } j \\ 0 \text{ otherwise} \end{cases}$$

$c_i$ : academic load of period $j$, $\forall j = 1, ..., n$

$$c_j = \sum_{i=1}^{m} \alpha_i * x_{ij} \qquad \forall j = 1..n \tag{1}$$

**Objective Function**

$$Min \;\; c = Max\{c_1, \ldots, c_n\} \tag{2}$$

**Constraints**

If the course $b$ has the course $a$ as prerequisite then: $x_a < x_b$

$$\beta \le c_k \le \gamma \quad \forall j = 1\ldots., n \tag{3}$$

$$\delta \le \sum_{k=1}^{m} x_{ij} \le \varepsilon \quad \forall j = 1, \ldots, n \tag{4}$$

Recent works have tried to solve this problem using genetic algorithms and constraint propagation [4], local search techniques [5], formal methods (HyperLingo) for the integer programming problems [6] and multiple optimization, using genetic algorithm of local search [7]. All these approach have found the optimal for the three test cases included in CSPLib and in some cases also for the curriculums of their universities.

In [6] was proposed a modified BACP problem where are considered constraints of academic load and total of courses within a specific range per period, i.e., not necessarily all periods will have the same ranges for their academic loads and number of courses; also add the restriction of to locate a course in a given period. This problem was modeled as an integer programming problem and is reported to find optimum

solutions using a formal method for some of its instances but not for all, and the solutions for the three instances included in CSPLib.

In this paper is solved the modified BACP using evolutionary strategies to find solutions to the instances that the formal method could not to solve.

## 2 Formulation for Model BACP Modified

In the model of interest proposed in [6] is considered to modify two constraints of the base formulation, the first one is to make flexible the course load per period and the second one is to make flexible the number of courses per period, i.e., that we can place different limits on course load and number of courses for each period. It also adds a restriction which allows the location of some of the courses in a specific period.

**Parameters**

$Nta$ : Number of courses
$Ntp$ : Number of academic periods
$crd_i$ : Number of course credits i=1..$Nta$
$mca_j$ : Minimum academic load allowed per period
$Mca_j$ : Maximum academic load allowed per period
$mna_j$ : Minimum number of courses per period
$Mna_j$ : Maximum number of courses per period
$c$ : Course it is desirable to locate between certain periods.
$mpcc$ : Minimum period of location of the course
$Mpcc$ : Maximum period of location of the course
$C_j$ : Academic load

$$C_j = \sum_{i=1}^{Nta} crd_i * x_{ij} \qquad \forall j = 1..Ntp \tag{5}$$

**Decision Variables**

$C_j$ : Academic load for the period $j$ =1..$Ntp$
$Cmx$ : Maximum course load

$$x_{ij} = \begin{cases} 1 \text{ if course } i \text{ is assigned to period } j \\ 0 \text{ otherwise} \end{cases}$$

**Objective Function**

$$f_{objective} = Min\{Cmx\} \tag{6}$$

where $Cmx = Max\ \{ c_1, c_2, \ldots, c_{Ntp} \}$

**Constraints**

The load of the period $j$ must be within the allowable range.

$$mca_i \leq C_j \leq Mca_j \qquad \forall j = 1..Ntp \tag{7}$$

The number of courses of the period must be within the allowable range.

$$mna_j \leq \sum_{i=1}^{Nta} x_{ij} \leq Mna_j \qquad \forall j = 1..Ntp \tag{8}$$

If the course *b* has the course *a* as prerequisite then

$$x_{bj} \leq \sum_{r=1}^{j-1} x_{ar} \qquad \forall j = 2..Ntp \tag{9}$$

Convenient location for the course *c*

$$\sum_{j=mpv_c}^{Mpc_c} x_{cj} = 1 \tag{10}$$

## 3 Evolutionary Strategies

Evolutionary Strategies are optimization algorithms based on Darwin's theory of evolution, which states that only those individuals best adapted to their environment survive and reproduce. The procedure starts choosing a number of possible solutions in the search space to generate a random initial population, after that, each possible solution is evaluated based on a fitness function and through of selection operation are selected individuals for carry out the genetic operations crossover and mutation, with the idea that new promising individuals will be evolved from their ancestors to produce an improved population. Crossover is the combination of information from two or more individuals and mutation is the alteration of the information of a single individual [8]. In the Evolutionary Strategies the mutation is the most important operation. There are several types of evolutionary strategies depending on the size of the population and how the individuals are replaced in the population prior to generating the new population. In our case we use an evolutionary strategy ES-(1+3), i.e., there is an initial population of a single individual and from this individual will generate 3 new individuals by mutation. Of these 4 individuals the best is choosing for the next population.

Evolutionary strategies were used, at least initially, to optimization problems of real functions, but are possible to use it successfully in other domains. In this paper we use evolutionary strategies in populations where individuals are vectors.

One element of the population is represented by a vector, where the position indicates the course and the content of each position indicates the period to which it was assigned, as shown in figure 1.

| 0 | 1 | 2 | 3 | 4 | 5 | | 59 | 60 | 61 | Course |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 1 | 2 | 2 | | 9 | 9 | 0 | Period |

**Fig. 1.** Element of the population

In our case we used a population with a single individual so that the only operation performed is mutation, which consists in changing the period of a course of the curriculum that meets the prerequisites and restrictions of preference period, as shown in figure 2, where the course 4 is changing to period 8.

| 0 | 1 | 2 | 3 | 4 | 5 | | 59 | 60 | 61 | Course |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 1 | **8** | 2 | | 9 | 9 | 0 | Period |

**Fig. 2.** Element of the population

We can consider that a balanced curriculum should have a uniform distribution of all the credits that make up the curriculum, so the fitness function used is the sum of the absolute error, which is calculated using the following formula.

$$Fitness(h) = \sum_{k=1}^{Npt} \left| C_k - P \right| \tag{11}$$

Where $C_k$ is the academic load of the period $k$ calculated with the formula (4) and $P$ is the average number of credits per period

$$P = \sum_{i=1}^{Ntp} \frac{C_i}{Ntp} \tag{12}$$

The initial population consists of the curriculum that we want to balance, this is a feasible solution.

Once that we have the first element of the population three new elements are generated through mutation. As the mutation is the random change of the value of a single element within the vector, randomly are chosen a course to be changed and the period where it will change.

Given the course and the period, are validated the restrictions of prerequisites, load, course and period preference, if they are satisfied, the change is made, otherwise are selected randomly another course and period and redo the validation. This continues until to find the pair course - period that meets with the restrictions. With this process will generate 3 new individuals from the individual in the present population, the four individuals are evaluated by the fitness function (formula 10) and the best is selected for the next generation, this process continue until the criteria of finalization are reached.

In the moment that is detected that a local optimum has been reached a change in the process of mutation is made. Now the mutation will change two elements of the vector, that is now going to get the periods with more load and less load and will try to exchange two courses randomly between these two periods.

Having the two courses which will be exchanged, are evaluated the restrictions of prerequisite , load, course and period preference, if the exchange can be given a new individual is generated in otherwise the mutation is not done, the minimum period is marked as ineligible for the next selection and is cleared until that an improvement occurs.

### 3.1 Algorithm

```
Evolutionary_Strategie(λ, μ, α)
/*
λ = size of the population (λ=1),
μ = number of new individuals generated with mutation
    (μ=3),
α = mutation factor
*/
Begin
  i0=Generation_of_initial_population(λ)
  cont=0;
  While (optimum ≠ i0)
    Begin
      If cont<=TOL /*TOL= maximum number of iterations in
                   a local optimum*/
      Then α=1; /*select for mutation only 1 curse
      Else α=2; /*select for mutation 2 courses
      For i=1 to μ
        Begin
          Individual[i]=mutation(i0);
          Evaluation[i]=fitness(individual[i]);
        End
      K=Select_the_best_individual();
      If evaluation[K]> evaluation[i0]
      Then
        i0 = individual[K];
        cont=0;
      Else cont=cont+1;
    End
End
```

## 4 Results

The tests were carried out for the three base cases included in CSPLib and the cases proposed by [6] for which no solution could be found.

### 4.1 Base Cases

The base cases included in CSPLib are: BACP8, BACP10 and BACP12, whose features are shown in tables 1 and 2.

Table 3 shows the results obtained with the proposed algorithm; in all cases the optimum was reached.

**Table 1.** General features of curriculums

| Code | BACP8 | BACP10 | BACP12 |
|---|---|---|---|
| # Total Courses | 46 | 42 | 66 |
| # Total credits | 133 | 134 | 204 |
| #Total Academic period | 8 | 10 | 12 |
| #Relation Prerequisite | 33 | 34 | 65 |

**Table 2.** Additional features of the curriculums

| Code | BACP8 | BACP10 | BACP12 |
|---|---|---|---|
| Min. Courses /period | 2 | 2 | 2 |
| Max. Courses / period | 10 | 10 | 10 |
| Min Load/ period | 10 | 10 | 10 |
| Max Load/ period | 24 | 24 | 24 |
| #Courses with location | 0 | 0 | 0 |

**Table 3.** Results summary

| Code | Optimum | Average Iterations | Average time (seg.) |
|---|---|---|---|
| **BACP 8** | 17 | 57.6 | 4.5 |
| **BACP 10** | 14 | 87.7 | 4.7 |
| **BACP 12** | 17 | 162.0 | 4.5 |

The academic load per period obtained by the algorithm is shown in table 4.

**Table 4.** Solution found for BACP 8

| Period | Load | Courses |
|---|---|---|
| 1 | 17 | 7 |
| 2 | 17 | 5 |
| 3 | 17 | 5 |
| 4 | 17 | 6 |
| 5 | 17 | 6 |
| 6 | 17 | 6 |
| 7 | 15 | 5 |
| 8 | 16 | 6 |

### 4.2 Proposed Cases

The cases not included in library CSPLib used to test this algorithm are taken from [6], the first is one for which could not always find the optimal and the second is where the optimum never was found. The features of these two problems are shown in tables 5 and 6.

**Table 5.** General features of curriculums

| **Code** | Ici-06 | Ind-06 |
|---|---|---|
| # Total Courses | 61 | 61 |
| # Total credits | 488 | 376 |
| #Total Academic period | 9 | 9 |
| #Relation Prerequisite | 48 | 47 |

**Table 6.** Additional features of the curriculums

| Code | Ici-06 | Ind-06 |
|---|---|---|
| **Min. Courses /period** | 5 | 4, 4, 4, 4, 4, 4, 4, 4, 2 |
| **Max. Courses/ period** | 8 | 9, 9, 9, 9, 9, 9, 9, 9, 4 |
| **Min Load/ period** | 20 | 20, 20, 20, 20, 20, 20, 20, 20, 15 |
| **Max Load/ period** | 60 | 60, 60, 60, 60, 60, 60, 60, 60, 40 |
| **#Courses with location** | 15 | 21 |

In tables 7 and 8 is showing the courses that have preference of location in each of the curriculums, Ici-06 and Ind-06 respectively.

**Table 7.** Preference of location Ici-06

| **Course Code** | **Minimum Period** | **Maximum Period** |
|---|---|---|
| C07001 | 7 | 9 |
| C07002 | 7 | 9 |
| C07003 | 7 | 9 |
| CIV200 | 1 | 2 |
| CIV400 | 6 | 9 |
| CIV401 | 8 | 9 |
| CIV403 | 6 | 9 |
| MAT005 | 1 | 5 |
| MAT006 | 1 | 5 |
| MAT008 | 1 | 5 |
| MAT009 | 1 | 5 |
| OI103101 | 1 | 4 |
| OI103102 | 1 | 4 |
| OI103103 | 1 | 4 |
| OI103104 | 1 | 4 |

**Table 8.** Preference of location Ind-06

| Course Code | Minimum Period | Maximum Period |
|---|---|---|
| C12001 | 7 | 9 |
| C12002 | 7 | 9 |
| C12003 | 7 | 9 |
| C12004 | 8 | 9 |
| FHU001 | 1 | 6 |
| FHU002 | 1 | 6 |
| FHU003 | 1 | 6 |
| IND100 | 1 | 2 |
| IND208 | 4 | 6 |
| IND212 | 4 | 6 |
| IND214 | 6 | 8 |
| IND400 | 7 | 9 |
| LPCI | 1 | 6 |
| LPCII | 1 | 6 |
| OH25001 | 1 | 6 |
| OI103101 | 1 | 6 |
| OI103102 | 1 | 6 |
| OI103103 | 1 | 6 |
| OI103104 | 1 | 6 |
| SSC001 | 5 | 9 |
| SSP002 | 5 | 9 |

Table 9 shows the results obtained with the algorithm; in all cases the optimum was reached.

**Table 9.** Results summary

| Code | Optimum | Average Iterations | Average time (min.) |
|---|---|---|---|
| **Ici-06** | 55 | 57.6 | 3.6 |
| **Ind-06** | 44 | 87.7 | 1.7 |

The academic load per period obtained by the algorithm is shown in Table 10.

**Table 10.** Solution found for Ici-06

| Period | Load | Courses |
|---|---|---|
| 1 | 54 | 7 |
| 2 | 54 | 6 |
| 3 | 54 | 6 |
| 4 | 54 | 7 |
| 5 | 55 | 6 |
| 6 | 55 | 6 |
| 7 | 54 | 8 |
| 8 | 54 | 7 |
| 9 | 54 | 8 |

## 5 Conclusions

In this paper we present the solution, using evolutionary strategies, for a modified Balanced Academic Curriculum Problem, where the load for each period can be equal or different and is allowed to have some courses in a specific period. In some previous works is showed that is possible to find solutions with HyperLingo for some of the instances of the problem, but it is not possible for all of them. However by the results obtained we can see that the use of evolutionary strategies helps us to find the solutions to the problems that could not be resolved with the formal method.

## References

1. Salazar, J.: Programación Matemática, Diaz de Santos, Madrid (2001)
2. Castro, C., Manzano, S.: Variable and value ordering when solving balanced academic curriculum problem. In: Proceedings of the ERCIM Working Group on Constraints (2001)
3. CSPLib: A problem library for constraints, http://www.csplib.org/
4. Lambert, T., Castro, C., Monfroy, E., Saubion, F.: Solving the Balanced Academic Curriculum Problem with an Hybridization of Genetic Algorithm and Constraint Propagation. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 410–419. Springer, Heidelberg (2006)
5. Di Gaspero, L., Schaerf, A.: Hybrid Local Search Techniques for the Generalized Balanced Academic Curriculum Problem. In: Blesa, M.J., Blum, C., Cotta, C., Fernández, A.J., Gallardo, J.E., Roli, A., Sampels, M. (eds.) HM 2008. LNCS, vol. 5296, pp. 146–157. Springer, Heidelberg (2008)
6. Aguilar-Solís, J.A.: Un modelo basado en optimización para balancear planes de estudio en Instituciones de Educación Superior. PhD Thesis. UPAEP, Puebla (2008)
7. Castro, C., Crawford, B., Monfroy, E.: A Genetic Local Search Algorithm for the Multiple Optimisation of the Balanced Academic Curriculum Problem. In: Shi, Y., Wang, S., Peng, Y., Li, J., Zeng, Y. (eds.) MCDM 2009. CCIS, vol. 35, pp. 824–832. Springer, Heidelberg (2009)
8. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin (1999)

# Solving the CVRP Problem Using a Hybrid PSO Approach

Yucheng Kao* and Mei Chen

Department of Information Management, Tatung University, Taipei, Taiwan
ykao@ttu.edu.tw

**Abstract.** The goal of the capacitated vehicle routing problem (CVRP) is to minimize the total distance of vehicle routes under the constraints of vehicles' capacity. CVRP is classified as NP-hard problems and a number of meta-heuristic approaches have been proposed to solve the problem. This paper aims to develop a hybrid algorithm combining a discrete Particle Swarm Optimization (PSO) with Simulated Annealing (SA) to solve CVRPs. The two-stage approach of CVRP (cluster first and route second) has been adopted in the algorithm. To save computation time, a short solution representation has been adopted. The computational results demonstrate that our hybrid algorithm can effectively solve CVRPs within reasonable time.

**Keywords:** Vehicle routing problem, Particle swarm optimization, Simulated annealing.

## 1 Introduction

The vehicle routing problem (VRP) is one of important research subjects in the field of logistics management. It is an interesting research topic and belongs to a category of combinatorial optimization problems. Since Dantzig [1] first proposed the truck dispatching problem, many variants of VRPs have been presented. Jozefowiez et al. [2] presented detailed classifications and comparisons for VRP problems. Classical VRP problems can be classified into two categories: capacitated vehicle routing problems (CVRP) and vehicle routing problem with time windows (VRPTW). The objective of CVRP is to find a set of routes with a minimum total distance traveled to deliver goods for all customers having different demands. The constraints of CVRPs include: every customer is to be served exactly once by a vehicle, each vehicle starts and ends at the same depot, every vehicle has a limited capacity, etc.

To solve CVRPs, the two-stage approach (cluster-first-route-second) is often used. In the first stage we assign each customer into a vehicle, while in second stage we arrange the visiting order of each vehicle which has customers assigned. Accordingly, the customer clustering result may affect the routing result, and the routing result determines the objective function values. On the other hand, the solution approaches can be classified into two categories: exact and heuristic approaches [3]. Branch-and-bound and branch-and-cut algorithms are exact algorithms, whereas route

* Corresponding author.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 59–67.
DOI: 10.1007/978-3-642-35638-4_5 

construction heuristics and two-phase heuristics are classical heuristic algorithms. Because the CVRP is classified as NP-hard problems (or combinatorial optimization problems), meta-heuristic approaches attract many researchers' attention and have been applied to the CVRP in recent decades. The performance of meta-heuristics is often better than classical heuristics. Popular meta-heuristics include Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Tabu Search (TS), Simulated Annealing (SA), Ant colony systems (ACS), Scatter Search (SS), etc. One of important advantages of using meta-heuristic approaches is that we can obtain optimal or near optimal solutions in a reasonable computation time, compared to exact algorithms.

Baker et al. [4] proposed a simple GA for solving CVRPs. The length of solution string is equal to the total number of customers. Each gene is an integer number which ranges from 1 to the total number of vehicles. The algorithm selects parent solutions by using a binary tournament method, produces offspring solutions by using two-point crossover operation, mutates solutions by swapping two randomly selected genes, and selects better solutions for next generation with a ranking replacement method. The GA-based approach was applied in the customer clustering stage. The algorithm also uses some local searches to improve the quality of solutions. The main contribution of this paper is to propose a method of generating structured initial solutions. Using this initialization method, the algorithm has fast convergence but loses solution diversities.

Bell et al. [5] proposed an ACO algorithm for solving the CVRP. The authors proposed a multiple-ant-colony method for solving large size problems (more than 100 customers). They also proposed two methods to improve the performance of their algorithm, including local exchange and candidate list. Zhang et al. [6] proposed an algorithm which integrates scatter search with ACO for solving CVRP. The paper uses scatter search as the main framework and applies ACO to route solution construction.

Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart in 1995 [7]. PSO has been applied to solve many optimization problems, including the CVRP. Chen et al. [8] first proposed a discrete PSO approach to solve the CVRP. Their approach follows the two-stage approach. They used DPSO to perform the task of customer clustering and utilized SA to determine the visiting order of each vehicle. Due to a long solution string (the length equal to the product of total customer number and total vehicle number) their algorithm often needs a larger amount of CPU time to find optimal solutions. Ai et al. [9] also proposed an algorithm based on PSO for CVRPs. In their approach, the solution string of a particle contains coordinate points and coverage radiuses of vehicles. Vehicle routes are constructed according to these cluster-center points and radiuses. The order of visiting customers of each route is found by using an insertion heuristic. Thus their paper also followed the cluster-first-route-second approach to solve CVRPs. Marinakis et al. [10] presented a hybrid PSO algorithm for solving CVRPs. Their algorithm consists of PSO, MPNS-GRASP (multiple phase neighborhood search-greedy randomized adaptive search procedure), Expanding Neighborhood Strategy and path relinking strategy.

This paper proposes a new hybrid PSO with SA to solve the CVRP. Similarly, it follows the cluster-first-route-second approach: use a new discrete PSO to find customer clustering results and then use simulated annealing to arrange the orders of visiting customers. To save computation time a short solution string structure has been adopted. Experimental results show that the proposed algorithm can solve the CVRPs efficiently.

## 2 PSO and SA

PSO is a population-based evolution algorithm and is one of swarm intelligence techniques. It was inspired by the foraging behavior of birds. The movement of a particle in the solution space is guided by its current position, its personal best position and the global best position of the swarm. The main advantage of PSO is fast convergence; however, PSO does not guarantee to find an optimal solution. The classical model of PSO proposed by Shi and Eberhart [11] is defined as follows:

$$V_{id} = W \times V_{id} + C_1 \times Rand \times (P_{best} - X_{id}) + C_2 \times Rand \times (G_{best} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

This paper applies combinatorial particle swarm optimization (CPSO) to solve CVRPs. CPSO was proposed by Jarboui et al. [12] and aims to solve discrete combinatorial optimization problems. CPSO makes use of the evolution framework of PSO and alternatively transforms particle solutions into continuous solutions or discrete solutions in order to find new solutions. CPSO not only retains the characteristics of PSO but also expands the searching ability of PSO from a continuous space to a discrete space.

SA was first proposed by Kirkpatrick in 1983 [13]. It was applied to solving a variety of combinatorial optimization problems through a series of local searches. To avoid being trapped in local optima, SA occasionally accepts worse solutions with an acceptance probability. Temperature is an important parameter of SA. For a certain temperature, SA performs local searches a couple of times. The acceptance probability is changed according to the amount of solution improvement and the current temperature. The temperature is decreased when the iteration number is increased. In fact, the cooling rate controls how fast the temperature drops.

## 3 CPSO-SA Algorithm

This paper proposes a new algorithm combining CPSO and SA, referred to CPSO-SA. It follows the two-stage (cluster-first-route-second) approach for solving CVRP. CPSO is used to deal with customers clustering while SA is applied to arrange the sequence of visiting customers. At the end of each iteration, the algorithm conducts local searches on the top three best particles. Personal best solutions of each particle and the global best solutions of the swarm are used to generate new particles for the next iteration. In this paper we adopt a similar two-stage approach used in Chen et al. [8], but we try to improve their approach by using a short solution string and a more efficient discrete PSO algorithm. Using CPSO-SA, we do not need to check solution feasibility to prevent the case that a customer is assigned to more than one vehicle. As a result, the proposed algorithm does save computational time.

### 3.1 Mathematical Model

The CVRP considered in this paper has a symmetric network. The objective is to minimize the total distance traveled by all vehicles. The following constraints are

considered: each customer is served exactly once by only one vehicle; each vehicle starts and ends its route at the depot; the total demand delivered for every route must not exceed the capacity of the vehicle. The problem can be formulated as follows:

Notations

0: depot;

$n$: number of customers;

$N$: customer set，$N = \{1, 2, \ldots, n\}$;

$v$: number of vehicles;

$V$: vehicle set, $V = \{1, 2, \ldots, v\}$;

$d_{ij}$: distance between customer $i$ and $j$, $d_{ij} = d_{ji}$, $\forall\, i, j \in N \cup \{0\}$;

$q_i$: demand for customer $i$, $q_0 = 0$;

$Q$: maximum capacity for every vehicle;

$X^k_{ij}$: edge $i$-$j$ is served by vehicle $k$ or not. 0: no, 1: yes;

$R_k$: customer set served by vehicle $k$, $R_k = \{j_{k,1}, j_{k,2}, \ldots, j_{k,\,|Rk|}\}$;

$|R_k|$: cardinality of $R_k$;

$j_{k,m}$: customer that is served by vehicle k in $m$th visiting order.

Objective function

$$\text{Minimize } f = \sum_{k=1}^{v}\sum_{i=0}^{n}\sum_{j=0}^{n} d_{ij} X^k_{ij} \tag{3}$$

Subject to

$$\sum_{k=1}^{v}\sum_{i=0}^{n} X^k_{ij} = 1,\ j = 1, 2, \ldots, n \tag{4}$$

$$\sum_{k=1}^{v}\sum_{j=0}^{n} X^k_{ij} = 1,\ i = 1, 2, \ldots, n \tag{5}$$

$$\sum_{i=0}^{n} X^k_{iu} - \sum_{j=0}^{n} X^k_{uj} = 0,\ k = 1, 2, \ldots, v\ ;\ u = 1, 2, \ldots, n \tag{6}$$

$$\sum_{j=1}^{n}\sum_{i=0}^{n} (q_j \times X^k_{ij}) \le Q\ ,\ k = 1, 2, \ldots, v \tag{7}$$

$$\sum_{j=1}^{n} X^k_{0j} \le 1,\ k = 1, 2, \ldots, v \tag{8}$$

$$\sum_{i=1}^{n} X^k_{i0} \le 1,\ k = 1, 2, \ldots, v \tag{9}$$

Eqs. (4) and (5) ensure that each customer must be served by a vehicle exactly once. Eq. (6) considers that the continuity for every vehicle can be maintained. Eq. (7) means that the total customer demand of a vehicle is not allowed to exceed its maximum capacity. Eqs. (8) and (9) mean that every vehicle can be used once or not be used.

### 3.2 Solution Representation

PSO-SA adopts the cluster-first-route-second approach to solve CVRPs. The solution representation for customer clustering is shown in Fig.1, and the one for finding customer sequences is shown in Fig. 2. An example of solution string for the first stage is also presented in Fig. 1. It tells us that customer 1 is assigned to vehicle 2, customer 2 is assigned to vehicle 1, and so on. An example of solution string for the second stage is also shown in Fig. 2. In the example, $|R_1| = 3$ means that the first vehicle has to serve three customers, and $j_{1,1} = 2$ means that the second customer will be visited first by the first vehicle. The hypothetical solution string indicates that three vehicle routes are 0→2→4→8→0, 0→1→5→9→0, and 0→3→6→7→0, respectively.

In the first stage, some PSO vectors have to be defined: $XV_p = (xv_{p1}, xv_{p2}, \ldots, xv_{pn})$ represents the discrete solution of particle $p$, $V_p = (v_{p1}, v_{p2}, \ldots, v_{pn})$ indicates the velocity vector of particle $p$; $P_p = (P_{p1}, P_{p2}, \ldots, P_{pn})$ indicates the personal best solution ever found of particle $p$; $G = (G_1, G_2, \ldots, G_n)$ indicates the global best-so-far solution of the swarm.

| Customer no. (cno) | $c_1$ | $c_2$ | $c_3$ | … | … | … | … | … | $c_n$ |
|---|---|---|---|---|---|---|---|---|---|
| Vehicle no ( $XV$) | $xv_1$ | $xv_2$ | $xv_3$ | … | … | … | … | … | $xv_n$ |

| Customer no.(cno) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Vehicle no ( $XV$) | 2 | 1 | 3 | 1 | 2 | 3 | 3 | 1 | 2 |

**Fig. 1.** Cluster-solution representation used in the first stage

| 0 | $j_{1,1}$ | … | $j_{1,\lvert R_1\rvert}$ | 0 | $j_{2,1}$ | … | $j_{2,\lvert R_2\rvert}$ | 0 | … | 0 | $j_{v,1}$ | … | $j_{v,\lvert R_v\rvert}$ | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| 0 | 2 | 4 | 8 | 0 | 1 | 5 | 9 | 0 | 3 | 6 | 7 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Fig. 2.** Tour-solution representation used in the second stage

### 3.3 CPSO-SA Algorithm

The proposed algorithm performs the evolution process iteratively until the iteration number reaches the maximum iteration number. The evolution process includes two stages, customer clustering and sequencing. Customer clustering is carried out by using CPSO [12] while the visiting order is determined by using SA. Then the algorithm computes the objective function value for every particle solution according to Eq. (3). After that, CPSO-SA performs a local search on top three best particles for solution improvement. The procedure of local search, with considering capacity constraints, is to choose two routes randomly, to select a customer from each of selected routes, and then to exchange their vehicle numbers. After that, SA is used again to find new visiting orders for these two routes. If the new solution is better, replace the current solution with it. Before iteration ends, particles update their Pbest and Gbest solutions respectively. We briefly introduce these two stages as follows.

### 3.3.1 CPSO for Customers Clustering

The first stage contains three phases: transition phase 1, flying phase and transition phase 2. In these phases, a new customer clustering solution can be found by using Eqs. (10) ~ (16). In the first phase, discrete solution vectors (*XV*s) become dummy solution vectors (*YV*s) after the first solution transition from the discrete space to the continuous space. After flying, dummy solution vectors (*YV*s) become continuous solution vectors ($\Lambda V$s). Using a threshold value $\alpha$, the algorithm transits all particle solutions from the continuous space back to the discrete space with their new discrete solutions.

Notations:

*Iter*: current iteration number;

*p*: particle number;

*d*: dimension number;

$y_{pd}^{iter}$ : dummy variable in dimension *d* of particle *p* at iteration *iter*;

$G_d^{iter}$ : best value in dimension *d* for all solutions up to iteration *iter*;

$P_{pd}^{iter}$ : best value in dimension *d* for particle solution *p* up to iteration *iter*;

$xv_{pd}^{iter}$ : current value in dimension *d* of particle solution *p* at iteration *iter*;

*c1*, *c2*: parameters representing the importance of the global best solution and particle's best solution separately;

$v_{pd}^{iter}$ : velocity in dimension *d* of particle *p* at iteration *iter*;

*r1*, *r2*: random numbers range from 0 to 1;

$\lambda_{pd}^{iter}$ : continuous value in dimension *d* for particle solution *p* at iteration *iter*;

$\alpha$: threshold value.

**Transition Phase 1**

$$y_{pd}^{iter} = \begin{cases} 1 & \text{if } (xv_{pd}^{iter-1} = G_d^{iter-1}) \\ -1 & \text{if } (xv_{\text{pd}}^{\text{iter-1}} = P_{pd}^{iter-1}) \\ -1 \text{ or } 1 & \text{randomly, if } (xv_{\text{pd}}^{\text{iter-1}} = P_{pd}^{iter-1} = G_d^{iter-1}) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

**Flying Phase**

$$d_1 = -1 - y_{pd}^{iter} \text{ , distance between } xv_{pd}^{iter-1} \text{ and } P_{pd}^{iter-1} \tag{11}$$

$$d_2 = 1 - y_{pd}^{iter} \text{ , distance between } xv_{pd}^{iter-1} \text{ and } G_d^{iter-1} \tag{12}$$

$$v_{pd}^{iter} = w \cdot v_{pd}^{iter-1} + c_1 \cdot r_1 \cdot d_1 + c_2 \cdot r_2 \cdot d_2 \tag{13}$$

$$\lambda_{pd}^{iter} = y_{pd}^{iter} + v_{pd}^{iter} \tag{14}$$

**Transition Phase 2**

$$y_{pd}^{iter} = \begin{cases} 1 & \text{if}(\lambda_{pd}^{iter} > \alpha) \\ -1 & \text{if}(\lambda_{pd}^{iter} < -\alpha) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$xv_{pd}^{iter} = \begin{cases} G_d^{iter-1} & \text{if } (y_{pd}^{iter} = 1) \\ P_{pd}^{iter-1} & \text{if } (y_{pd}^{iter} = -1) \\ \text{any vehicle} & \text{otherwise} \end{cases} \tag{16}$$

### 3.3.2 Customer Sequencing

Because capacity constraints have been considered in the first stage, we just need to consider the total distance traveled by vehicles in the second stage. Thus the problem of customer sequencing for a vehicle is equivalent to a TSP problem. Initial solutions are first generated by using a greedy method, and then a SA algorithm is used to improve the initial solutions. Some worse solutions may be accepted by SA depending on the acceptance probability. Eq. (17) computes the amount of solution improvement. In Eq. (18), $p(S')$ is the acceptance probability that SA accepts new solution $S'$ that is worse than the current solution $S$.

$$\Delta = f(S') - f(S) \tag{17}$$

$$p(S') = \exp(-\Delta / t) \tag{18}$$

## 4 Experiments and Comparisons

To verify the proposed approach, we compare CPSO-SA with the algorithm proposed by Chen et al. (referred to as DPSO-SA) in terms of solution quality and CPU time. CPSO-SA was coded in Java and executed on a PC with 3.5GB of RAM and Intel Core 2 CPU E8400 3GHz. The parameters of CPSO-SA used for all CVRP problems are taken from the results of preliminary experiments. For CPSO, parameters are: *pop* (total number of particles) = $n$ (the number of customers), $\alpha = 0.45$, $w = 0.8$, $C_1 = 1.1$, $C_2 = 1.4$, max iteration number = 300. For SA, the parameters are set as follows: $t_0 = 3$, $t_f = 0.01$, $L$ (temperature length) = $n \times 2$, $\theta$ (cooling rate) = 0.8. All CVRP test problems are collected from the website: http://www.branchandcut.org/VRP/data/.

Fig. 3 presents the convergence trend of running CPSO-SA for solving the first test problem. Table 1 lists the best results of 16 test problems obtained by using the proposed algorithm. The CPU time required to find the best solution for each problem is also listed. The data of DPSO-SA are directly taken from Chen's paper for the purpose of comparison. The better results are typed in bold in this table. To fairly compare computational time, the data of Chen's work have been properly converted using the following equation: CPU time = Instruction count × CPI × Clock cycle time [14]. The comparison results demonstrate that the solutions of CPSO-SA are very close to those of Chen's approach but CPSO-SA takes much less CPU time.

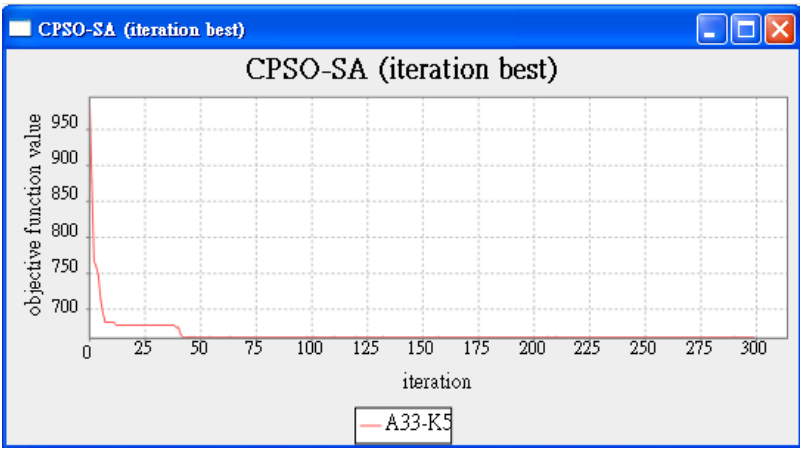



**Fig. 3.** Convergence trend when using CPSO-SA to solve problem A33-K5

**Table 1.** Computational results

| No. | problem | *n* | *v* | Objective function value | | | CPU time (second) | |
|---|---|---|---|---|---|---|---|---|
| | | | | BKS | DPSO-SA | CPSO-SA | DPSO-SA | CPSO-SA |
| 1 | A-n33-k5 | 32 | 5 | 661 | 661 | 661 | 19.4 | **0.7** |
| 2 | A-n46-k7 | 45 | 7 | 914 | **914** | 917 | 77.3 | **2.4** |
| 3 | A-n60-k9 | 59 | 9 | 1354 | 1354 | 1354 | 185.3 | **6.5** |
| 4 | B-n35-k5 | 34 | 5 | 955 | 955 | 955 | 22.6 | **1.2** |
| 5 | B-n45-k5 | 44 | 5 | 751 | 751 | 751 | 80.5 | **4.8** |
| 6 | B-n68-k9 | 67 | 9 | 1272 | **1272** | 1274 | 206.6 | **27.2** |
| 7 | B-n78-k10 | 77 | 10 | 1221 | 1239 | **1237** | 257.6 | **24.0** |
| 8 | E-n30-k3 | 29 | 3 | 534 | 534 | 534 | 17.0 | **0.3** |
| 9 | E-n51-k5 | 50 | 5 | 521 | 528 | **521** | 180.3 | **4.6** |
| 10 | E-n76-k7 | 75 | 7 | 682 | **688** | 692 | 315.9 | **9.5** |
| 11 | F-n72-k4 | 71 | 4 | 237 | 244 | **237** | 239.0 | **5.3** |
| 12 | F-n135-k7 | 134 | 7 | 1162 | 1215 | **1200** | 915.8 | **202.8** |
| 13 | M-n101-k10 | 100 | 10 | 820 | **824** | 825 | 524.5 | **6.1** |
| 14 | M-n121-k7 | 120 | 7 | 1034 | **1038** | 1039 | 1040.1 | **51.5** |
| 15 | P-n76-K4 | 75 | 4 | 593 | 602 | **596** | 297.8 | **27.6** |
| 16 | P-n101-k4 | 100 | 4 | 681 | 694 | **691** | 586.5 | **29.4** |

## 5 Conclusions

In our proposed approach, CPSO is first used to cluster customers into vehicles and then SA is employed to arrange the visiting order of each vehicle. A short solution representation has been proposed. Compared with the solution string used in [8], CPSO-SA can save computation time because it can reduce the number of infeasible

solutions. Experimental results show that CSPO-SA can effectively solve any of 16 CVRP problems within a reasonable time period. Applying CSPO-SA to VRPTW problems will be considered as the future work.

**Acknowledgements.** The authors are grateful to National Science Council, Taiwan, (Grant No. NSC 99-2410-H-036-003-MY2) for the financial support.

## References

1. Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem. Manage. Sci. 6(1), 80–91 (1959)
2. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective Vehicle Routing Problems. Eur. J. Oper. Res. 189, 293–309 (2008)
3. Cordeau, J.F., Laporte, G., Savelsbergh, M.W.P., Vigo, D.: Chapter 6 Vehicle Routing. In: Barnhart, C., Laporte, G. (eds.) Handbook in Operations Research and Management Science, vol. 14, pp. 367–428. Elsevier (2007)
4. Baker, B.M., Ayechew, M.A.: A Genetic Algorithm for the Vehicle Routing Problem. Comput. Oper. Res. 30, 787–800 (2003)
5. Bell, J.E., McMullen, P.R.: Ant Colony Optimization Techniques for the Vehicle Routing Problem. Adv. Eng. Inform. 18, 41–48 (2004)
6. Zhang, X., Tang, L.: A New Hybrid Ant Colony Optimization Algorithm for the Vehicle Routing Problem. Pattern Recognit. Lett. 30, 848–855 (2009)
7. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
8. Chen, A.L., Yang, G.K., Wu, Z.M.: Hybrid Discrete Particle Swarm Optimization Algorithm for Capacitated Vehicle Routing Problem. Journal of Zhejiang University Science A 7(4), 607–614 (2006)
9. Ai, T.J., Kachitvichyanukul, V.: Particle Swarm Optimization and Two Solution Representations for Solvingthe Capacitated Vehicle Routing Problem. Comput. Ind. Eng. 56, 380–387 (2009)
10. Marinakis, Y., Marinaki, M., Dounias, G.: A Hybrid Particle Swarm Optimization Algorithm for the Vehicle Routing Problem. Eng. Appl. Artif. Intell. 23(4), 463–472 (2010)
11. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
12. Jarboui, B., Cheikh, M., Siarry, P., Rebai, A.: Combinatorial Particle Swarm Optimization (CPSO) for PartitionalClustering Problem. Appl. Math. Comput. 92, 337–345 (2007)
13. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
14. Patterson, D.A., Hennessy, J.L.: Computer Organization and Design: the Hardware/Software Interface. Morgan Kaufmann, Burlington (2011)

# Adaptive Differential Evolution with Hybrid Rules of Perturbation for Dynamic Optimization

Krzysztof Trojanowski[1], Mikołaj Raciborski[2], and Piotr Kaczyński[2]

[1] Institute of Computer Science, Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw, Poland
[2] Cardinal Stefan Wyszyński University, Faculty of Mathematics and Natural Sciences
Wóycickiego 1/3, 01-938 Warsaw, Poland

**Abstract.** This work presents a differential evolution (DE) algorithm equipped with a new perturbation operator applied for dynamic optimization. The selected version of DE, namely the jDE algorithm has been extended by a new type of mutation mechanism which employs random variates controlled by the $\alpha$-stable distribution. Precisely, in the modified version of jDE the population of individuals consist of two types of members: a small number of those which undergo the new mutation procedure and much larger number of the remaining ones which are mutated according to the regular DE mutation. This hybrid structure of population makes the algorithm more effective for some types of the dynamic environments. The experiments were performed for two well known benchmarks: Generalized Dynamic Benchmark Generator (GDBG) and Moving Peaks Benchmark (MPB) reimplemented together as a new benchmark suite `Syringa`. Obtained results show advantages and disadvantages of the new approach.

## 1 Introduction

Optimization in dynamic environments is a continuous subject of interest for many research groups. In the case of dynamic optimization the algorithm has to cope with changes in the fitness landscape, that is, in the evaluation function parameters or even in the evaluation function formula which appear during the process of optimum search. There exists a number of dynamic optimization benchmarks designed to estimate efficiency of optimization algorithms. Among these benchmarks we selected two with the search space defined in $\mathbf{R}^n$ to evaluate the differential evolution (DE) approach and especially the idea of hybrid population application. DE approach which originated with the Genetic Annealing algorithm [1] has been studied from many points of view (for detailed discussion see, for example, monographs [2,3]). Among the number of instances of the DE approach we selected the jDE algorithm [4] which is distinct from the other DE algorithms in that some of its parameters, precisely *F* and *CR*, adaptively adjust their values during the run respectively to the varying conditions of the search process.

In the presented research we are interested in verification of the positive or negative role of a mutation operator originating from another heuristic approach when it is applied in the DE algorithm. The proposed type of mutation employs random variates

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 69–83.
DOI: 10.1007/978-3-642-35638-4_6 

controlled by the $\alpha$-stable distribution which already proved their usefulness in other version of evolutionary approach and in the particle swarm optimization as well.

The paper is organized as follows. In Section 2 a brief description of the optimization algorithm is presented. Section 3 discuss properties of new type of mutation introduced to jDE. The description of a new benchmark suite `Syringa` is given in Section 4 whereas Section 5 includes some details of the applied measure and the range of the performed tests. Section 6 shows the results of experiments. Section 7 concludes the presented research.

## 2 The jDE Algorithm

The differential evolution algorithm is an evolutionary method with a very specific mutation operator controlled by the scale factor $F$. Three different, randomly chosen solutions are needed to mutate a target solution $\mathbf{x}^i$: a base solution $\mathbf{x}^0$ and two difference solutions $\mathbf{x}^1$ and $\mathbf{x}^2$. After the selection of the three solutions, a mutant undergoes discrete recombination with the target solution which is controlled by the crossover probability factor $CR \in [0, 1]$. The new solutions created during the mutation step are called trial solutions. Finally, in the selection stage trial solutions compete with their target solutions for the place in the population. This strategy of population management is called ***DE/rand/1/bin*** which means that the base solution is **rand**omly chosen, **1** difference vector is added to it and the crossover is based on a set of independent decisions for each of coordinates, that is, a number of parameters donated by the mutant closely follows a **bin**omial distribution.

The jDE algorithm (depicted in Figure 1) extends functionality of the basic approach in many ways. First, each object representing a solution in the population is extended by a couple of its personal parameters $CR$ and $F$. They are adaptively modified every generation [4]. The next modifications have been introduced just for better coping in the dynamic optimization environment. The population of solutions has been divided into five subpopulations of size ten. Each of them has to perform its own search process, that is, no information is shared between subpopulations. Every solution is a subject to the aging procedure protecting against stagnation in local minima and just the global-best solution is excluded from this. To avoid overlapping between subpopulations a distance between subpopulation leaders is calculated and in the case of too close localization one of subpopulations is reinitialized. However, as in previous case the subpopulation with the global-best is never the one to reinitialize. The last extension is a memory structure called archive. The archive is increased after each change in the fitness landscape by the current global-best solution. Recalling from the archive can be executed every reinitialization of a subpopulation, however, decision about the execution depends on a few conditions. Details of the above-mentioned extensions can be found in [5].

## 3 Proposed Extension of jDE

The novelty in the algorithm concerns introduction of a new type of solutions into the population of size $M$. A small number of new solutions, that is, just one, two, or three pieces replace the classic ones so the population size remains the same. The difference

**Algorithm 1** jDE algorithm

Create and initialize the reference set of $(k \cdot m)$ solutions
**repeat**
  **for** $l = 1$ to $k$ **do** {*for each subpopulation*}
    **for** $i = 1$ to $m$ **do** {*for each solution in a subpopulation*}
      Select randomly three solutions: $\mathbf{x}^{l,0}$, $\mathbf{x}^{l,1}$, and $\mathbf{x}^{l,2}$ such that: $\mathbf{x}^{l,i} \neq \mathbf{x}^{l,0}$ and $\mathbf{x}^{l,1} \neq \mathbf{x}^{l,2}$
      **for** $j = 1$ to $n$ **do** {*for each dimension in a solution*}
        **if** $(rand(0,1) > CR^{l,i})$ **then**
          $u_j^{l,i} = x_j^{l,0} + F^{l,i} \cdot (x_j^{l,1} - x_j^{l,2})$
        **else**
          $u_j^{l,i} = x_j^{l,i}$
        **end if**
      **end for**
    **end for**
  **end for**
  **for** $i = 1$ to $(k \cdot m)$ **do** {*for each solution*}
    **if** $(f(\mathbf{u}^i) < f(\mathbf{x}^i)$ **then** {*Let's assume this is a minimization problem*}
      $\mathbf{x}^i = \mathbf{u}^i$
    **end if**
    Recalculate $F^i$ and $CR^i$
    Apply aging for $\mathbf{x}^i$
  **end for**
  Do overlapping search
**until** the stop condition is satisfied

between the classic solutions and the new ones lies in the way they are mutated. The new type of mutation operator is based on the rules of movement governing quantum particles in mQSO [6].

In the first phase of the mutation, we generate a new point in the search space. The new point is uniformly distributed within a hypersphere surrounding the mutated solution. In the second phase, the point is shifted along the direction determined by the hypersphere center and the point. The distance $d'$ from the hypersphere center to the final location of the point is calculated as follows:

$$d' = d \cdot S\alpha S(0, \sigma) \cdot \exp(-f'(\mathbf{x}_i)), \tag{1}$$

where $d$ is a distance from the original location obtained in the first phase, $S\alpha S(\cdot, \cdot)$ denotes a symmetric $\alpha$-stable distribution variate, $\sigma$ is evaluated as in eq. (2) and $f'(\mathbf{x}_i)$ as in eq. (3):

$$\sigma = r_{S\alpha S} \cdot (D_{\mathrm{w}}/2) \tag{2}$$

$$f'(\mathbf{x}_i) = \frac{f(\mathbf{x}_i) - f_{\min}}{(f_{\max} - f_{\min})} \tag{3}$$

where:

$$f_{\max} = \max_{j=1,\ldots,M} f(\mathbf{x}_j), \;\; f_{\min} = \min_{j=1,\ldots,M} f(\mathbf{x}_j).$$

The $\alpha$-stable distribution (called also a Levý distribution) is controlled by four parameters: stability index $\alpha$ ($\alpha \in (0, 2]$), skewness parameter $\beta$, scale parameter $\sigma$ and location parameter $\mu$. In our case we assume $\mu = 0$ and apply the symmetric version of this distribution (denoted by $S\alpha S$ for "**S**ymmetric $\alpha$-**S**table distribution"), where $\beta$ is set to 0.

The resulting behavior of the proposed operator is characterized by two parameters: the parameter $r_{S\alpha S}$ which controls the mutation strength, and the parameter $\alpha$ which determines the shape of the $\alpha$-stable distribution. The solutions mutated in this way are labeled as $\mathbf{s}_{\text{Levý}}$ in the further text.

## 4 The `Syringa` Benchmark Suite

For the experimental research we developed a new testing environment `Syringa` which is able to simulate behavior of a number of existing benchmarks and to create completely new benchmark instances as well. The structure of the `Syringa` code originates from a fitness landscape model where the landscape consists of a number of simple components. A sample dynamic landscape consists of a number of components of any types and individually controlled by a number of parameters. Each of the components covers a subspace of the search space. The final landscape is the result of a union of a collection of components such that each of the solutions from the search space is covered by at least one component. In the case of a solution belonging to the intersection of a number of components the solution value equals (1) the minimum (for minimization problems) or (2) maximum (otherwise) value among the values obtained for the intersected components or (3) this can be also a sum of the fitness vales obtained from these components. Eventually, the `Syringa` structure is a logical consequence of the following assumptions:

1. the fitness landscape consists of a number of any different component landscapes,
2. the dynamics of each of the components can be different and individually controlled,
3. a component can be defined for a part or the whole of the search space, thus, in the case of a solution covered by more than one component the value of this solution can be the minimum, the maximum or the sum of values returned by the covering components.

### 4.1 The Components

Current version of `Syringa` consists of six types of component functions (Table 1) defined for the real-valued search space. All formulas include the number of the search apace dimensions $n$ which makes them able to define search spaces of any given complexity.

There can be defined a number of parameters which individually define the component properties and allow to introduce dynamics as well. For each of the components we can distinguish two groups of parameters which influence the formula of the component fitness function: the parameters from the former one are embedded in the component function formula whereas the parameters from the latter one control rather the output

**Table 1.** `Syringa` components

| name | formula | domain |
|---|---|---|
| Peak ($F_1$) | $f(\mathbf{x}) = \frac{1}{1+\sum_{j=1}^{n} x_j^2}$ | [-100,100] |
| Cone ($F_2$) | $f(\mathbf{x}) = 1 - \sqrt{\sum_{j=1}^{n} x_j^2}$ | [-100,100] |
| Sphere ($F_3$) | $f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$ | [-100,100] |
| Rastrigin ($F_4$) | $f(\mathbf{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$ | [-5,5] |
| Griewank ($F_5$) | $f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n}(x_i)^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$ | [-100,100] |
| Ackley ($F_6$) | $f(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) -$ $\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$ | [-32,32] |

of the formula application. For example, when we want to stretch the landscape over the search space each of the solution coordinates is multiplied by a scaling factor. For a non–uniform stretching we need to use a vector of factors containing individual values for each of the coordinates. We call this type of modification a horizontal scaling and this represents the first type of component changes. The example of the second type is a vertical scaling where just the fitness value of a solution is multiplied by a scaling factor. The first group of parameters controls changes like horizontal translation, horizontal scaling, and rotation. For simplicity they are called *horizontal changes* in the further text. The second group of changes (called respectively *vertical changes*) is represented by vertical scaling and vertical translation. All of the changes can be obtained by dynamic modification of respective parameters during the process of search.

### 4.2 Horizontal Change Parameters

In this case the coordinates of the solution $\mathbf{x}$ (a vector, that is, a matrix of size $n$ by 1) are modified before the component function equation is applied. The new coordinates are obtained with the following formula:

$$\mathbf{x}' = \mathbf{M} \cdot (\mathbf{W} \cdot (\mathbf{x} + \mathbf{X})) \tag{4}$$

where $\mathbf{X}$ is a translation vector, $\mathbf{W}$ is a diagonal matrix of scaling coefficients for the coordinates, and $\mathbf{M}$ is an orthogonal rotation matrix.

### 4.3 Vertical Change Parameters

Changes of the fitness function value are executed according to the following formula:

$$f'(\mathbf{x}) = f(\mathbf{x}) \cdot v + h \tag{5}$$

where $v$ is a vertical scaling coefficient and $h$ is a vertical translation coefficient.

### 4.4 Parameters Control

In the case of dynamic optimization the fitness landscape components has to change the values of their parameters during the process of search. There were defined four different characteristics of variability which were applied to the component parameters: small step change ($\mathbf{T}_1$—eq. (6)), large step change ($\mathbf{T}_2$—eq. (7)), and two versions of random changes ($\mathbf{T}_3$—eq. (8) and $\mathbf{T}_4$—eq. (9)). The change $\Delta$ of a parameter value is calculated as follows:

$$\Delta = \alpha \cdot r \cdot (\max - \min), \quad \text{where } \alpha = 0.04,\ r = U(0,1), \tag{6}$$

$$\Delta = (\alpha \cdot sign(r_1) + (\alpha_{\max} - \alpha) \cdot r_2) \cdot (\max - \min), \quad \text{where } \alpha = 0.04,\ r_{1,2} = U(0,1),\ \alpha_{\max} = 0.1 \tag{7}$$

$$\Delta = N(0,1) \tag{8}$$

$$\Delta = U(r_{\min}, r_{\max}) \tag{9}$$

In the above-mentioned equations $\max$ and $\min$ represent upper and lower boundary of the search space, $N(0,1)$ is a random value obtained with standardized normal distribution, $U(a,b)$ is a random value obtained with uniform distribution form the range $[a,b]$, and $[r_{\min}; r_{\max}]$ define the feasible range of $\Delta$ values.

The model of `Syringa` assumes that the component parameter control is separated from the component, that is, a dynamic component has to consist of two objects: the first one represents an *evaluator* of solutions (that is, a component of any type mentioned in Table 1) and the second one is an *agent* which controls the behavior of the evaluator. The agent defines initial set of values for the evaluator parameters and during the process of search the values are updated by the agent according to the assumed characteristic of variability. Properties of all the types of components are unified so as to make possible assignment of any agent to any component. This architecture allows to create multiple classes of dynamic landscapes. In the presented research we started with simulation of two existing benchmarks: Generalized Dynamic Benchmark Generator (GDBG) [7] and the Moving Peaks Benchmark generator [8]. In both cases optimization is carried out in a real-valued multidimensional search space, and the fitness landscape is built of multiple component functions controlled individually by their parameters. For appropriate simulation of any of the two benchmarks there are just two things to do: select a set of the components and build agents which will control the components in the same manner like in the simulated benchmark.

### 4.5 Reimplementation of Moving Peaks Benchmark (MPB)

In the case of MPB, three scenarios of the benchmark parameters control are defined [8]. We performed experiments for the first and the second scenario.

The selected fitness landscape consists of a set of peaks ($F_1$ — the first scenario) or cones ($F_2$ — the second scenario) which undergo two types of horizontal changes:

the translation and the scaling and just one vertical change, that is, the translation. The horizontal scaling operator has the same scale coefficient for each of the dimensions, so in this specific case this coefficient is represented as a one-dimensional variable $w$ instead of the vector $\mathbf{W}$.

The parameters $\mathbf{X}$, $w$ and $v$ are embedded into the peak function formula $f(\mathbf{x})$ in the following way:

$$f_{\text{peak}}(\mathbf{x}) = \frac{v}{1 + w \cdot \sum_{j=1}^{n} \frac{(\mathbf{x}_j - \mathbf{X}_j)^2}{n}} \tag{10}$$

The parameters $\mathbf{X}$, $w$ and $h$ are embedded into the cone function formula $f(\mathbf{x})$ in the following way:

$$f_{\text{cone}}(\mathbf{x}) = h - w \cdot \sqrt{\sum_{j=1}^{n} (\mathbf{x}_j - \mathbf{X}_j)^2} \tag{11}$$

All the modifications of the component parameters belong to the fourth characteristic of variability $\mathbf{T}_4$ where for every change $r_{\min}$ and $r_{\max}$ are redefined in the way to keep the value of each modified parameter in the predefined interval of feasible values. Simply, for every modified parameter of translation or scaling, which can be represented as a symbol $p$: $r_{\min} = p_{\min} - p$ and $r_{\max} = p_{\max} - p$. For the horizontal scaling the interval is set to $[1; 12]$ and for the vertical scaling — to $[30; 70]$. For the horizontal translation there is a constraint for the euclidean length of the translation: $|\mathbf{X}| \leq 3$. For both scenarios in the first version there are ten moving components whereas in the second version 50 moving components is in use.

### 4.6 Reimplementation of Generalized Dynamic Benchmark Generator (GDBG)

GDBG consists of two different benchmarks: Dynamic Rotation Peak Benchmark Generator (DRPBG) and Dynamic Composition Benchmark Generator (DCBG). There are five types of component functions: peak ($F_1$), sphere ($F_3$), Rastrigin ($F_4$), Griewank ($F_5$), and Ackley ($F_6$). $F_1$ is the base component for DRPBG whereas all the remaining types are employed in DCBG.

**Dynamic Rotation Peak Benchmark Generator (DRPBG).** There are four types of the component parameter modification applied in DRPBG: horizontal translation, scaling and rotation and vertical scaling. As in the case of MPB the horizontal scaling operator has the same scale coefficient for each of the dimensions, so in this specific case this coefficient is also represented as a one-dimensional variable $w$ instead of the vector $\mathbf{W}$. The component function formula is the same as in the eq. (10).

Values of the translation vector $\mathbf{X}$ in subsequent changes are evaluated with use of the rotation matrix $\mathbf{M}$. Clearly, we apply the rotation matrix to the current coordinates of the component function optimum $\mathbf{o}$, that is: $\mathbf{o}(t+1) = \mathbf{o}(t) \cdot \mathbf{M}(t)$ (where $t$ is the number of the current change in the component) and then the final value of $\mathbf{X}(t+1)$ is calculated: $\mathbf{X}(t+1) = \mathbf{o}(t+1) - \mathbf{o}(0)$.

Subsequent values of the horizontal scaling parameter $w$ and the vertical scaling parameter $v$ are evaluated according to the first, the second or the third characteristic of variability, that is, $\mathbf{T}_1$, $\mathbf{T}_2$ or $\mathbf{T}_3$.

For every change a new rotation matrix $\mathbf{M}$ is generated which is common for all the components. The rotation matrix $M$ is obtained as a result of multiplication of a number of rotation matrices $R$ where each of $R$ represents rotation in just one plane of the multidimensional search space. A matrix $R_{ij}(\theta)$ represents rotation by the $\theta$ angle along the plane $i$–$j$ and such a matrix can be easily generated as described by [9]. In DRPBG we start with a selection of the rotation planes, that is, we need to generate a vector $\mathbf{r}$ of size $l$ where $l$ is an even number and $l \leq n/2$. The vector contains search space dimension indices selected randomly without repetition. Then for every plane defined in $\mathbf{r}$ by subsequent pairs of indices: $[1,2]$, $[3,4]$, $[5,6]$, $\ldots$ $[l-1,l]$ a rotation angle is randomly generated and finally respective matrices $R_{r[1],r[2]}, \ldots R_{r[l-1],r[l]}$ are calculated. Eventually, the rotation matrix $M$ is calculated as follows:

$$M(t) = R_{r[1],r[2]}(\theta(t)) \cdot R_{r[3],r[4]}(\theta(t)) \cdots R_{r[l-1],r[l]}(\theta(t)). \tag{12}$$

In `Syringa` the method of the rotation matrix generation slightly differs from the one described above. Instead of the vector $\mathbf{r}$ there is a vector $\Theta$ which represents a sequence of rotation angles for all the possible planes in the search space. The position in the vector $\Theta$ defines the rotation plane. Simply, $\Theta(1)$ represents the plane [1,2], $\Theta(2)$ represents the plane [2,3] and so on until the plane [n-1,n]. The next values in $\Theta$ represent planes created from every second dimensions, that is, [1,3], [2,4] and so on until the plane [n-2,n]. Then values in $\Theta$ represent planes created from every third dimensions, then those created from every fourth, and so on until there appears the value for the last plane created from the first and the last dimension. If $\Theta(i)$ equals zero, then there is no rotation for the $i$-th plane, otherwise the respective rotation matrix $R$ is generated. The final stage of generation of the matrix $M$ is the same as in the description above, that is, the rotation matrix $M$ is the result of multiplication of all the matrices $R$ generated from the vector $\Theta$.

The matrix $\mathbf{M}$ is used twice for the evaluation of the component modification parameters: the first time when the translation vector $\mathbf{X}$ is calculated and the second time when the rotation is applied, that is, just before the application of the equation (10).

**Dynamic Composition Benchmark Generator (DCBG).** DCBG performs five types of the component parameter modification: horizontal translation, scaling and rotation and vertical translation and scaling. The respective parameters are embedded into the function formula $f"(\mathbf{x})$ in the following way [10,11]:

$$f"(\mathbf{x}) = (v \cdot (f'(\mathbf{M} \cdot (\mathbf{W} \cdot (\mathbf{x} + \mathbf{X}))) + h)) \tag{13}$$

where:

— $v$ is the weight coefficient depending of the currently evaluated $\mathbf{x}$,
— $\mathbf{W}$ is called a stretch factor which equals 1 when the search range of $f(\mathbf{x})$ is the same as the entire search space and grows when the search range of $f(\mathbf{x})$ decreases,

— $f'(\mathbf{x})$ represent the value of $f(\mathbf{x})$ normalized in the following way: $f'(\mathbf{x}) = C \cdot f(\mathbf{x})/|f_{\max}|$ where the constant $C = 2000$ and $f_{\max}$ is the estimated maximum value of function $f$ which is one of the four: sphere ($F_3$), Rastrigin ($F_4$), Griewank ($F_5$), or Ackley ($F_6$).

In `Syringa` the properties of some of the parameters has had to be changed. The first difference is in the evaluation of the weight coefficient $v$ which due to the structure of the assumed model cannot depend of the currently evaluated $\mathbf{x}$. Therefore, we assumed that $v = 1$. There is also no scaling, that is, $\mathbf{W}$ is an identity matrix because we assumed that the component functions are always defined for the entire search space. The last issue is about the rotation matrix $\mathbf{M}$ which is calculated in the same way as for the `Syringa` version of DRPBG. Eventually, the `Syringa` version of $f"(\mathbf{x})$ looks as follows:

$$f"(\mathbf{x}) = ((f'(\mathbf{M} \cdot ((\mathbf{x} + \mathbf{X}))) + h)) \tag{14}$$

Thus, the `Syringa` version of DCBG differs from the original one because it does not contain the horizontal scaling, the rotation matrix $\mathbf{M}$ is evaluated in the different way and the stretch factor always equals one. However, a kind of the vertical scaling is still present and can be found in the step of the $f(\mathbf{x})$ normalization.

## 5 Plan of Experiments

### 5.1 Performance Measure

For comparisons between the results obtained for different benchmark instances and different versions of the algorithms the offline error [8] (briefly *oe*) was selected. The measure represents the average deviation from the optimum of the best individual value since the last change in the fitness landscape. Formally:

$$oe = \frac{1}{N_{\text{changes}}} \sum_{j=1}^{N_{\text{changes}}} \left( \frac{1}{N_{\text{e}}(j)} \sum_{i=1}^{N_{\text{e}}(j)} (f(\mathbf{x}^{*j}) - f(\mathbf{x}_{\text{best}}^{ji})) \right) , \tag{15}$$

where $N_{\text{e}}(j)$ is a total number of solution evaluations performed for the $j$-th static state of the landscape, $f(\mathbf{x}^{*j})$ is the value of an optimal solution for the $j$-th landscape and $f(\mathbf{x}_{\text{best}}^{ji})$ is the current best value found for the $j$-th landscape. It should be clear that the measure *oe* should be minimized, that is, the better result the smaller the value of *oe*.

Our algorithm has no embedded strategy for detecting changes in the fitness landscapes. Simply, the last step in the main loop of the algorithm executes the reevaluation of the entire current solution set. Therefore, our optimization system is informed of the change as soon as it occurs, and no additional computational effort for its detection is needed.

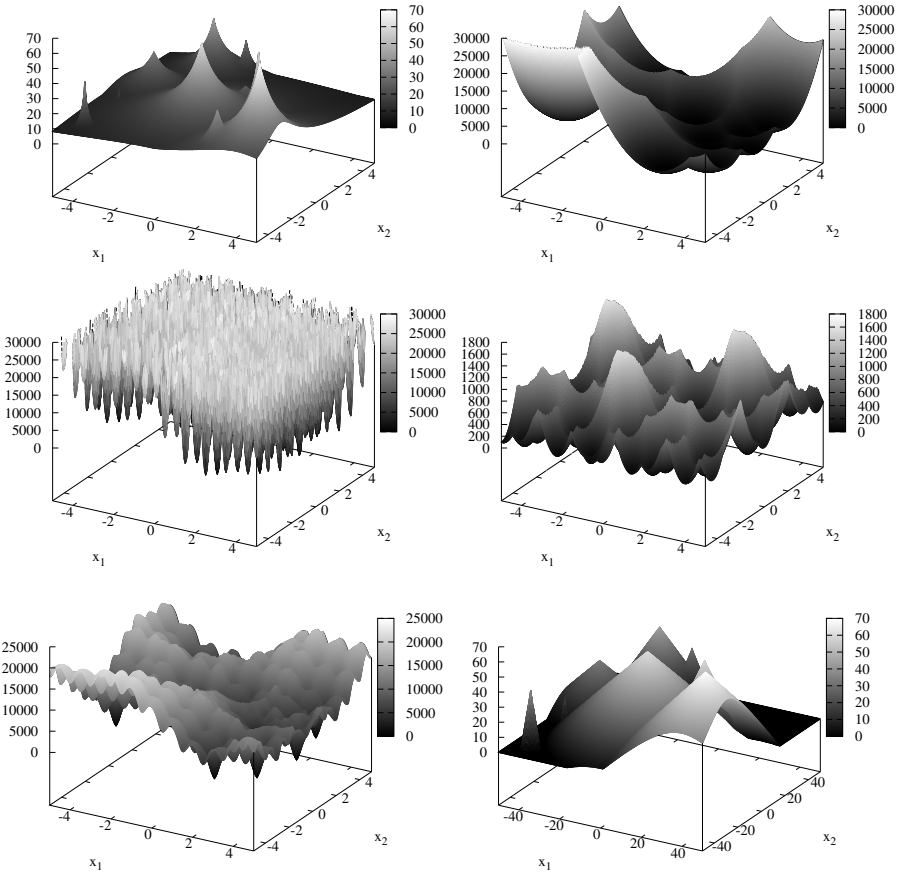



**Fig. 1.** Fitness landscapes of the benchmark instances for 2D search space: DRPBG with ten peak components and DCBG with ten sphere components (first row), DCBG with ten Rastrigin components and DCBG with ten Griewank components (second row), DCBG with ten Ackley components and MPB sc. 2 with ten cones (the last row)

### 5.2 The Tests

We performed experiments with a subset of GDBG benchmark functions as well as with four versions of MPB. For each of the components a feasible domain is defined which, unfortunately, is not the same in every case (see the last column in Table 1). For this reason the boundaries for the search space dimensions in the test-cases are not the same but adjusted respectively to the components. For GDBG the feasible search space is within the hypercube with the same boundaries for each dimension, namely $[-7.1, 7.1]$ whereas for MPB — $[-50, 50]$. These box constraints mean that both solutions and components should not leave this area during the entire optimization process.

**Table 2.** The least offline error (*oe*) mean values obtained for all the benchmark instances; the instances are sorted in ascending order of *oe*; for each of the instances three values are presented for three change types of the GDBG control parameters: $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ (except for MPB where there are two cases: with ten and 50 peaks/cones)

| the benchmark instance | change type | $\alpha$ | $ind_{S\alpha S}$ | *oe* |
|---|---|---|---|---|
| $F_5$ (Griewank) | $\mathbf{T}_1$ | 1.25 | 3.00 | 0.16813 |
| | $\mathbf{T}_2$ | 1.50 | 3.00 | 0.13977 |
| | $\mathbf{T}_3$ | 2.00 | 3.00 | 0.35606 |
| MPB sc. 1 with ten peaks | | 1.00 | 1.00 | 0.35622 |
| MPB sc. 1 with 50 peaks | | 1.50 | 1.00 | 0.66492 |
| DCBG with $F_3$ (sphere components) | $\mathbf{T}_1$ | 1.50 | 2.00 | 1.22583 |
| | $\mathbf{T}_2$ | 0.75 | 1.00 | 1.81625 |
| | $\mathbf{T}_3$ | 1.75 | 3.00 | 5.32476 |
| DRPBG with ten $F_1$ | $\mathbf{T}_1$ | 2.00 | 1.00 | 1.98783 |
| | $\mathbf{T}_2$ | 1.00 | 1.00 | 2.51758 |
| | $\mathbf{T}_3$ | 0.50 | 1.00 | 3.76098 |
| DRPBG with 50 $F_1$ | $\mathbf{T}_1$ | 1.25 | 1.00 | 3.35855 |
| | $\mathbf{T}_2$ | 1.00 | 1.00 | 4.24594 |
| | $\mathbf{T}_3$ | 0.75 | 1.00 | 5.87459 |
| MPB sc. 2 with ten cones | | — | 0.00 | 4.11117 |
| MPB sc. 2 with 50 cones | | 0.50 | 1.00 | 3.74856 |
| DCBG with $F_6$ (Ackley components) | $\mathbf{T}_1$ | 2.00 | 1.00 | 8.41941 |
| | $\mathbf{T}_2$ | 1.50 | 1.00 | 9.77345 |
| | $\mathbf{T}_3$ | 1.50 | 1.00 | 14.24450 |
| DCBG with $F_4$ (Rastrigin components) | $\mathbf{T}_1$ | — | 0.00 | 570.72 |
| | $\mathbf{T}_2$ | — | 0.00 | 610.565 |
| | $\mathbf{T}_3$ | — | 0.00 | 661.395 |

The number of fitness function evaluations between subsequent changes was calculated according to the rules as in the CEC'09 competition, that is, for $10^4 \cdot n$ fitness function calls between subsequent changes where $n$ is a number of search space dimensions and in this specific case $n$ equals five for all of the benchmark instances.

To decrease the number of algorithm configurations which would be experimentally verified we decided to fix the value of the parameter $r_{S\alpha S}$ and the only varied parameter was $\alpha$. In the preliminary phase of experimental research we tested efficiency of the algorithm for different values of $r_{S\alpha S}$ and analyzed obtained values of error. Eventually, for GDBG $r_{S\alpha S} = 0.6$ whereas for MPB it is ten times smaller, that is, $r_{S\alpha S} = 0.06$. Thus, for each of the benchmark instances there were performed just 32 experiments: for $\alpha$ between 0.25 and 2 varying with step 0.25 and for 0, 1, 2 and 3 solutions of

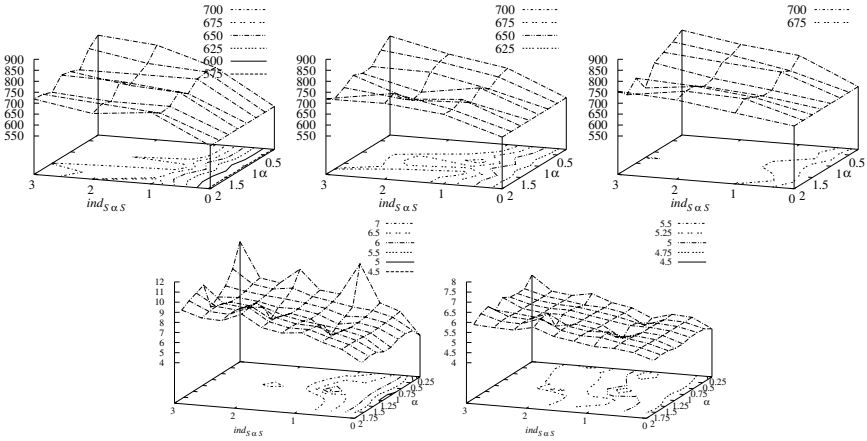


**Fig. 2.** Characteristics of *oe* for jDE: the cases where hybrid solution deteriorates the results, i.e., DCBG with $F_4$ (Rastrigin components) (the first row) and MPB sc.2 with 10 and 50 $F_2$ (cones) (the second row); the subsequent graphs in the first row represent three change types of the GDBG control parameters: $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$

new type present in the population. For each of the configurations the experiments were repeated 20 times and each of them consisted of 60 changes in the fitness landscape. Graphs and tables present mean values of *oe* calculated for these series.

## 6 The Results

The best values of offline error obtained for each of the benchmark instances are presented in Table 2. The table contains names of the instances, the mutation parameter configurations (i.e. values of $\alpha$ and $ind_{S\alpha S}$) and the mean values of *oe*. The instances are sorted in ascending order of obtained *oe* values (more or less). This way we can easily show which of the instances were the most difficult and which were the easiest.

All the results are depicted in Figures 2 and 3. The graphs are divided into two groups: the first one where due to the presence of $\mathbf{s}_{\text{Levý}}$ solutions obtained results deteriorated (Figure 2), and the second one where application of $\mathbf{s}_{\text{Levý}}$ solutions in the population improved the results, that is, obtained error decreased (Figure 3).

For each of the benchmark instances there were obtained 32 values of *oe* which are presented in a graphical form as a surface generated for different values of the number of $\mathbf{s}_{\text{Levý}}$ solutions (that is, $ind_{S\alpha S}$) and $\alpha$. The benchmark instances in the Figures are ordered in the same way as in Table 2.

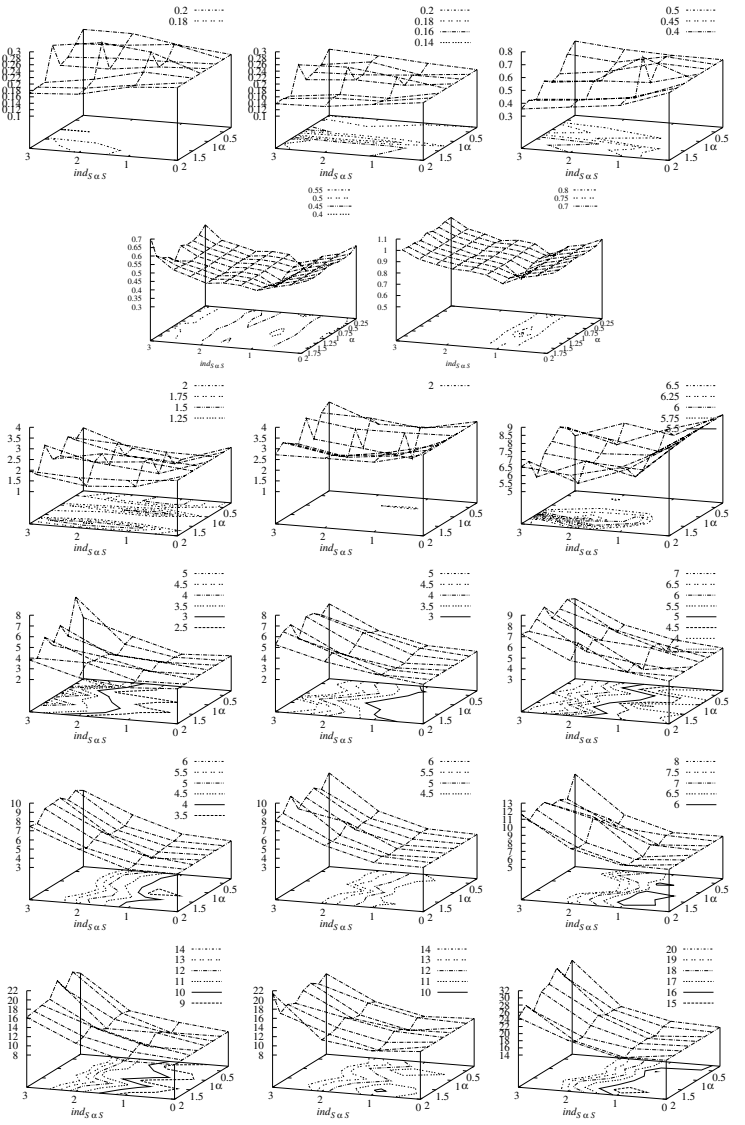


**Fig. 3.** Characteristics of *oe* for jDE: the cases where hybrid solution improves the results: DCBG with $F_5$ (Griewank components) (the first row), MPB sc.1 with ten and with 50 $F_1$ (peaks) (the second row) DCBG with $F_2$ (sphere components) (the third row), DRPBG with ten and with 50 $F_1$ (peaks) (the fourth and the fifth row), DCBG with $F_6$ (Ackley components) (the last row); the columns represent three change types of the GDBG control parameters: $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ (except for MPB where there are just two cases: with ten and 50 peaks)

## 7 Conclusions

In this paper we introduce hybrid structure of population in differential evolution algorithm jDE and study its properties when applied to dynamic optimization tasks. This is a case of a kind of technology transfer where promising mechanisms form one heuristic approach appear to be useful in another. The results show that mutation operator using random variates based on $\alpha$-stable distribution, that is, the operator where both fine local modification and macro-jumps can appear, improves the values of *oe*. Lack of improvement appeared in two cases, that is, for the DCBG with F4 (Rastrigin components) which is a very difficult landscape looking like a hedgehog (Figure 1, the graph in the second row on the left) and for the MPB sc.2 with ten cones. In both cases macromutation most probably introduces an unnecessary noise rather than a chance for exploration of a new promising area. Difficulty of the former benchmark instance is confirmed by extremely high values of error obtained for all three types of change.

In the remaining cases the influence of $\mathbf{s}_{\text{Levý}}$ solution presence is positive, however, it must be stressed that the number of these solutions should be small. In most cases the best results were obtained for just one piece. The exception is the DCBG with F5 (Grievank components) which was the easiest landscape for the algorithm, easier even than the landscape build of the spheres. In the case of the DCBG with F5 the higher number of $\mathbf{s}_{\text{Levý}}$, the smaller value of *oe* (for the discussion on other aspects of the influence of $\mathbf{s}_{\text{Levý}}$ solutions on the jDE effectiveness the reader is referred to [12]).

Finally, it is worth to stress that the different complexity of the tested instances shows also that when we take them as a suite of tests and evaluate overall gain of the algorithm we need to apply different weight for the successes obtained for each of the instances. Simply, an improvement of efficiency obtained for DCBG with Grievank components have different significance than the same improvement for, e.g., DCBG with Ackley components.

## References

1. Price, K.V.: Genetic annealing. Dr. Dobb's Journal, 127–132 (1994)
2. Feokistov, V.: Differential Evolution, In Search of Solutions. Optimization and Its Applications, vol. 5. Springer (2006)
3. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution. A Practical Approach to Global Optimization. Natural Computing Series. Springer (2005)
4. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10, 646–657 (2006)
5. Brest, J., Zamuda, A., Boskovic, B., Maucec, M.S., Zumer, V.: Dynamic optimization using self-adaptive differential evolution. In: IEEE Congr. on Evolutionary Computation, pp. 415–422. IEEE (2009)
6. Trojanowski, K.: Properties of quantum particles in multi-swarms for dynamic optimization. Fundamenta Informaticae 95, 349–380 (2009)
7. Li, C., Yang, S.: A Generalized Approach to Construct Benchmark Problems for Dynamic Optimization. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 391–400. Springer, Heidelberg (2008)

8. Branke, J.: Memory enhanced evolutionary algorithm for changing optimization problems. In: Proc. of the Congr. on Evolutionary Computation, vol. 3, pp. 1875–1882. IEEE Press, Piscataway (1999)
9. Salomon, R.: Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. BioSystems 39, 263–278 (1996)
10. Liang, J.J., Suganthan, P.N., Deb, K.: Novel composition test functions for numerical global optimization. In: IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, pp. 68–75 (2005)
11. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore (2005)
12. Trojanowski, K., Raciborski, M., Kaczyński, P.: Self-adaptive differential evolution with hybrid rules of perturbation for dynamic optimization. Journal of Telecommunications and Information Technology, 20–30 (2011)

# Modified Constrained Differential Evolution for Solving Nonlinear Global Optimization Problems

Md. Abul Kalam Azad and M.G.P. Fernandes

Algoritmi R&D Centre, University of Minho, 4710-057 Braga, Portugal
{akazad,emgpf}@dps.uminho.pt

**Abstract.** Nonlinear optimization problems introduce the possibility of multiple local optima. The task of global optimization is to find a point where the objective function obtains its most extreme value while satisfying the constraints. Some methods try to make the solution feasible by using penalty function methods, but the performance is not always satisfactory since the selection of the penalty parameters for the problem at hand is not a straightforward issue. Differential evolution has shown to be very efficient when solving global optimization problems with simple bounds. In this paper, we propose a modified constrained differential evolution based on different constraints handling techniques, namely, feasibility and dominance rules, stochastic ranking and global competitive ranking and compare their performances on a benchmark set of problems. A comparison with other solution methods available in literature is also provided. The convergence behavior of the algorithm to handle discrete and integer variables is analyzed using four well-known mixed-integer engineering design problems. It is shown that our method is rather effective when solving nonlinear optimization problems.

**Keywords:** Nonlinear programming, Global optimization, Constraints handling, Differential evolution.

## 1 Introduction

Problems involving global optimization over continuous spaces are ubiquitous throughout the scientific community. Many real world problems are formulated as mathematical programming problems involving continuous variables with linear/nonlinear objective function and constraints. The constraints can be of inequality and/or equality type. Generally, the constrained nonlinear optimization problems are formulated as follows:

$$\begin{array}{lll} \text{minimize} & f(\boldsymbol{x}) & \\ \text{subject to} & g_k(\boldsymbol{x}) \leq 0 & k = 1, 2, \ldots, m_1 \\ & h_l(\boldsymbol{x}) = 0 & l = 1, 2, \ldots, m_2 \\ & lb_j \leq x_j \leq ub_j & j = 1, 2, \ldots, n, \end{array} \tag{1}$$

where $f, g_k, h_l : \mathbb{R}^n \longrightarrow \mathbb{R}$ with feasible set $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{g}(\boldsymbol{x}) \leq 0, \boldsymbol{h}(\boldsymbol{x}) = 0$ and $\boldsymbol{lb} \leq \boldsymbol{x} \leq \boldsymbol{ub}\}$. $f$, $g_k$, $h_l$ may be differentiable and the information about derivatives may or may not be provided.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 85–100.
DOI: 10.1007/978-3-642-35638-4_7 

Problem (1) involving global optimization (here a minimization problem) of a multivariate function with constraints is widespread in the mathematical modeling of real world systems. Many problems can be described only by nonlinear relationships, which introduce the possibility of multiple local minima. The task of global optimization is to find a point where the objective function obtains its most extreme value, the global minimum, while satisfying the constraints.

Several deterministic and stochastic solution methods with different constraints handling techniques have been proposed to solve (1). Unlike the stochastic methods, the outcome of a deterministic algorithm does not depend on pseudo random variables. In general, its performance depends heavily on the structure of the problem since the design relies on the mathematical attributes of the optimization problem. Compared with deterministic methods, the implementation of stochastic algorithms is often easier. To handle the constraints of the problem, some methods try to make the solution feasible by repairing the infeasible one or penalizing an infeasible solution with a penalty function. However, to find the appropriate penalty parameter is not an easy task. Deb [6] proposed an efficient constraints handling technique for genetic algorithm based on the feasibility and dominance rules. The author used a penalty function that does not require any penalty parameter. Barbosa and Lemonge [2] proposed a parameter-less adaptive penalty scheme for genetic algorithm. In the very recent paper the authors proposed this adaptive penalty scheme for differential evolution [23]. Hedar and Fukushima [12] proposed a simulated annealing method that uses the filter method [10] rather than the penalty function method to handle the constraints. Runarsson and Yao proposed a stochastic ranking [21] and a global competitive ranking [22] techniques for constrained nonlinear optimization problems based on evolution strategy. The authors presented a new view on the usual penalty function methods in terms of the dominance of penalty and objective functions. Dong et al. [8] proposed a swarm optimization based on the constraint fitness priority-based ranking technique. Zahara and Hu [28] proposed a hybrid of Nelder-Mead simplex method and a particle swarm optimization based on this technique [8]. Rocha and Fernandes proposed a electromagnetism-like algorithm based on the feasibility and dominance rules [18] and the self-adaptive penalties [19]. Rocha et al. [20] used an augmented Lagrangian method coupled with an artificial fish swarm algorithm for global optimization. Coello Coello [4] proposed constraints handling using an evolutionary multiobjective optimization technique. Coello Coello and Cortés [5] proposed hybridizing of a genetic algorithm with an artificial immune system that uses genotypic-based distances to move from infeasible solution to feasible one. Another constraints handling technique is the multilevel Pareto ranking based on the constraints matrix [16,17]. Ray and Tai [16] proposed an evolutionary algorithm with a multilevel pairing strategy and Ray and Liew [17] proposed a society and civilization algorithm based on the simulation of social behavior.

Differential evolution (DE) proposed by Storn and Price [24] is a population-based heuristic approach that is very efficient to solve global optimization problems with simple bounds. DE performance depends on the amplification factor of differential variation and crossover control parameter. Hence adaptive control parameters have been implemented in DE in order to obtain a competitive algorithm. Further, to improve solution accuracy, techniques that are able to exploit locally certain regions, detected in

the search space as promising, are also required. When the solutions ought to be restricted to a set of inequality and equality constraints, an efficient constraints handling technique is also required in the solution method. In this paper, we propose a modified constrained differential evolution algorithm (herein denoted as mCDE) that uses the self-adaptive control parameters [3], a mixture of modified mutations, and also includes the inversion operation, a modified selection and the elitism to be able to progress efficiently towards global solutions of problems (1).

The organization of this paper is as follows. Firstly, we describe the constraints handling techniques in Section 2 and then the modified constrained differential evolution is outlined in Section 3. Section 4 describes the experimental results and finally we draw the conclusions of this study in Section 5.

## 2 Constraints Handling Techniques

Stochastic methods have been primary developed for the global optimization of unconstrained problems. Extensions to the constrained problems then appear with the modification of some solution procedures. To deal with a constrained problem, a widely used approach is based on penalty functions where a penalty term is added to the objective function in order to penalize the constraint violation. This enable us to transform a constrained problem into a sequence of unconstrained subproblems. The penalty function method can be applied to any type of constraints, but the performance of penalty-type method is not always satisfactory because of choosing an appropriate penalty parameter. For this reason alternative constraints handling techniques have been proposed in the last decades. Three different techniques, usually used in population-based methods have been implemented and extensively tested in our proposed mCDE algorithm: a) the *feasibility and dominance rules*, b) the *stochastic ranking*, and c) the *global competitive ranking*. They are briefly described below. In a population-based solution method with $N$ candidate solutions $\boldsymbol{x}_i,\ i = 1, 2, \ldots, N$ at each generation, a common measure of infeasibility of an individual point $\boldsymbol{x}_i$ is the average measure of constraint violation given by

$$\zeta(\boldsymbol{x}_i) = \frac{1}{m}\left(\sum_{k=1}^{m_1} \max\{0, g_k(\boldsymbol{x}_i)\} + \sum_{l=1}^{m_2} |h_l(\boldsymbol{x}_i)|\right),$$

where $m = m_1 + m_2$ and $\zeta(\boldsymbol{x}_i)$ is a non-negative real-valued function, with $\zeta(\boldsymbol{x}_i) = 0$ if the point $\boldsymbol{x}_i$ is feasible.

### 2.1 Feasibility and Dominance Rules

Deb [6] proposed a constraints handling technique for population-based solution methods based on a set of rules that uses feasibility and dominance (FD) principles, as follows. First, the constraint violation $\zeta$ is calculated for all the individuals in a population. Then the objective function $f$ is evaluated only for feasible individuals. Two individual points are compared at a time, and the following criteria are always enforced:

a) any feasible point is preferred to any infeasible point;
b) between two feasible points, one having better objective function is preferred;
c) between two infeasible points, one having smaller constraint violation is preferred.

In this case, the fitness of each individual point $\boldsymbol{x}_i$ is calculated as follows

$$\Phi_{\text{FD}}(\boldsymbol{x}_i) = \begin{cases} f(\boldsymbol{x}_i) & \text{if } \boldsymbol{x}_i \text{ is feasible} \\ f_{\max,f} + \zeta(\boldsymbol{x}_i) & \text{otherwise,} \end{cases} \tag{2}$$

where $f_{\max,f}$ is the objective function of the worst feasible solution in the population. When all individuals are infeasible then its value is set to zero. This fitness function is used to choose the best individual point in a population.

### 2.2 Stochastic Ranking

Runarsson and Yao [21] first proposed the stochastic ranking (SR) for the constrained nonlinear optimization problems. This is a bubble-sort-like algorithm to give ranks to individuals in a population stochastically. In this ranking method, two adjacent individual points are compared and given ranks and swapped. The algorithm is halt if there is no swap. Individuals are ranked primarily based on their constraint violations. The objective function values are then considered if: i) individuals are feasible, or ii) a uniform random number between 0 and 1 is less than or equal to $P_f$. The probability $P_f$ is used only for comparisons of the objective function in the infeasible region of the search space. Such ranking ensures that good feasible solutions as well as promising infeasible ones are ranked in the top of the population.

In our implementation of the stochastic ranking method in the modified constrained differential evolution, each individual point $\boldsymbol{x}_i$ is evaluated according to the fitness function

$$\Phi_{\text{SR}}(\boldsymbol{x}_i) = \frac{I_i - 1}{N - 1}, \tag{3}$$

where $I_i$ represents the rank of point $\boldsymbol{x}_i$. From (3), the fitness of an individual point having the highest rank will be 0 and that with the lowest rank will be 1. The best individual point in a population has the lowest fitness value.

### 2.3 Global Competitive Ranking

Runarsson and Yao [22] proposed another constraints handling technique in order to strike the right balance between the objective function and the constraint violation. This method is called global competitive ranking (GR), where an individual point is ranked by comparing it against all other members in the population.

In this ranking process, after calculating $f$ and $\zeta$ for all the individuals, $f$ and $\zeta$ are sorted separately in ascending order (since we consider the minimization problem) and given ranks. Special consideration is given to the *tied individuals*. In case of tied individuals the same higher rank will be given. For example, in these eight individuals, already in ascending order, $\langle 6, (5, 8), 1, (2, 4, 7), 3\rangle$ (individuals in parentheses have same value) the corresponding ranks are $I(6) = 1, I(5) = I(8) = 2, I(1) = 4,$

$I(2) = I(4) = I(7) = 5, I(3) = 8$. After giving ranks to all the individuals based on the objective function $f$ and the constraint violation $\zeta$, separately, the fitness function of each individual point $\boldsymbol{x}_i$ is calculated by

$$\Phi_{\text{GR}}(\boldsymbol{x}_i) = P_f \frac{I_{i,f} - 1}{N - 1} + (1 - P_f) \frac{I_{i,\zeta} - 1}{N - 1}, \tag{4}$$

where $I_{i,f}$ and $I_{i,\zeta}$ are the ranks of point $\boldsymbol{x}_i$ based on the objective function and the constraint violation, respectively. $P_f$ indicates the probability that the fitness is calculated based on the rank of objective function. It is clear from the above that $P_f$ can be used easily to bias the calculation of fitness according to the objective function or the constraint violation. The probability should take a value $0.0 < P_f < 0.5$ in order to guarantee that a feasible solution may be found. From (4), the fitness of an individual point is a value between 0 and 1, and the best individual point in a population has the lowest fitness value.

## 3 Modified Constrained Differential Evolution

The population-based differential evolution algorithm [24] has become popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is easy to understand. DE is a floating point encoding that creates a new candidate point by adding the weighted difference between two individuals to a third one in the population. This operation is called mutation. The mutant point's components are then mixed with the components of target point to yield the trial point. This mixing of components is referred to as crossover. In selection, a trial point replaces a target point for the next generation only if it is considered an equal or better point. In unconstrained optimization, the selection operation relies on the objective function. DE has three control parameters: amplification factor of differential variation $F$, crossover control parameter $Cr$, and population size $N$.

It is not an easy task to set the appropriate control parameters since these depend on the nature and size of the optimization problems. Hence, the adaptive control parameters ought to be implemented. Brest et al. [3] proposed the self-adaptive control parameters for DE when solving global optimization problems with simple bounds. In most original DE, three points are chosen randomly for mutation and the base point is then chosen at random within the three. This has an exploratory effect but it slows down the convergence of DE. Kaelo and Ali [14] proposed a modified mutation for differential evolution.

The herein presented modified constrained differential evolution algorithm - mCDE - for constrained nonlinear optimization problems (1) includes:

1) the self-adaptive control parameters $F$ and $Cr$, as proposed by Brest et al.;
2) a modified mutation that mixes the modification proposed by Kaelo and Ali with the cyclical use of the overall best point as the base point;
3) the inversion operation;
4) a modified selection that is based on the fitness of individuals;
5) the elitism.

The modification in mutation allows mCDE to keep the exploration as well as enhance the exploitation around the overall best point. In modified selection of mCDE, we implement and test the three different techniques described so far for calculating the fitness of individuals that are capable to handle the constraints of problems (1). The modified constrained differential evolution is outlined below.

The target point of mCDE, at iteration/generation $z$, is defined by $\boldsymbol{x}_{i,z} = (x_{i1,z}, x_{i2,z}, \ldots, x_{in,z})$, where $n$ is the number of variables of the optimization problem and $i = 1, 2, \ldots, N$. The initial population is chosen randomly and should cover the entire component spaces.

**Self-adaptive Control Parameters.** In mCDE, we use the self-adaptive control parameters for $F$ and $Cr$, as proposed by Brest et al. [3] by generating a different set $(F_i, Cr_i)$ for each point $\boldsymbol{x}_i$ in the population. The new control parameters for the next generation $F_{i,z+1}$ and $Cr_{i,z+1}$ are calculated by

$$F_{i,z+1} = \begin{cases} F_l + \lambda_1 \times F_u & \text{if } \lambda_2 < \tau_1 \\ F_{i,z} & \text{otherwise} \end{cases}$$
$$Cr_{i,z+1} = \begin{cases} \lambda_3 & \text{if } \lambda_4 < \tau_2 \\ Cr_{i,z} & \text{otherwise,} \end{cases}$$

where $\lambda_k \sim \mathrm{U}[0,1], k = 1, \ldots, 4$ and $0 < \tau_1, \tau_2 < 1$ represent the probabilities to adjust parameters $F_i$ and $Cr_i$, respectively, and $0 < F_l < F_u < 1$, so the new $F_{i,z+1}$ takes a value from $(0, 1)$ in a random manner. The new $Cr_{i,z+1}$ takes a value from $[0, 1]$. $F_{i,z+1}$ and $Cr_{i,z+1}$ are obtained before the mutation is performed. So, they influence the mutation, crossover and selection operations of the new point $\boldsymbol{x}_{i,z+1}$.

**Modified Mutation.** In mCDE, this is a mixture of two different types of mutation operations. We use the mutation proposed in [14]. After choosing three points randomly, the best point among three based on the fitness function is selected for the base point and the remaining two points are used as differential variation, i.e., for each target point $\boldsymbol{x}_{i,z}$, a mutant point is created according to

$$\boldsymbol{v}_{i,z+1} = \boldsymbol{x}_{r_3,z} + F_{i,z+1}(\boldsymbol{x}_{r_1,z} - \boldsymbol{x}_{r_2,z}), \tag{5}$$

where $r_1, r_2, r_3$ are randomly chosen from the set $\{1, 2, \ldots, N\}$, mutually different and different from the running index $i$ and $r_3$ is the index with the best fitness (among the three points). This modification has a local effect when the points in the population form a cluster around the global minimizer.

Furthermore, at every $B$ generations, the best point found so far is used as the base point and two randomly chosen points are used as differential variation, i.e.,

$$\boldsymbol{v}_{i,z+1} = \boldsymbol{x}_{\text{best}} + F_{i,z+1}(\boldsymbol{x}_{r_1,z} - \boldsymbol{x}_{r_2,z}). \tag{6}$$

This modified mutation allows mCDE to maintain its exploratory feature as well as at the same time to exploit the region around the best individual point in the population expediting the convergence.

**Crossover.** In order to increase the diversity of the mutant points' components, crossover is introduced. To this end, the crossover point $\boldsymbol{u}_{i,z+1}$ is formed, where

$$u_{ij,z+1} = \begin{cases} v_{ij,z+1} & \text{if } (r_j \leq Cr_{i,z+1}) \text{ or } j = s_i \\ x_{ij,z} & \text{if } (r_j > Cr_{i,z+1}) \text{ and } j \neq s_i. \end{cases} \tag{7}$$

In (7), $r_j \sim \mathrm{U}[0,1]$ performs the mixing of $j$th component of points, $s_i$ is randomly chosen from the set $\{1, 2, \ldots, n\}$ and ensures that $\boldsymbol{u}_{i,z+1}$ gets at least one component from $\boldsymbol{v}_{i,z+1}$.

**Inversion.** Since in mCDE, a point has $n$-dimensional real components, inversion [13] can easily be applicable. With the inversion probability ($p_{\text{inv}} \in [0,1]$), two positions are chosen on the point $\boldsymbol{u}_i$, the point is cut at those positions, and the cut segment is reversed and reinserted back into the point to create the trial point $\boldsymbol{u}'_i$. In practice, mCDE with the inversion has been shown to give better results than those obtained without the inversion. An illustrative example of inversion is shown in Figure 1.

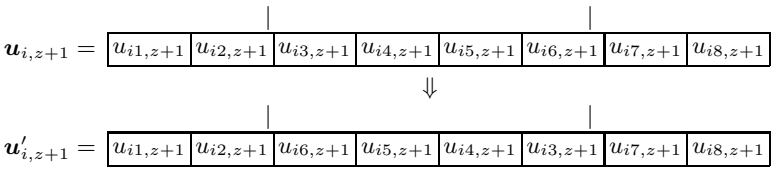



**Fig. 1.** Inversion used in mCDE

**Bounds Check.** When creating the mutant point and when the inversion operation is performed, some components can be created outside the bound constraints. So, in mCDE after inversion the bounds of each component should be checked with the following projection of bounds:

$$u'_{ij,z+1} = \begin{cases} l_j & \text{if } u'_{ij,z+1} < l_j \\ u_j & \text{if } u'_{ij,z+1} > u_j \\ u'_{ij,z+1} & \text{otherwise.} \end{cases}$$

**Modified Selection.** In original DE, the target and the trial points are compared based on their corresponding objective function value to decide which point becomes a member of next generation, that is if the trial point's objective function is less than or equal to the that of target point, then the trial point will be the target point for the next generation.

In this paper, for constrained nonlinear optimization problems, we propose a modified selection based on one of the fitness functions of individuals discussed so far (Section 2). When using the stochastic ranking technique, all the target points at generation $z$ and trial points at generation $z+1$ are ranked together and their corresponding fitness $\Phi_{\text{SR}}$ are calculated. Then the modified selection is performed, i.e., the trial and the target points are compared to decide which will be the new target points for the next generation based on their calculated fitness by the following way

$$\boldsymbol{x}_{i,z+1} = \begin{cases} \boldsymbol{u}'_{i,z+1} & \text{if } \Phi_{\text{SR}}(\boldsymbol{u}'_{i,z+1}) \leq \Phi_{\text{SR}}(\boldsymbol{x}_{i,z}) \\ \boldsymbol{x}_{i,z} & \text{otherwise.} \end{cases}$$

A similar procedure is performed when the global competitive ranking technique is implemented. After performing selection in mCDE, the best point is chosen in the current generation based on the lowest fitness of the target points.

On the other hand, when using the feasibility and dominance rules, the trial and the target points are compared based on the three feasibility and dominance principles to decide which will be the new target points for the next generation. After performing selection, the fitness function $\Phi_{\text{FD}}$ for all the target points are calculated, and the best point based on the lowest fitness function in the current generation is chosen. We remark that this point is the overall best point in the entire generations so far.

**Elitism.** The elitism is also performed to keep the best point found so far in the entire generations. The elitism aims at preserving in the entire generations the individual point that, with the constraint violation 0 or smaller than others, has the smallest objective function. This is required when either the stochastic ranking or the global competitive ranking is used to calculate fitness of individuals. We remark that in these two techniques, fitness values of individuals are calculated at every generation based on their corresponding ranks. Thus, the fitness of best individual point (based on the objective function and the constraint violation) may not be the lowest one.

**Termination Criterion.** Let $G_{\max}$ be the maximum number of generations. If $f_{\text{best}}$ is the best objective function value found so far and $f_{\text{opt}}$ is the known optimal value, then our proposed mCDE algorithm terminates if $z > G_{\max}$ or $|f_{\text{best}} - f_{\text{opt}}| \leq \eta$, for a small positive number $\eta$.

**mCDE Algorithm.** The algorithm of the herein proposed modified constrained differential evolution for constrained nonlinear optimization problems is described in Algorithm 1.

## 4 Experimental Results

We code mCDE in C with AMPL [11] interfacing and compile with Microsoft Visual Studio 9.0 compiler in a PC having 2.5 GHz Intel Core 2 Duo processor and 4 GB RAM. We set $N = \min(100, 10n)$, $B = 10$, $P_f = 0.45$, $\tau_1 = \tau_2 = 0.1$, $F_l = 0.1$, $F_u = 0.9$, $p_{\text{inv}} = 0.05$ and $\eta = 10^{-6}$. We consider 13 benchmark constrained nonlinear optimization problems [21]. Their characteristics are outlined in Table 1. For these problems, we consider an individual point as a feasible one if $\zeta(\boldsymbol{x}) \leq \delta$, where $\delta$ is a very small positive number. Here we set $\delta = 10^{-8}$.

At first, we compare the three different variants of mCDE: a) mCDE_FD (based on *feasibility and dominance rules*), b) mCDE_SR (based on *stochastic ranking*) and c) mCDE_GR (based on *global competitive ranking*) using the performance profiles as described in [7]. A comparison with other solution methods available in literature is also included. To be able to fairly compare the variants mCDE_SR and mCDE_GR with the variant mCDE_FD, after the modified selection step of the algorithm, the fitness function was recalculated using (2) so that the best and the worst target points in the population are identified according to the objective function and constraint violation.

**Algorithm 1.** mCDE algorithm

**Require:** $N$, $G_{\max}$, $B$, $P_f$, $F_l$, $F_u$, $\tau_1$, $\tau_2$, $p_{\text{inv}}$, and $\eta$.
Set $z = 1$. Randomly initialize $F_{i,1}$, $Cr_{i,1}$ and the population $\boldsymbol{x}_{i,1}\ \forall i = 1, \ldots, N$.
Calculate the fitness $\Phi(\boldsymbol{x}_{i,1})$, for all $i$, and perform elitism to choose $f_{\text{best}}$ and $\boldsymbol{x}_{\text{best}}$.
**while** the termination criterion is not met **do**
  **for** $i = 1$ to $N$ **do**
    Compute the control parameters $F_{i,z+1}$ and $Cr_{i,z+1}$.
    **if** MOD$(z + 1, B) = 0$ **then**
      Compute the mutant point $\boldsymbol{v}_{i,z+1}$ using (6).
    **else**
      Compute the mutant point $\boldsymbol{v}_{i,z+1}$ using (5).
    **end if**
    Perform the crossover to make point $\boldsymbol{u}_{i,z+1}$.
    **if** $\gamma \sim \text{U}[0, 1] \leq p_{\text{inv}}$ **then**
      Perform inversion to make the trial point $\boldsymbol{u}'_{i,z+1}$.
    **end if**
    Check the bounds of the trial point.
  **end for**
  Calculate the fitness $\Phi(\boldsymbol{x}_{i,z})$, $\Phi(\boldsymbol{u}'_{i,z+1})$, for all $i$.
  Perform modified selection.
  Perform elitism to choose $f_{\text{best}}$ and $\boldsymbol{x}_{\text{best}}$. Set $z = z + 1$.
**end while**

**Table 1.** Characteristics of the test problems

| Prob. | Type of $f$ | $f_{\text{opt}}$ | $n$ | $m_1$ | $m_2$ | $m$ |
|---|---|---|---|---|---|---|
| g01 | quadratic | -15.0000 | 13 | 9 | 0 | 9 |
| g02 | general | -0.8036 | 20 | 2 | 0 | 2 |
| g03 | polynomial | -1.0005 | 10 | 0 | 1 | 1 |
| g04 | quadratic | -30665.5387 | 5 | 6 | 0 | 6 |
| g05 | cubic | 5126.4967 | 4 | 2 | 3 | 5 |
| g06 | cubic | -6961.8139 | 2 | 2 | 0 | 2 |
| g07 | quadratic | 24.3062 | 10 | 8 | 0 | 8 |
| g08 | general | -0.0958 | 2 | 2 | 0 | 2 |
| g09 | general | 680.6301 | 7 | 4 | 0 | 4 |
| g10 | linear | 7049.2480 | 8 | 6 | 0 | 6 |
| g11 | quadratic | 0.7499 | 2 | 0 | 1 | 1 |
| g12 | quadratic | -1.0000 | 3 | 1 | 0 | 1 |
| g13 | general | 0.0539 | 5 | 0 | 3 | 3 |

### 4.1 Comparison by Performance Profiles

We ran the three variants of mCDE for 30 times and recorded the results. We used different $G_{\max}$ for the 13 problems, but used the same value for all the variants in comparison. The performance profiles proposed by Dolan and Moré [7] are the graphical representation of the performance ratio of different solvers/variants when solving a set of test problems. The profiles plot the cumulative distribution function of the performance ratio obtained from an appropriate performance metric.

Let $\mathcal{P}$ be the set of test problems and $\mathcal{S}$ be the set of all variants of mCDE in comparison. In our comparative study, the metric, $m_{(p,s)}$, found by variant $s \in \mathcal{S}$ on problem $p \in \mathcal{P}$, measures the average improvement of the objective function values, based on a relative scaled distance to the optimal objective function value $f_{\text{opt}}$ [1], defined by

$$m_{(p,s)} = \frac{f_{\text{avg}(p,s)} - f_{\text{opt}}}{f_w - f_{\text{opt}}}, \tag{8}$$

where $f_{\text{avg}(p,s)}$ is the average of the best solutions obtained by the variant $s$ on problem $p$ after 30 runs and $f_w$ is the worst objective function value of problem $p$ after 30 runs among all variants. The performance ratio is thus defined by

$$r_{(p,s)} = \begin{cases} 1 + m_{(p,s)} - q & \text{if } q \leq 10^{-5} \\ \dfrac{m_{(p,s)}}{q} & \text{otherwise,} \end{cases}$$

where $q = \min\{m_{(p,s)} : s \in \mathcal{S}\}$.

The fraction of problems for which variant $s$ has a performance ratio $r_{(p,s)}$ within a factor $\tau \in \mathbb{R}$, is given by $\rho_s(\tau) = (n_{P_\tau})/(n_P)$, where $n_{P_\tau}$ is the number of problems in $\mathcal{P}$ with $r_{(p,s)} \leq \tau$ and $n_P$ is the total number of problems in $\mathcal{P}$. $\rho_s(\tau)$ is the probability (for $s \in \mathcal{S}$) that the performance ratio $r_{(p,s)}$ is within a factor $\tau$ of the best possible ratio.

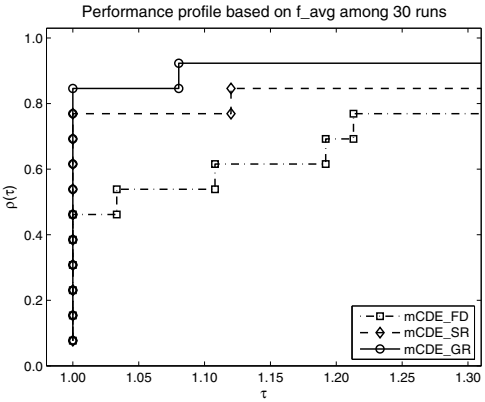



**Fig. 2.** Performance profile based on the average improvement of function values

Figure 2 shows the profiles of the performance metric in (8). If we are only interested in knowing which variant is the most efficient, in the sense that it reaches the best solutions mostly, we compare the values of $\rho_s(1)$, for $s \in \mathcal{S}$, and find the highest value which is the probability that the variant will win over the remaining ones. However, to assess the robustness of variants, we compare the values of $\rho_s(\tau)$ for large values of $\tau$. The variant with the largest probability is the most robust one. In this figure it is shown that the variant mCDE_GR wins over the other two variants. Hence, the comparison with other solution methods available in literature uses the variant mCDE_GR, hereafter denoted by mCDE.

### 4.2 Comparison with Other Methods

We also compare mCDE with the stochastic ranking, SRES, presented in [21] and the global competitive ranking, GRES, presented in [22]. The authors proposed these techniques based on a $(30, 200)$ evolution strategy. An adaptive penalty scheme for constraints handling with dynamic use of variants of differential evolution (DUVDE) [23] is also used in this comparison. According to [21,22], we set $G_{\max} = 1750$ for all problems except problem g12, where $G_{\max} = 175$. Here, we aim to get a solution within $0.001\%$ of the known optimal solution $f_{\text{opt}}$.

Tables 2 and 3 show the experimental results of the 13 problems, where '$f_{\text{best}}$' is the best of the objective function values obtained among 30 runs, '$f_{\text{avg}}$' is the average of the best objective function values and '$\text{std}_f$' means the standard deviation of objective function values among 30 runs. The results from SRES, GRES and DUVDE are taken from their corresponding literatures. In mCDE, we use the population size $N$ dependent on the dimension of the test problem and in DUVDE the authors used the population size $50$ and the maximum number of generations $3684$ for all the test problems. Problems g12 and g13 were not considered with DUVDE. From Tables 2 and 3 we may conclude that for most of the problems, and with respect to all measures of comparison, mCDE performs rather well when compare with SRES, GRES and DUVDE.

### 4.3 Solving Mixed-Integer Design Problems

We now consider four engineering design problems to show the effectiveness of our proposed method when solving problems with discrete, integer and continuous variables. Engineering problems with mixed-integer design variables are quite common. Therefore, the convergence behavior of our proposed mCDE when handling discrete and integer variables is to be provided.

For discrete variables, we randomly generate values from an appropriate discrete set in the two procedures: initialization and mutation.

For integer variables, a simple heuristic that relies on the rounding off to the nearest integer at evaluation stages is implemented.

We considered four well-known engineering design problems. Since the optimal solutions of the considered problems are unknown, we used only $G_{\max}$ for the termination criterion and $\delta = 0$. For each problem 30 independent runs were carried out.

**Spring Design**

This is a real world optimization problem involving discrete, integer and continuous design variables. The objective is to minimize the volume of a compression spring under static loading. The design problem has three variables and eight inequality constraints [15], where $x_1$, the wire diameter, is taken from a set of discrete values and $x_3$, the number of coils, is integer. We set $G_{\max} = 500$. We compare the obtained results from our mCDE with DE [15] and ranking selection-based particle swarm optimization, RPSO [26]. The comparative results are shown in Table 4.

**Table 2.** Experimental results from SRES and GRES

| Prob. | SRES | | | GRES | | |
|---|---|---|---|---|---|---|
| | $f_{\text{best}}$ | $f_{\text{avg}}$ | $\text{std}_f$ | $f_{\text{best}}$ | $f_{\text{avg}}$ | $\text{std}_f$ |
| g01 | -15.0000 | -15.0000 | 0.00E+00 | -15.0000 | – | 0.00E+00 |
| g02 | -0.8035 | -0.7820 | 2.00E-02 | -0.8035 | – | 1.70E-02 |
| g03 | -1.0000 | -1.0000 | 1.90E-04 | -1.0000 | – | 2.60E-05 |
| g04 | -30665.5390 | -30665.5390 | 2.00E-05 | -30665.5390 | – | 5.40E-01 |
| g05 | 5126.4970 | 5128.8810 | 3.50E+00 | 5126.4970 | – | 1.10E+00 |
| g06 | -6961.8140 | -6875.9400 | 1.60E+02 | -6943.5600 | – | 2.90E+02 |
| g07 | 24.3070 | 24.3740 | 6.60E-02 | 24.3080 | – | 1.10E-01 |
| g08 | -0.0958 | -0.0958 | 2.60E-17 | -0.0958 | – | 2.60E-17 |
| g09 | 680.6300 | 680.6560 | 3.40E-02 | 680.6310 | – | 5.80E-02 |
| g10 | 7054.3160 | 7559.1920 | 5.30E+02 | * | – | * |
| g11 | 0.7500 | 0.7500 | 8.00E-05 | 0.7500 | – | 7.20E-05 |
| g12 | -1.0000 | -1.0000 | 0.00E+00 | -1.0000 | – | 0.00E+00 |
| g13 | 0.0539 | 0.0675 | 3.10E-02 | 0.0539 | – | 1.30E-04 |

(–) not available; (*) not solved

**Table 3.** Experimental results from DUVDE and mCDE

| Prob. | DUVDE | | | mCDE | | |
|---|---|---|---|---|---|---|
| | $f_{\text{best}}$ | $f_{\text{avg}}$ | $\text{std}_f$ | $f_{\text{best}}$ | $f_{\text{avg}}$ | $\text{std}_f$ |
| g01 | -15.0000 | -12.5000 | 2.37E+00 | -15.0000 | -15.0000 | 1.16E-06 |
| g02 | -0.8036 | -0.7688 | 3.57E-02 | -0.8036 | -0.8007 | 4.95E-03 |
| g03 | -1.0000 | -0.2015 | 3.45E-01 | -1.0000 | -1.0000 | 3.90E-05 |
| g04 | -30665.5000 | -30665.5000 | 0.00E+00 | -30665.5387 | -30665.5387 | 2.38E-05 |
| g05 | 5126.4965 | 5126.4965 | 0.00E+00 | 5126.4978 | 5126.4979 | 1.83E-04 |
| g06 | -6961.8000 | -6961.8000 | 0.00E+00 | -6961.8161 | -6950.5609 | 6.16E+01 |
| g07 | 24.3060 | 30.4040 | 2.16E+01 | 24.2316 | 24.2317 | 7.44E-05 |
| g08 | -0.0958 | -0.0958 | 0.00E+00 | -0.0958 | -0.0958 | 2.71E-06 |
| g09 | 680.6300 | 680.6300 | 3.00E-05 | 680.6301 | 680.6301 | 1.38E-06 |
| g10 | 7049.2500 | 7351.1700 | 5.26E+02 | 7049.2533 | 7053.3441 | 6.99E+00 |
| g11 | 0.7500 | 0.9875 | 5.59E-02 | 0.7500 | 0.7506 | 3.11E-03 |
| g12 | † | † | † | -1.0000 | -1.0000 | 2.33E-06 |
| g13 | † | † | † | 0.0539 | 0.0539 | 3.53E-17 |

(†) not considered

**Table 4.** Comparative results of spring design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $f_{\text{best}}$ | $G_{\max}$ |
|---|---|---|---|---|---|
| DE | 0.283 | 1.223 | 9 | 2.65856 | 650 |
| RPSO | 0.283 | 1.223 | 9 | 2.65856 | 750 |
| mCDE | 0.283 | 1.223 | 9 | 2.65856 | 500 |

**Pressure Vessel Design**

The design of a cylindrical pressure vessel with both ends capped with a hemispherical head is to minimize the total cost of fabrication [5,25]. The problem has four design variables and four inequality constraints. This is a mixed variables problem where

$x_1$, the shell thickness, and $x_2$, the head thickness, are discrete of integer multiples of $0.0625$ inch., and other two are continuous. We set $G_{\max} = 1000$. The comparative results from mCDE with hybrid genetic algorithm, HGA [5] and cost-effective particle swarm optimization, CPSO [25] are shown in Table 5.

**Table 5.** Comparative results of pressure vessel design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f_{\text{best}}$ | $G_{\max}$ |
|---|---|---|---|---|---|---|
| HGA | 0.8125 | 0.4375 | 42.0870 | 176.7791 | 6061.123 | 5000 |
| CPSO | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | 10000 |
| mCDE | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.714 | 1000 |

**Speed Reducer Design**

The weight of the speed reducer is to be minimized subject to the constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stress in the shafts as described in [5,25]. There are seven variables and 11 inequality constraints. This is a mixed variables problem, where $x_3$ is integer (number of teeth) and the others are continuous. We set $G_{\max} = 500$. The comparative results among mCDE, HGA and CPSO are shown in Table 6.

**Table 6.** Comparative results of speed reducer design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $f_{\text{best}}$ | $G_{\max}$ |
|---|---|---|---|---|---|---|---|---|---|
| HGA | 3.5 | 0.7 | 17 | 7.3 | 7.7153 | 3.3502 | 5.2867 | 2994.342 | 5000 |
| CPSO | 3.5 | 0.7 | 17 | 7.3 | 7.8000 | 3.3502 | 5.2867 | 2996.348 | 10000 |
| mCDE | 3.5 | 0.7 | 17 | 7.3 | 7.7153 | 3.3502 | 5.2867 | 2994.342 | 500 |

**Stepped Cantilever Beam Design**

The design variables of a stepped cantilever beam are the widths and depths of rectangular cross-sections. The objective of this problem is to minimize the volume of the beam under static loading [27]. This is a mixed-integer design problem having 10 variables and 11 constraints, where $x_1$ and $x_2$ are integer, $x_3$ to $x_6$ are discrete and the remaining are continuous. We set $G_{\max} = 1000$. The comparative results from our mCDE with genetic algorithm, GA [9], and linearization techniques method [27] are shown in Table 7.

**Table 7.** Comparative results of stepped cantilever beam design problem

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $f_{\text{best}}$ | $G_{\max}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GA | 3 | 60 | 3.1 | 55 | 2.6 | 50 | 2.2700 | 45.2500 | 1.7500 | 35.0000 | 64447.00 | – |
| in [27] | 3 | 60 | 3.1 | 55 | 2.6 | 50 | 2.2045 | 44.0907 | 1.7498 | 34.9960 | 63892.56 | – |
| mCDE | 3 | 60 | 3.1 | 55 | 2.6 | 50 | 2.2055 | 44.0855 | 1.7502 | 34.9924 | 63897.45 | 1000 |

(–) not available

From the Tables 4 - 7, it is found that mCDE is competitive with other solution methods when solving engineering design problems.

From the above discussion it is clear that the herein presented modified constrained differential evolution algorithm, based on global competitive ranking for constraints handling, is rather effective when converging to global solutions.

## 5 Conclusions

In this paper, to make the DE methodology more efficient to handle the constraints, a modified constrained differential evolution algorithm (mCDE) is proposed. The modifications focus on the self-adaptive control parameters, a modified mutation, a modified selection and the elitism. Inversion has also been implemented in the proposed mCDE.

The modifications that mostly influence the efficiency of the algorithm are the following: a) the mixed modified mutation, aiming at exploring both the entire search space (when using the mutation as in [14]) and the neighborhood of the best point found so far (when using the best point as the base point cyclically); b) the modified selection, to handle the constraints effectively, that uses a fitness function based on the global competitive ranking technique. In this technique, fitness of all target and trial points are calculated all together after giving them ranks based on the objective function and the constraint violation separately, for competing in modified selection to decide which points win for the next generation. This technique seems to have stricken the right balance between the objective function and the constraint violation for obtaining a global solution while satisfying the constraints.

To test the effectiveness of our mCDE, 13 benchmark constrained nonlinear optimization problems have been considered. These problems have also been solved with the stochastic ranking and the feasibility and dominance rules techniques and a comparison has been carried out based on their performance profiles. We could observe that the performance of the mCDE with the global competitive ranking is relatively better than the other two in comparison. The numerical experiments also show that mCDE is rather competitive when compared with the other solution methods available in literature. Further, it is also found that the mCDE is competitive with other known heuristics when solving mixed-integer engineering design problems.

**Acknowledgements.** The first author acknowledges Ciência 2007 of FCT, Fundação para a Ciência e a Tecnologia (Foundation for Science and Technology), Portugal for the financial support under fellowship grant: C2007-UMINHO-ALGORITMI-04. The second author acknowledges FEDER COMPETE, Programa Operacional Fatores de Competitividade (Operational Programme Thematic Factors of Competitiveness) and FCT for the financial support under project grant: FCOMP-01-0124-FEDER-022674.

## References

1. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Glob. Optim. 31, 635–672 (2005)
2. Barbosa, H.J.C., Lemonge, A.C.C.: A new adaptive penalty scheme for genetic algorithms. Inf. Sci. 156, 215–251 (2003)
3. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10, 646–657 (2006)

4. Coello Coello, C.A.: Constraint-handling using an evolutionary multiobjective optimization technique. Civ. Eng. Environ. Syst. 17, 319–346 (2000)
5. Coello Coello, C.A., Cortés, N.C.: Hybridizing a genetic algorithm with an artificial immune system for global optimization. Eng. Optim. 36, 607–634 (2004)
6. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. 186, 311–338 (2000)
7. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. 91, 201–213 (2002)
8. Dong, Y., Tang, J., Xu, B., Wang, D.: An application of swarm optimization to nonlinear programming. Comput. Math. Appl. 49, 1655–1668 (2005)
9. Erbatur, F., Hasançebi, O., Tütüncü, İ., Kılıç, H.: Optimal design of planar and space structures with genetic algorithms. Comput. Struct. 75, 209–224 (2000)
10. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. 91, 239–269 (2002)
11. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: A modeling language for mathematical programming. Boyd & Fraser Publishing Co., Massachusetts (1993)
12. Hedar, A.R., Fukushima, M.: Derivative-free filter simulated annealing method for constrained continuous global optimization. J. Glob. Optim. 35, 521–549 (2006)
13. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
14. Kaelo, P., Ali, M.M.: A numerical study of some modified differential evolution algorithms. Eur. J. Oper. Res. 169, 1176–1184 (2006)
15. Lampinen, J., Zelinka, I.: Mixed integer-discrete-continuous optimization by differential evolution. In: Proceedings of the 5th International Conference on Soft Computing, pp. 71–76 (1999)
16. Ray, T., Tai, K.: An evolutionary algorithm with a multilevel pairing strategy for single and multiobjective optimization. Found. Comput. Decis. Sci. 26, 75–98 (2001)
17. Ray, T., Liew, K.M.: Society and civilization: An optimization algorithm based on the simulation of social behavior. IEEE Trans. Evol. Comput. 7, 386–396 (2003)
18. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Feasibility and Dominance Rules in the Electromagnetism-Like Algorithm for Constrained Global Optimization. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part II. LNCS, vol. 5073, pp. 768–783. Springer, Heidelberg (2008)
19. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Self adaptive penalties in the electromagnetism-like algorithm for constrained global optimization problems. In: Proceedings of the 8th World Congress on Structural and Multidisciplinary Optimization, pp. 1–10 (2009)
20. Rocha, A.M.A.C., Martins, T.F.M.C., Fernandes, E.M.G.P.: An augmented Lagrangian fish swarm based method for global optimization. J. Comput. Appl. Math. 235, 4611–4620 (2011)
21. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evol. Comput. 4, 284–294 (2000)
22. Runarsson, T.P., Yao, X.: Constrained evolutionary optimization – the penalty function approach. In: Sarker, et al. (eds.) Evolutionary Optimization. International Series in Operations Research and Management Science, vol. 48, pp. 87–113. Springer, New York (2003)
23. Silva, E.K., Barbosa, H.J.C., Lemonge, A.C.C.: An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization. Optim. Eng. 12, 31–54 (2011)
24. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 11, 341–359 (1997)

25. Tomassetti, G.: A cost-effective algorithm for the solution of engineering problems with particle swarm optimization. Eng. Optim. 42(5), 471–495 (2010)
26. Wang, J., Yin, Z.: A ranking selection-based particle swarm optimizer for engineering design optimization problems. Struct. Multidisc. Optim. 37(2), 131–147 (2008)
27. Wang, P.-C., Tsai, J.-F.: Global optimization of mixed-integer nonlinear programming for engineering design problems. In: Proceedings of the International Conference on System Science and Engineering, pp. 255–259 (2011)
28. Zahara, E., Hu, C.-H.: Solving constrained optimization problems with hybrid particle swarm optimization. Eng. Optim. 40, 1031–1049 (2008)

# Dynamical Modeling and Parameter Identification of Seismic Isolation Systems by Evolution Strategies

Anastasia Athanasiou[1], Matteo De Felice[2], Giuseppe Oliveto[1], and Pietro S. Oliveto[3]

[1] Department of Civil and Environmental Engineering, University of Catania, Catania, Italy
[2] Energy and Environment Modeling Technical Unit, ENEA, Rome, Italy
[3] School of Computer Science, The University of Birmingham, Birmingham, U.K.
{athanasiou,golive}@dica.unict.it,
matteo.defelice@enea.it, P.S.Oliveto@cs.bham.ac.uk

**Abstract.** An application of Evolution Strategies (ESs) to the dynamic identification of hybrid seismic isolation systems is presented. It is shown how ESs are highly effective for the optimisation of the test problem defined in previous work for methodology validation. The acceleration records of a number of dynamic tests performed on a seismically isolated building are used as reference data for the parameter identification. The application of CMA-ES to a previously existing model considerably improves previous results but at the same time reveals limitations of the model. To investigate the problem three new mechanical models with higher number of parameters are developed. The application of CMA-ES to the best designed model allows improvements of up to 79% compared to the solutions previously available in literature.

**Keywords:** Earthquake engineering, Structural system identification, Evolution strategies, CMA-ES, Real-world applications.

## 1 Introduction

Structural engineering is a special technological field dealing with the analysis and design of engineering structures that must resist internal and/or external loads. Such structures may be integral parts of buildings, bridges, dams, ship hulls, aircraft, engines and so on. The design of such structures is an optimisation process by which the resistance capacity of the system is made to meet the demands posed to it by the environment. This process is based on the satisfaction of the basic design inequality by which the capacity must be no lower than the demand. While the capacity can be established by the engineer at each step of the design process, the demand depends both on the characteristics of the system itself and on its interaction with the surrounding environment.

The evaluation of the demands requires the simulation of the behaviour of the structural system (i.e., the *response*) under service and/or extreme loading conditions (e.g., earthquakes, tornadoes, turbulence etc). Such simulations require the construction of mechanical models which enable the prediction of the system's behaviour. Usually a mechanical model is described by a system of linear or non-linear differential equations and a set of physical parameters. While the system of differential equations is derived from first principles in mechanics, the physical parameters are derived from laboratory tests on materials and/or on structural parts of the system.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 101–118.
DOI: 10.1007/978-3-642-35638-4_8 

Structural identification can serve the dual purpose of establishing whether a given model is suitable to describe the behaviour of a structural system or to verify that the physical parameters fed into a reliable model correspond to the characteristics of the actual materials used in the construction of the system.

Structural identification finds applications in virtually every field of structural engineering. In this paper interest is focused on an application in earthquake engineering. As already mentioned, structural identification requires on one hand some kind of excitation and on the other hand the recording of the response of the structure to the given excitation.

Base isolation is a modern system for the protection of buildings and other constructions against earthquake excitations and works on the principle of decoupling the motion of the ground from that of the building. Ideally the building should stay still while the ground moves beneath it. This is achieved by interposing a set of special bearings (i.e., *seismic isolators*) between the foundation and the superstructure.

Although the basic idea is simple and intuitive, has been known since ancient times and has been tried often in past centuries, successful applications have become possible only in the second half of the twentieth century when technological advancements have allowed cost effective seismic isolators to be constructed. Initially the applications were scanty and limited to important buildings in highly industrialized seismic countries like Japan, the US and European Union ones, but after the excellent performance of base isolated buildings during the 1994 Northridge (California, US) and 1995 Kobe (Japan) earthquakes, the use of base isolation has received a strong impulse worldwide for the seismic protection of new buildings and for the seismic retrofitting of existing ones. An appreciation of the momentum of research and practical applications in this specific scientific and technological field may be gathered from recent literature [12,23].

Given the low stiffness of the building structure due to the presence of the seismic isolators it is rather easy to displace (i.e., move) the building by pushing it at the base with suitable actuators (i.e., *hydraulic jacks*). This system has been used in a handful of applications around the world [6,7,15,22] including one in the town of Solarino in Eastern Sicily [17].

The Solarino building was tested by release of imposed displacements in July 2004 and accelerations were recorded at each floor. These recordings were used as response functions for the identification of the base isolation system [17].

An iterative procedure based on the least squares method was used in [17] for the identification. This required tedious calculations of gradients which were done approximately by means of an ingenious numerical procedure. Before applying the identification procedure to the experimental data, the same procedure was evaluated against a test problem for which the solution was known. Hence, the ability of the optimisation algorithm was assessed in the absence of measurement noise and with the guarantee that the function to be identified fits the model. The procedure, was then applied to the real data derived from the tests on the Solarino building.

Although, the authors of [17] are satisfied with their results, they conclude that a "need for improvement both in the models and testing procedures also emerges from the numerical applications and results obtained". In particular, finding the "best" algorithm

for the identification of such a kind of problem would provide an improvement on the state-of-the-art in the identification of building and structures from dynamic tests.

In this chapter the described problem is addressed by applying Evolutionary Algorithms (EAs) for the identification of structural engineering systems. Indeed, the first applications of evolution computations were directed towards parameter optimisation for civil engineering simulation models e.g., simulating stress and displacement patterns of structures under load, [21].

Firstly, the performance of well known evolutionary algorithms for numerical optimization (i.e., Evolution Strategies (ESs)) is evaluated on the same test problem considered in [17]. Several ESs are applied and their performance is compared amongst themselves and against the previous results obtained in [17]. It is shown that even simple ESs outperform the previously used methods, while state-of-the-art ones such as the CMA-ES, provide solutions improved by several orders of magnitude, practically the exact solution.

By applying efficient ESs to the real data from the Solarino experiments, further and convincing evidence is given of the limitations of the model for the identification of the base isolation system. Such limitations could not be as visible from the results obtained with the previously used optimisation methods. Finally, new improved models designed to overcome the limitations exhibited by the previous ones are tested. It is stressed that application simplicity and performance reliability of ESs allowed to evaluate improved models of higher dimensionality in a much smaller amount of time than otherwise would have been required.

The chapter is structured as follows. The system identification problem is described in Section 2 where previous results are presented. In Section 3 we introduce the ESs considered throughout the chapter. A comparative study is performed on the test problem in Section 4. The best performing ESs are applied in Section 5 to data from experimental tests on the Solarino building. Three new models for the identification of hybrid base-isolation systems are presented in Section 5.2 together with the results obtained from the identification of the Solarino building. In the final section conclusions are drawn.

## 2 Preliminaries

### 2.1 The Mechanical Model

The mechanical model simulating the experiments performed on the Solarino building is provided by the one degree of freedom system shown in Fig. 1. The justification for its use can be found in [17]. The mechanical model consists of a mass restrained by a bi-linear spring (BS) in parallel with a linear damper (LD) and a friction device (FD). Fig. 1 (a) describes the mechanical system, while Fig. 1 (b) shows the constitutive behaviour of the bi-linear spring (modelling rubber bearings). Fig. 1 (c) shows the relationship between the force in the friction device and the corresponding displacement (modelling sliding bearings).

The mechanical model is governed by the following second order ordinary differential equation

$$m \cdot \ddot{u} + c \cdot \dot{u} + f_s(u, \dot{u}) + f_d \cdot sign(\dot{u}) = 0 \tag{1}$$

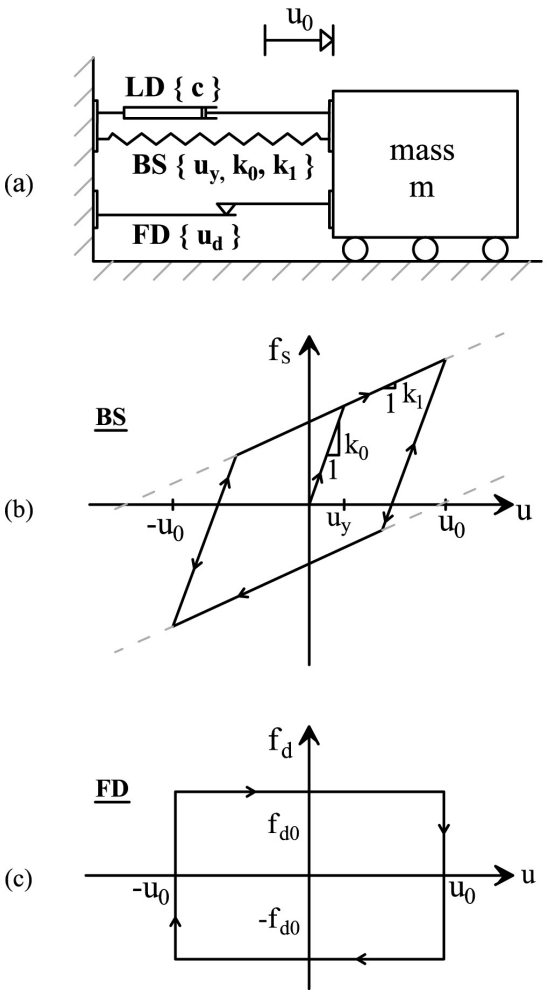



**Fig. 1.** The mechanical model: (a) mechanical system: (b) constitutive behaviour of the bi-linear spring; (c) constitutive behaviour of the friction device

where $c$ is the constant of the linear damper (LD), $f_d$ is the dynamic friction force in the friction device while $\dot{u}$ and $\ddot{u}$ are respectively the first and the second derivatives of the displacement $u(t)$ with respect to time. Physically, the derivatives represent the velocity ($\dot{u}$) and the acceleration ($\ddot{u}$) of the mass $m$ of the building. Finally, the restoring force in the bi-linear spring $f_s(u, \dot{u})$ depends on the various phases of motion of the mechanical model, that is the various branches shown in Fig. 1 (b):

$$f_s(u, \dot{u}) = k_0 \cdot [u - u_i - u_y \cdot sign(\dot{u})] + k_1 \cdot [u_i + u_y \cdot sign(\dot{u})]$$

for the branches of slope $k_0$ and

$$f_s(u, \dot{u}) = k_0 \cdot u_y \cdot sign(\dot{u}) + k_1 \cdot [u - u_y \cdot sign(\dot{u})]$$

for the branches of slope $k_1$. In the above equations $u_i$ is the displacement at the beginning of the considered phase of motion while $u_y$ is the yield displacement of the bi-linear spring as shown in Fig. 1 (b). The equation of motion (1), is supplemented by the following two initial conditions which are implicit to the considered experiment:

$$u(t_0) = u_0 \qquad \dot{u}(t_0) = 0$$

and $u_0$ is the imposed displacement.

The stated problem is highly non-linear but due to the very simple excitation it nevertheless admits an analytical solution (refer to [17] for the solution). The existence of the analytical solution is convenient but by no means essential because the equation of motion could be solved numerically at the expense of additional computational costs and of some loss in precision.

The parameters that define the mechanical model are shown in Fig. 1, and for convenience are listed in the following vector: $(m, c, k_0, k_1, u_y, f_d)$. They represent the basic physical properties that must be identified. However, in view of the form given to the solution in [17], a new set of parameters is defined as follows: $(\omega_0, \omega_1, u_d, u_y, \zeta_0)$. This is related to the previous one by the following relationships:

$$\omega_0 = \sqrt{\frac{k_0}{m}}, \quad \omega_1 = \sqrt{\frac{k_1}{m}}, \quad u_d = \frac{f_{d_0}}{k_0}, \quad \zeta_0 = \frac{c}{2m\omega_0}$$

From Eq. (1) it can be inferred that three related response functions could be used for identification purposes: the displacement $u(t)$, the velocity $\dot{u}(t)$, and the acceleration $\ddot{u}(t)$. For the application at hand, the acceleration is the function that can be measured most easily and therefore is the one that will be used.

As already mentioned an initial displacement $u_0$ is imposed in the dynamical tests. Since the measurement of $u_0$ can be difficult it may be considered as an additional parameter that must be identified. Therefore, the system parameter vector to be optimised is the following one:

$$S = (u_0, \omega_0, \omega_1, u_d, u_y, \zeta_0)$$

Let $A_0$ be a vector of accelerations and $t_0$ be the vector of the corresponding times. Furthermore, let $A$ be a vector of the same length as $A_0$ and $t$ the vector of the corresponding times representing a candidate solution. Then, a measure of the distance between the experimental data and the modelled ones is provided by the following expression:

$$e^2 = \frac{(A_0 - A, A_0 - A)}{(A_0, A_0)} + \frac{(t_0 - t, t_0 - t)}{(t_0, t_0)}$$

where $(A, B) = \sum_{i=1}^{N} A_i \cdot B_i$ and $N$ is the length of the considered vectors.

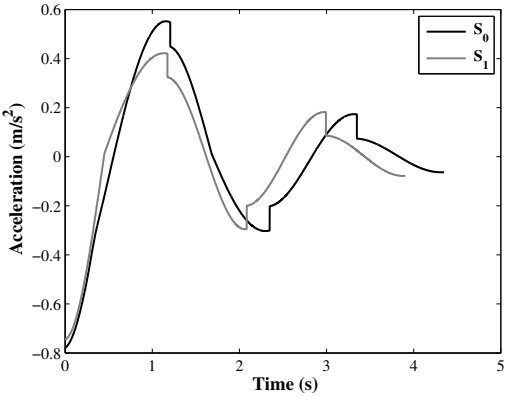



**Fig. 2.** Acceleration functions corresponding to the sets of system parameters shown in Table 1

**Table 1.** First line: the two sets of system parameters considered in Fig. 2. $S_1$ is also the solution vector of the Test problem. Second line: range of admissible values for the system parameters. Third line: performance of the identification procedure according to the number of branches considered.

| System | - | $u_0$(m) | $\omega_0$(Hz) | $\omega_1$(Hz) | $u_d$(m) | $u_y$(m) | $\zeta_0$ |
|---|---|---|---|---|---|---|---|
| $S_0$ | - | 0.12 | 0.50 | 0.40 | 0.005 | 0.02 | 0.05 |
| $S_1$(opt) | - | 0.12 | 0.55 | 0.35 | 0.004 | 0.03 | 0.03 |
| Lower Bound | - | 0.10 | 0.52 | 0.24 | 0.003 | 0.02 | 0.01 |
| Upper Bound | - | 0.14 | 0.58 | 0.48 | 0.005 | 0.04 | 0.05 |
| Branches | Error | | | | | | |
| 1 | $9.0981 \cdot 10^{-17}$ | 0.12 | 0.5500 | 0.3500 | 0.0040 | 0.0300 | 0.0300 |
| 1-2 | $9.4827 \cdot 10^{-10}$ | 0.12 | 0.5500 | 0.3500 | 0.0040 | 0.0300 | 0.0300 |
| 1-2-3 | $5.4252 \cdot 10^{-5}$ | 0.12 | 0.5503 | 0.3520 | 0.0038 | 0.0296 | 0.0338 |
| 1-2-3-4 | $2.4896 \cdot 10^{-4}$ | 0.12 | 0.5509 | 0.3534 | 0.0037 | 0.0294 | 0.0372 |

### 2.2 Previous Results

Before applying the iterative least squares method to the experimental data, in [17] the procedure was tested on a mathematically generated dataset. In such a way, the optimal solution was known beforehand and the measurement noise excluded. Two system parameter vectors were defined so that one could be used to generate a set of experimental (analytical) data (i.e., $S_1$), and the other to define the starting point of the identification process (i.e., $S_0$). The two vectors are given in Table 1 and the related acceleration series are shown in Fig. 2.

The observation of Fig. 2 shows two kinds of discontinuities in the acceleration graphs, a function discontinuity and a slope discontinuity. Between two function discontinuities a continuous branch can be identified. The two graphs show the same number of branches; however, depending on the values of the system parameter vector the number of branches can be different.

Table 1 shows the results obtained in [17] with the iterative least squares method for the test problem. Better results are obtained by using only one or two branches of the acceleration record as may be seen from the error amplitude and by the coincidence of the identified parameters with the assumed ones. As the number of branches included in the identification procedure increases so does the error and the identified parameters are no longer coincident with the assumed ones.

The iterative least squares procedure described in [17] is based on the numerical calculation of the gradients of the test functions with respect to the system parameters (refer to [17], equations (29) – (35) for the actual procedure). A suitable arc length is chosen in the trial system parameter vector space by appropriately modifying the system of equations so that each component of the unknown vector is dimensionless. In order to make the procedure efficient it is necessary to "manually" reduce the "arc length" as the procedure converges and the error becomes smaller.

Although there certainly is no presence of noise and the function to be identified does "fit" the model, the best found solution exhibits an error of the order of $10^{-4}$. The identified parameters differ from the given ones already at the third decimal digit. This is far from desired and excludes obtaining better results for the real data recorded at Solarino for the presence of noise and modelling errors.

The results obtained for the Solarino data are presented directly in Section 5. In a private communication the first author of reference [17] asserts that the use of Powell's method (a popular optimization method which does not require the calculation of gradients) to the considered problem does not lead to better results than those obtained by the least squares method (see [18], pages 97-108 for a description of Powell's method).

A main problem in applying the least squares method or any optimization method is the definition of the starting point which in the case shown above was the set of parameters in Table 1 corresponding to the acceleration graph denoted by $S_0$ in Fig. 2. In practice this problem can be overcome by providing suitable lower and upper bounds to the sought parameters. This can be done by using physical insight on the observation of the given acceleration record. The bounds shown in Table 1 for the set of parameters $S_1$ were derived in [2].

## 3 Evolution Strategies

The first applications of ESs were directed towards the parameter optimisation for civil engineering simulation models i.e., simulating stress and/or displacement states of structures under load [21]. Obviously the performance of the ESs was compared against their natural competitors i.e., the mathematical methods used for the purpose, especially those not requiring the explicit use of derivatives, [21].

Similarly, in the next section the performance of ESs will be compared against the methods applied in [17] on the same mathematically generated dataset (i.e., the test problem). To this end both very simple ESs and more complicated ones such as the CMA-ES will be considered.

A general $(\mu \dagger \lambda)$-ES maintains a *parent population* of $\mu$ individuals, each consisting of a *solution vector* $\mathbf{x}$ and of a *strategy vector* $\mathbf{s}$. The solution vector represents the candidate solution to the optimisation problem. The strategy vector is a set of one or

more parameters that are used in combination with the solution vector to create new candidate solutions.

In the considered problem an individual is represented as $(\mathbf{x}, \mathbf{s})$ where the solution vector $\mathbf{x} = (x_1, x_2, \ldots, x_6) \in \mathbb{R}^6$, is a real-valued vector for the candidate solution of the system parameter vector $\mathbf{S}$ introduced in Section 2. The initial $\mu$ solution vectors are generated uniformly at random within the parameter bounds given in Table 1.

In each optimisation step an *offspring population* of $\lambda$ individuals is generated. Each individual is created by first selecting one of the $\mu$ individuals out of the parent population uniformly at random, and then by moving it in the search space of a quantity determined by applying its strategy vector $\mathbf{s}$. The generation is completed by selecting the best $\mu$ individuals out of the parent and of the offspring populations if the *plus selection strategy* is used (i.e., ($\mu$+$\lambda$)-EA) or out of the offspring population if the *comma selection strategy* is adopted (i.e., ($\mu$,$\lambda$)-EA). The latter requires that $\lambda$ be greater than $\mu$.

The way the strategy vector $\mathbf{s}$ is applied to generate new individuals is explained when describing each ES considered in this paper. The main differences between various subclasses of ESs are in the size of the strategy vector, on how it is used and on how its values change during the optimisation process (i.e., *adaptation*). The question of how to adapt the strategy vector (i.e., the step-size) is central in the field of stochastic optimization. Already in 1971 a survey of adaptation techniques was written by [24], see [4] for a *tour d'horizon*.

### 3.1 1/5 Rule and Self-adaptation

The first considered algorithm is the simple (1+1)-ES using a strategy vector consisting of only one strategy parameter $\sigma$ (i.e., $\mathbf{s} \in \mathbb{R}^1$). The solution vector $\mathbf{x}$ of the parent individual ($\mathbf{x} \in \mathbb{R}^6$, $\sigma$) is initialised uniformly at random in the bounded search space. At each generation a new candidate solution is obtained by applying $\tilde{\mathbf{x}} := \mathbf{x} + \mathbf{z}$ where $\mathbf{z} := \sigma(\mathcal{N}_1(0,1), \ldots, \mathcal{N}_N(0,1))$ and $\mathcal{N}_i(0,1)$ are independent random samples from the standard normal distribution. The only parameter of the algorithm is the standard deviation $\sigma$ of the normal distribution used to generate the offspring solution vector.

One of the first methods proposed to control the mutation rate in an ES was the $1/5$-*rule* adaptation strategy [5]. The idea is to tune $\sigma$ in a way that the *success rate* (i.e., the measured ratio between the number of steps when the offspring is retained and that when it is discarded) is about $1/5$. This idea was already proposed by [20] and can also be found in [8]. Rechenberg introduced it to the field of ESs and gave it the "1/5-success rule" name [19]. For the sphere function the $1/5$-rule has been proved to lead the (1+1)-ES to optimal mutation rates, hence optimal performance [11].

The "classical" strategy works as follows. After a given number of steps $G$, the mutation strength (i.e., the standard deviation $\sigma$) is reduced by $\alpha$ if the rate of successful mutations $P_s := G_s/G$ is less than $1/5$. On the other hand, if $P_s > 1/5$, the mutation rate is increased by $\alpha$. Otherwise it remains unchanged. Recommended values are $G = N$, if the dimensionality of the search space $N$ is sufficiently large, and $0.85 \leq \alpha < 1$, [5].

Kern et al. [14] recently proposed a $1/5$-rule strategy that allows to update the step size after each generation removing the need for a "window phase" $G$. In this simpler implementation, at each generation the step size is updated according to:

$$\sigma^{t+1} = \sigma^t \cdot \begin{cases} \alpha & \text{if } f(x^{t+1}) \leq f(x^t) \\ \alpha^{(-1/4)} & \text{otherwise.} \end{cases}$$

Here $\alpha = 1/3$ and the $(-1/4)$ in the exponent corresponds to the success rate of $1/5$.

*Self-adaptation* has been introduced as a mechanism for the ES to automatically adjust the mutation strength, by evolving not only the solution vector but also the strategy parameters. The strategy vector is also mutated such that standard deviations producing fitter solutions have higher probabilities of survival, hence are evolved implicitly.

By still considering only one strategy parameter, a mutation with self-adaptation of individual $(\mathbf{x}, \sigma)$ involves first generating a new $\sigma$-value and then applying it to the object vector $\mathbf{x}$. This is done by setting $\tilde{\sigma} := \sigma \exp(\tau \mathcal{N}(0,1))$, $\mathbf{z} := \tilde{\sigma}(\mathcal{N}_1(0,1), \ldots, \mathcal{N}_N(0,1))$ and $\tilde{\mathbf{x}} := \mathbf{x} + \mathbf{z}$. Here $\tau = 1/\sqrt{2N}$ is generally recommended as standard deviation for $\tilde{\sigma}$, [5]. By using only one strategy parameter $\sigma$, the mutation distribution is isotropic, i.e., surfaces of equal probability densities are hyper-spheres (circles for $N = 2$ and spheres for $N = 3$).

If $N$ strategy parameters are used, individual step sizes for each dimension are obtained leading to ellipsoidal surfaces of constant probability density as the standard deviations evolve. With a strategy vector $\mathbf{s} := (\sigma_1, \ldots \sigma_N)$ a new individual is generated by setting:

$$\tilde{\mathbf{s}} := \exp(\tau_0 \mathcal{N}_0(0,1)) \cdot \left(\sigma_1 \exp(\tau \mathcal{N}_1(0,1), \ldots, \sigma_N \exp(\tau \mathcal{N}_N(0,1))\right)$$

and $\mathbf{z} := (\sigma_1 \mathcal{N}_1(0,1), \ldots, \sigma_N \mathcal{N}_N(0,1))$. Recommended values for the parameters are $\tau_0 = 1/\sqrt{2N}$ and $\tau = 1/\sqrt{2\sqrt{N}}$, [5].

### 3.2 CMA-ES

Since the success of the described self-adaptation technique relies on one-step improvements it is often referred to as a *local adaptation* approach. By introducing *correlations* between the components of $\mathbf{z}$ the ellipsoid may be arbitrarily rotated in the search space and evolved to point in the direction of optimal solutions. Another step towards more advanced parameter adaptation techniques is to consider *non-local* information gathered from more than one generation. Both features are used by the $(\mu, \lambda)$-CMA-ES considered in this section.

The CMA-ES creates a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ determined by its mean vector $\mathbf{m} \in \mathbb{R}^{\mathbb{N}}$ and its covariance matrix $\mathbf{C} \in \mathbb{R}^{\mathbb{N}\mathbb{X}\mathbb{N}}$. Instead of keeping a population of $\mu$ individuals (as in the previously considered ESs), the covariance matrix $\mathbf{C}$ and the mean vector $\mathbf{m}$ are evolved. At each step $\lambda$ individuals are sampled from $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{C})$ and the best $\mu$ are used to generate the new mean $\tilde{\mathbf{m}}$ and covariance matrix $\tilde{\mathbf{C}}$. For further details on the CMA-ES refer to [10].

Unless stated otherwise, all the algorithmic parameters are set as recommended in [10] including $\lambda = 4 + \lfloor 3 \ln N \rfloor$ and $\mu = \lambda/2$.

## 4 Test Problem Study

In this section the performance of popular ESs with standard parameter settings are applied to the test problem considered in [17].

**Table 2.** (a) Summary of best results found for the Test function with 1000 fitness function evaluations; (b) 10000 fitness function evaluations. The initial step sizes of the algorithms are $\sigma_0 = 0.01$ for the (1+1)-ES, $\sigma_0 = 1$ for the (1+$\lambda$)-ES and $\sigma_0 = 0.3$ for the CMA-ES.

| ES | adaptation | (a) Avg | Med | Min | (b) Avg | Med | Min |
|---|---|---|---|---|---|---|---|
| (1+1)-ES | No | 0.0595 | 1.48E-04 | 4.51E-06 | 1.02E-05 | 8.35E-06 | 8.47E-07 |
| (1+1)-ES | 1/5 ($\alpha = 0.95$, $G = 5$) | 1.20E-04 | 7.97E-05 | 2.08E-07 | 1.32E-08 | 2.36E-09 | 1.39E-12 |
| (1+$\lambda$)-ES | self ((1+5)-ES) | 0.0023 | 3.54E-04 | 2.30E-06 | 7.89E-05 | 5.69E-06 | 2.16E-08 |
| CMA-ES | non-loc. rot. ellips. self | 2.51E-05 | 5.79E-06 | 2.35E-10 | 2.22E-15 | 5.66E-16 | 2.14E-16 |

Given the different value ranges for the bounds of each parameter, the solution is normalised according to $x_n = (x - \ell)/(u - \ell)$ where $x_n$ is the normalised solution and $u$, $\ell$ are the upper and lower bound vectors on the solution space. For the plus-selection algorithms a large penalty value is given to points outside the feasible area. Since the algorithms are initialised inside the bounds, plus-selection will never accept infeasible points. Concerning CMA-ES, its standard but more sophisticated constraint handling technique is applied. The fitness of an infeasible solution is evaluated by adding a penalty function to the fitness of its projection on the feasible domain (i.e., to the feasible solution that is closest to the infeasible one). The penalty depends on the distance of the infeasible solution to the feasible domain and is weighted and scaled in each coordinate, see [9] for more details. All the algorithms are not allowed to start from feasible points of fitness=1 corresponding to a different number of branches between candidate and optimal solutions. Large step sizes would be required to escape from these artificial plateaus.

The solutions of the best performing algorithms for each adaptation method are given in Table 2. Average, median and best found solutions out of 100 runs are shown. From the table it can be seen how simple evolution strategies outperform the least squares method (Table 1) and that the mean and best results scored by the CMA-ES are of order $10^{-15}$ and $10^{-16}$ respectively; practically the exact solution. Figures 3 and 4 respectively show how the fitness and the step-size of the algorithms evolve when $\sigma_0 = 0.01$. More detailed results for the test problem involving a wider variety of algorithm configurations have been presented in [1].

## 5 Solarino Data

In July 2004, six free vibration tests under imposed displacement were performed on a four story reinforced concrete building, seismically retrofitted by base isolation. The nominal displacements varied from a minimum of 4.06 cm to a maximum of 13.29 cm in the six dynamic tests. Unfortunately, these may not be true displacements since they may include residual displacement from tests performed previously in the sequence. The main objective, is the identification of the properties of the isolation system (i.e., the parameters of the previously described model) and the initial displacement as discussed above using the recorded accelerations.

The results obtained in [17] using an iterative least squares method are shown in Table 3. The procedure implied to start from a reasonable guess for the system parameters (for the identification of the first of the six tests), and use the first branch of the recorded

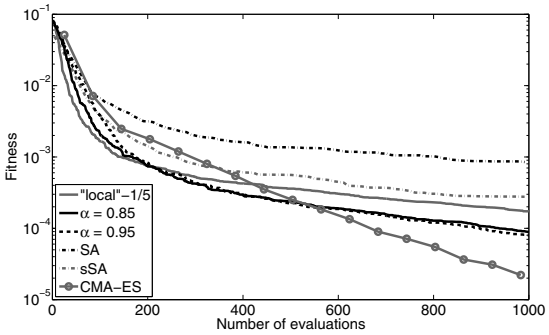



**Fig. 3.** Performance of the three 1/5 success rule strategies, the CMA-ES, the isotropic (sSA) and the ellipsoidal (SA) self-adaptation strategies on the test problem when $\sigma_0 = 0.01$. Median values out of 100 runs are plotted.

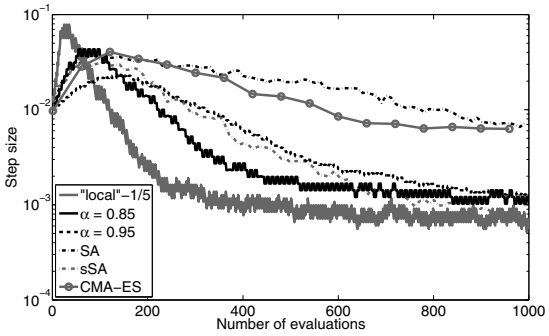



**Fig. 4.** Step size evolution of the median fitness run for the three 1/5 success rule strategies, the CMA-ES, the isotropic (sSA) and the ellipsoidal (SA) self-adaptation strategies on the test problem when $\sigma_0 = 0.01$

acceleration for the identification of a new set of improved parameter values. This new set of parameters was then used for the identification from the first two branches and so on until five of the branches were considered at once. The last branch was derived from the governing equations using the parameters identified from the previous segments of the signal. For the identification of the following tests the result of the first test was used as the initial guess (i.e., the correct solution should be the same for all tests excluding damage of the building caused by the tests and/or effects due to environmental changes).

### 5.1 Previous Model

In this section advantage is taken of the simplicity of ES application and the ESs that performed best on the test problem are used. The parameter bounds to the search space, derived in [2] for the Solarino data, are shown in Table 3.

While the CMA-ES once again outperforms the other strategies, the difficulty of the identification problem is highlighted by the need to increase the population size to

a (18,36)-CMA-ES to obtain improved results. Ten runs each are performed of (4,9), (9,18) and (18,36) CMA-ES and of the three 1/5 rule ESs allowing 100k fitness evaluations in each run. Starting from parent population sizes of $\mu = 9$ the CMA-ES converges more than once to the same best found solution.

Table 3 shows the best found solutions and the number of times they were repeated. The other solutions do not differ significantly from the best ones but for slightly larger errors and final parameter digits. Compared to the results from previous work, closer bounds for the likely real values of the parameters are now established.

The nominal values of the initial displacements, in all likelyhood affected by measurement errors, are given for each test in [17] together with the identified ones. The estimated displacements were always smaller than the nominal ones; here they are found even smaller, except for test 8.

In three tests out of five damping ($\zeta_0$) is practically zero in line with what found in [17], while in the remaining two $\zeta_0$ is small but not negligible and a little larger than in [17]. This presence of damping in tests 5 and 8 could point to an incapacity of the optimization procedures to identify the absolute minimum, producing instead local minima. Alternatively, data inconsistency due to measurement noise and/or inadequate signal treatment could be responsible for the discrepancy.

The remaining physical parameters show less dispersion in the identified values than before, with a coefficient of variation nearly halved in each case. Actual values change from 0.13 to 0.07 for $u_d$, from 0.16 to 0.08 for $u_y$, from 0.02 to 0.01 for $\omega_0$ and from 0.03 to 0.01 for $\omega_1$. The situation could improve even further if the inconsistency highlighted by damping could be solved.

From the results shown in Table 3 and as commented above, a considerable improvement in the identification procedure has been achieved by using ESs with the Solarino test data. In particular the results are improved up to 53% compared to those previously available in literature. The performance discrepancy of ESs between the test problem and the real problem seems to suggest that further improvements might be required in the formulation of the mechanical model. Some preliminary investigations in that direction are pursued in the next section.

### 5.2 New Models

Two small changes to the described model are investigated herein. They affect only the description of the low friction slider of Fig. 1 and will reflect in changes to the diagram of Fig. 1 (c). The changes stem from the experimental observation that the friction force is not constant during the motion. The diagram of Fig. 5 assumes that the friction force is constant within a half-cycle of motion but it can change from one half-cycle to the next. The considered change does not affect the equations of the mechanical model as it only changes one parameter value, but not the structure of the problem. However the dimension of the system parameter vector is affected because an additional parameter is required for each half-cycle of motion as compared to the single one required earlier.

The considered model, for convenience denoted Model 2, provides results improved only slightly by running the CMA-ES on it (i.e., quadratic error reduced from $0.0147$ to $0.0145$ on test 3). Contrary to the previous Model 1 the CMA-ES does not seem to

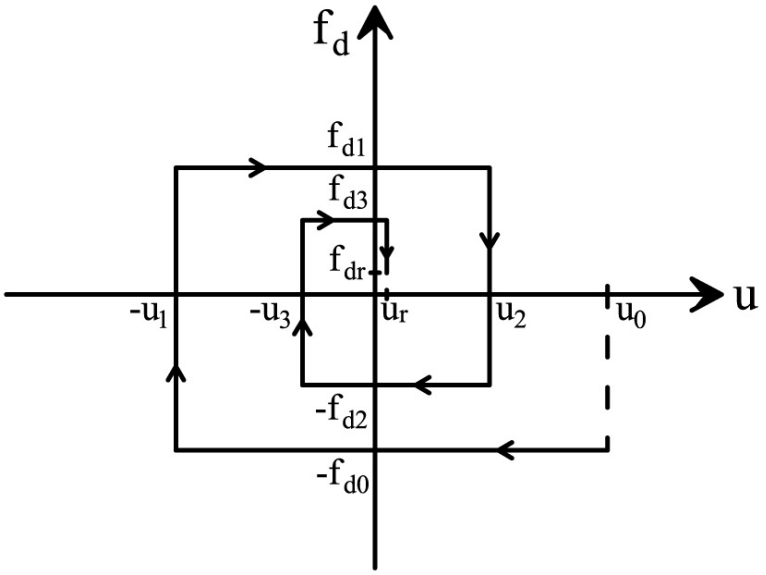



**Fig. 5.** Model 2 friction force-displacement relationship

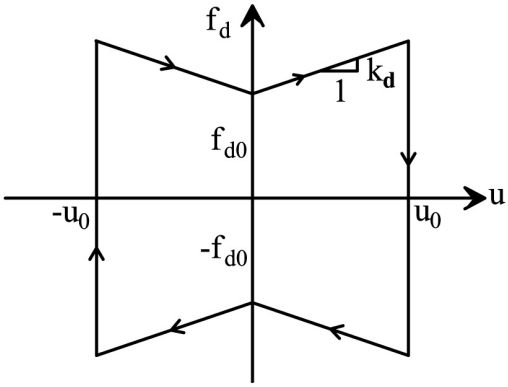



**Fig. 6.** Model 3 friction force-displacement relationship

succeed to repeat results in more than one run on Model 2. This may be due to the greatly increased dimension of the problem (number of parameters nearly doubled) and to having maintained the same number of function evaluations (100k).

The feeble success with this model may be due to two factors; the first is the computational expensiveness due to the higher dimension of the problem, the second is the insignificant mechanical advantage because the friction force changes more within a half-cycle than it does from half-cycle to half-cycle. The latter aspect will be clearer with the introduction of Model 3.

An analytical solution for the mechanical model shown in Fig.1 with the friction law sketched in Fig. 6 has just been given in [3]. This third model helps to spread some light

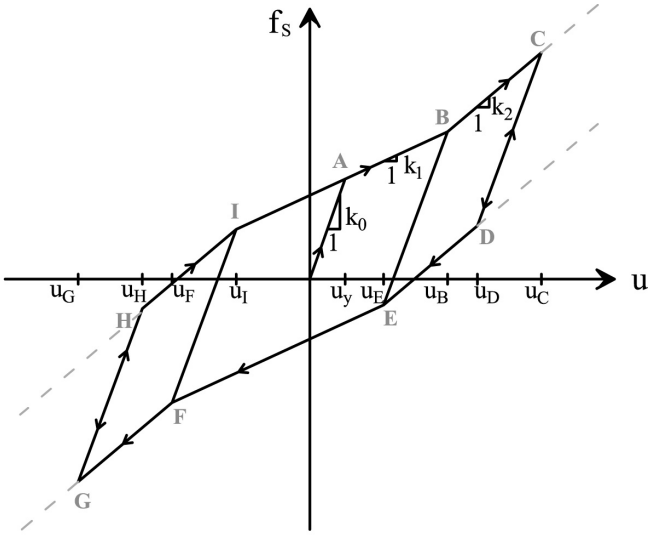



**Fig. 7.** Constitutive behaviour for the tri-linear spring model

on the little efficiency of Model 2. While in Model 3 the friction force varies linearly over a half-cycle, in Model 2 the friction force is constant over the same half-cycle. It can be expected therefore that Model 3 can fine tune the friction force much better than Model 2 and for that matter much better than Model 1, where the friction force is of constant amplitude. All this is achieved at the expense of only one additional parameter, the slope $k_d$ in Fig.6. The application of the (18,36)-CMA-ES with Model 3 to Test 3 converges 4 times out of 10 to a solution with quadratic error $e^2 = 0.0139$ which is an extra improvement of more than 5% on the final fitness.

Finally, we introduce changes in the model describing the rubber bearings. A tri-linear spring model is used instead of the bi-linear one used until now to model the rubber bearings. This change only affects the diagram of Fig. 1 (b). Apart from considering the elastic stiffness $k_0$ and one post-yielding stiffness $k_1$ like in the bi-linear spring, the new model also considers a second post-yielding stiffness $k_2$. Together with the second yielding displacement $u_B$ this new model requires two additional parameters. The tri-linear spring model is explained in Fig. 7 and the results obtained by the (18-36)-CMA-ES using Model 3 for the low friction slider and the tri-linear model for the rubber bearings are shown in Table 3. With the new models combined the CMA-ES delivers improvements between 31% and 79% on all tests and again best found solutions are repeated several times. An idea of the improvement achieved by applying the CMA-ES to the most advanced of the considered models can be obtained by comparing the identified response of Test 3 on the Solarino building with the experimental data; first by using the original model and the Least Squares (LS) identification procedure, Fig. 8, and then by using the linear friction model (i.e., Model 3) coupled with the tri-linear spring model and the CMA-ES identification algorithm, Fig. 9. Apart from the

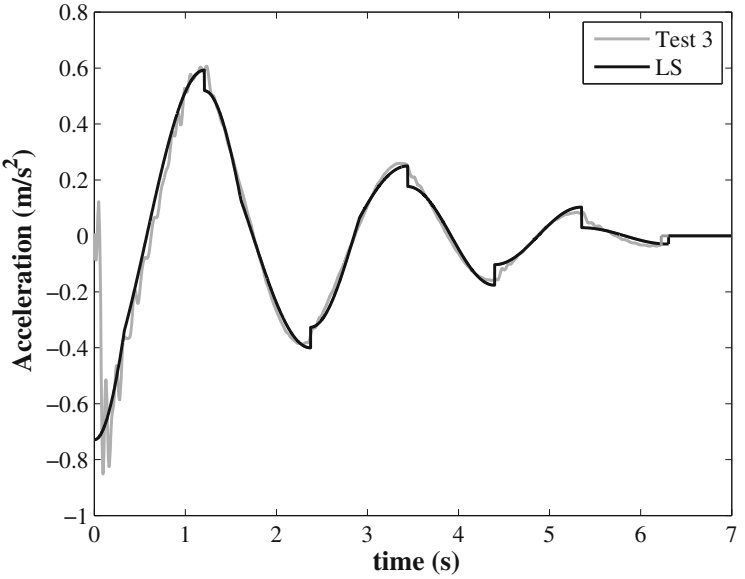



**Fig. 8.** Comparison between the experimental data and the identified response of Test 3 on the Solarino building obtained by using the original model and the Least Squares (LS) identification procedure

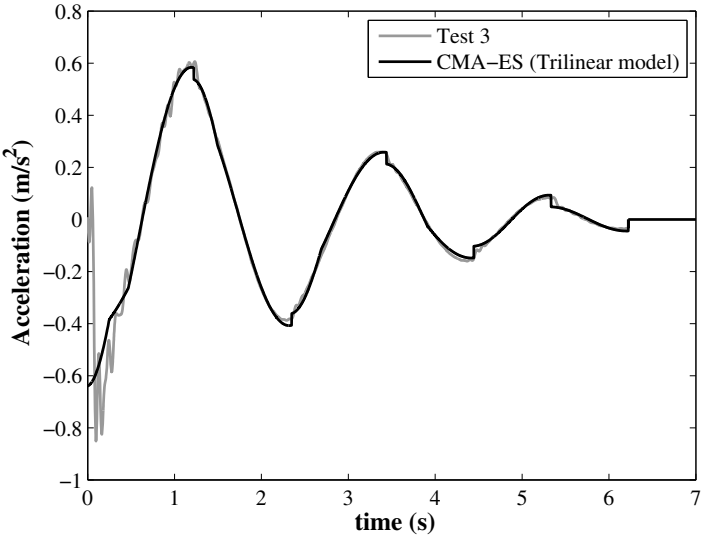



**Fig. 9.** Comparison between the experimental data and the identified response of Test 3 on the Solarino building obtained by using the combination of the linear Coulomb friction model and of the tri-linear spring model and the CMA-ES for the identification procedure

**Table 3.** Identification of the Solarino building base isolation system. The best identification results obtained in the literature by the least squares (LS) method from five tests are shown first. The corresponding results obtained by the CMA-ES are shown next. The number of times the best found solution was obtained in the runs is shown in brackets next to the test number. Quadratic errors and improvement achieved by CMA-ES are shown in the last two columns.

| Test | $u_0$ (m) | $u_d$ (m) | $u_y$ (m) | $r_{d0}$ | $\zeta_0$ | $\omega_0$ (Hz) | $\omega_1$ (Hz) | $\omega_2$ (Hz) | $u_B$ (m) | $e^2$ | Impr. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LB | 0.0813 | 0.0018 | 0.0092 | 1E-10 | 0 | 0.4832 | 0.2819 | 0.2163 | 0.032 | - | - |
| UB | 0.1333 | 0.0069 | 0.0366 | 0.10 | 0.1 | 0.5651 | 0.4417 | 0.45 | 0.1 | - | - |
| LS | | | | | | | | | | | |
| 3 | 0.1108 | 0.0034 | 0.0181 | - | 4.5E-08 | 0.5235 | 0.3947 | - | - | 0.0234 | - |
| 5 | 0.1169 | 0.0034 | 0.0167 | - | 0.0127 | 0.5117 | 0.4070 | - | - | 0.0105 | - |
| 6 | 0.1228 | 0.0035 | 0.0179 | - | 3.6E-08 | 0.5269 | 0.3909 | - | - | 0.0129 | - |
| 7 | 0.0927 | 0.0033 | 0.0173 | - | 3.87E-08 | 0.5222 | 0.3964 | - | - | 0.0122 | - |
| 8 | 0.0965 | 0.0025 | 0.0118 | - | 0.0306 | 0.5402 | 0.4242 | - | - | 0.0055 | - |
| CMA-ES | Model 1 | | | | | | | | | | |
| 3 (4) | 0.1063 | 0.0026 | 0.0132 | - | 1E-10 | 0.5507 | 0.4014 | - | - | 0.0147 | 37% |
| 5 (2) | 0.1153 | 0.0025 | 0.0127 | - | 0.0153 | 0.5384 | 0.4108 | - | - | 0.0049 | 53% |
| 6 (2) | 0.1122 | 0.0027 | 0.0130 | - | 1E-10 | 0.5559 | 0.4036 | - | - | 0.0096 | 25% |
| 7 (3) | 0.0856 | 0.0030 | 0.0132 | - | 1E-10 | 0.5455 | 0.4130 | - | - | 0.0106 | 13% |
| 8 (4) | 0.0976 | 0.0026 | 0.0109 | - | 0.03074 | 0.5382 | 0.4235 | - | - | 0.0052 | 5% |
| CMA-ES | Model 3 + Tri-linear spring model | | | | | | | | | | |
| 3 (8) | 0.1076 | 0.0018 | 0.0091 | 1E-10 | 0.0292 | 0.5627 | 0.4310 | 0.2649 | 0.0695 | 0.0049 | 79% |
| 5 (3) | 0.1167 | 0.0023 | 0.0118 | 0.0028 | 0.0221 | 0.5393 | 0.4168 | 0.3762 | 0.0826 | 0.0043 | 59% |
| 6 (3) | 0.1148 | 0.002 | 0.011 | 1E-10 | 0.021 | 0.561 | 0.4212 | 0.3109 | 0.0767 | 0.0054 | 58% |
| 7 (2) | 0.085 | 0.0022 | 0.0088 | 1E-10 | 0.0298 | 0.5564 | 0.4419 | 0.3337 | 0.0545 | 0.0068 | 44% |
| 8 (5) | 0.0924 | 0.0024 | 0.0081 | 1E-10 | 0.0351 | 0.5528 | 0.4436 | 0.3111 | 0.0633 | 0.0038 | 31% |

high frequency components at very beginning of the test, that are outside the scope of the considered models, the better matching achieved with the advanced model and the CMA-ES identification technique can be perceived at first sight.

The mathematical solution for the newly proposed model is considerably more complex than the ones derived for the previous models and will appear in [16].

## 6 Conclusions

A study of ESs for the identification of base isolation systems in buildings designed for earthquake action has been presented. The results clearly show that ESs are highly effective for the optimisation of the test problem defined in previous work for methodology validation. All the considered ESs perform at least as well as previous methods and the quality of solutions improves with more sophisticated adaptation strategies. The CMA-ES, considerably outperforms previous algorithms of several orders of magnitude.

Having established the good performance of ESs for the system identification, the same ESs are applied to the real data recorded on the Solarino building in July 2004 with the model used previously in literature. Although the CMA-ES converges to more precise solutions, these do not allow the sought system parameter vector to be established with the highest confidence level. Two possible causes have been considered: the

presence of noise in the recorded data (i.e., the function to be optimised) and the model adequacy to properly simulate the system response.

To investigate the latter problem three new mechanical models with higher number of parameters have been developed. All models, in conjunction with the application of CMA-ES, enable to obtain improved results. Combining the best model for the low friction sliders (linear Coulomb model) with the best model for the rubber bearings (trilinear spring model) allows the fine tuning of the behaviour of both devices providing improvements of up to 79% in the overall solution quality.

Thus, in this chapter ESs are shown to be very powerful tools for the dynamic identification of structural systems. In particular, the CMA-ES combines application simplicity with convergence reliability. In view of the effectiveness shown in the applications considered in the present work, the authors believe that CMA-ES could be used advantageously with more complex models and various excitation sources including environmental ones.

The data collected during the 2011 Tohoku Pacific Ocean Earthquake (Japan) on several base isolated buildings and their availability to the international scientific community [13] make the present work even more significant.

**Acknowledgements.** A. Athanasiou and G. Oliveto carried out this work within the PRIN 2007 Project "Seismic Retrofitting of Buildings Using Isolation and/or Energy Dissipation Techniques".

## References

1. Athanasiou, A., De Felice, M., Oliveto, G., Oliveto, P.S.: Evolutionary algorithms for the identification of structural systems in earthquake engineering. In: International Conference on Evolutionary Computation Theory and Applications, pp. 52–62 (2011)
2. Athanasiou, A., Oliveto, G.: Upper and lower bounds for the parameter vector in dynamic identification of hybrid base isolation systems. In: Conference in Honour of Prof. Guido Sara': Lezioni dai Terremoti: Fonti di Vulnerabilita', Nuove Strategie Progettuali, Sviluppi Normativi, Chianciano Terme, Siena, Italy (2010)
3. Athanasiou, A., Oliveto, G.: Modelling hybrid base isolation systems for free vibration simulations. In: 8CUEE: Proceedings of the 8th International Conference on Urban Earthquake Engineering, Tokyo, Japan, pp. 1293–1302 (2011)
4. Auger, A., Hansen, N.: Theory of evolution strategies: A new perspective. In: Auger, A., Doerr, B. (eds.) Theory of Randomized Heuristics - Foundations and Recent Developments, pp. 289–325. World Scientific (2011)
5. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies, a comprehensive introduction. Natural Computing 1, 3–52 (2002)
6. Braga, F., Laterza, M.: Field testing of low-rise base isolated buildings. Engineering Structures 26, 1599–1610 (2004)
7. Braga, F., Laterza, M., Gigliotti, R., Laterza, M.: Nonlinear dynamic response of HRDB and hybrid HRDB-friction sliders base isolation systems. Bullettin of Earthquake Engineering 3, 333–353 (2005)
8. Devroye, R.: The compound random search. In: International Symposium on Systems Engineering and Analysis, pp. 195–210. Purdue University (1972)

9. Hansen, N., Niederberger, S., Guzzella, L., Koumoutsakos, P.: A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. IEEE Transactions on Evolutionary Computation 13(1), 180–197 (2009)
10. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9(2), 159–195 (2001)
11. Jägersküpper, J.: Analysis of a Simple Evolutionary Algorithm for Minimization in Euclidean Spaces. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 1068–1079. Springer, Heidelberg (2003)
12. Kasai, K. (ed.): Special Issue on Japan's Advanced Technology for Building Seismic Protection. Journal of Disaster Research, vol. 4(3) (2009)
13. Kasai, K.: Response-Controlled Structures, Tohoku Pacific Ocean Earthquake. Clearinghouse at International Center for Urban Earthquake Engineering, Tokyo Institute of Technology, Japan (2011)
14. Kern, S., Müller, S.D., Hansen, N., Büche, D., Ocenasek, J., Koumoutsakos, P.: Learning probability distributions in continuous evolutionary algorithms - a comparative review. Natural Computing 3, 77–112 (2003)
15. Moroni, M., Sarazzin, M., Boroshek, R.: Experiments on a base-isolated building in Santiago, Chile. Engineering Structures 20(8), 720–725 (1998)
16. Oliveto, G., Markou, A.A., Athanasiou, A.: Recent advances in dynamic identification and response simulation of hybrid base isolation systems. In: 15th World Conference on Earthquake Engineering (15WCEE), Lisbon, September 24-28 (to appear, 2012)
17. Oliveto, N.D., Scalia, G., Oliveto, G.: Time domain identification of hybrid base isolation systems using free vibration tests. Earthquake Engineering and Structural Dynamics 39(9), 1015–1038 (2010)
18. Ravindran, A., Ragsdell, K.M., Reklaitis, G.V. (eds.): Engineering Optimization: Methods and Applications, 2nd edn. John Wiley & Sons, Inc., New Jersey (2006)
19. Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog Verlag, Stuttgart (1973)
20. Schumer, M., Steiglitz, K.: Adaptive step size random search. IEEE Transactions on Automatic Control 13, 270–276 (1968)
21. Schwefel, H.-P. (ed.): Evolution and Optimum Seeking: The Sixth Generation. John Wiley & Sons, Inc., New York (1993)
22. Seki, M., Miyazaki, M., Tsuneki, Y., Kataoka, K.: A masonry school buiding retrofitted by base isolation. In: 12th World Conference on Earthquake Engineering (12WCEE), Auckland, New Zealand (2000)
23. Wada, A., Constantinou, M.C., (eds.): Special Issue on Seismic Protection Techniques. Earthquake Engineering and Structural Dynamics, vol. 39(13) (2010)
24. White, R.: A survey of random methods for parameter optimization. Simulation 17, 197–205 (1971)

# Skeletal Algorithms in Process Mining*

Michal R. Przybylek

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
mrp@mimuw.edu.pl

**Abstract.** This paper[1] studies sample applications of skeletal algorithm to process mining and automata discovery. The basic idea behind the skeletal algorithm is to express a problem in terms of congruences on a structure, build an initial set of congruences, and improve it by taking limited unions/intersections, until a suitable condition is reached. Skeletal algorithms naturally arise in the context of process minig and automata discovery, where the skeleton is the "free" structure on initial data and a congruence corresponds to similarities in data. In such a context, skeletal algorithms come equipped with fitness functions measuring the complexity of a model. We examine two fitness functions for our sample problem — one based on Minimum Description Length Principle, and the other based on Bayesian Interpretation.

## 1 Introduction

The idea of evolutionary computing dates back to the late 1950, when it was first introduced by Bremermann in [3], Friedberg, Dunham and North [6,7], and then developed by Rechenberg in [15], and Holland in [11]. Skeletal algorithm derives its foundations from these methods and creates a new branch of evolutionary metaheuristics concerned on data and process mining. The crucial observation that leads to skeletal algorithms bases on Minimum Description Length Principle [9], which among other things, says that the task of finding "the best model" describing given data is all about discovering similarities in the data. Thus, when we start from a model that fully describes the data (i.e. the skeletal model of the data), but does not see any similarities, we may obtain a "better model" by unifying some parts of that model. Unifying parts of a model means just taking a quotient of that model, or equally — finding a congruence relation.

### 1.1 Process Mining

Process mining [18,25,5,22,21,24,27,19,20] is a new and prosperous technique that allows for extracting a model of a business process [13] based on information gathered during real executions of the process. The methods of process mining are used when there is no enough information about processes (i.e. there is no a priori model), or there is a need to check whether the current model reflects the real situation (i.e. there is a priori model, but of a dubious quality). One of the crucial advantages of process mining

* This work has been partially supported by Polish National Science Center, project DEC-2011/01/N/ST6/02752.

[1] The article is an essentially revised version of conference paper [14].

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 119–134.
DOI: 10.1007/978-3-642-35638-4_9 

| Case | Observable Action | Actor | Timestamp | Data |
|---|---|---|---|---|
| 127 | START | Dr. Moor | 11:30:52 07.02.2011 | |
| 127 | Listen to patient's complaints | Dr. Moor | 11:34:27 07.02.2011 | headache |
| 127 | Listen to patient's complaints | Dr. Moor | 11:35:59 07.02.2011 | fever |
| 107 | START | Dr. No | 11:36:50 07.02.2011 | |
| 127 | Listen to patient's complaints | Dr. Moor | 11:39:33 07.02.2011 | catarrh |
| 107 | Listen to patient's complaints | Dr. No | 11:39:37 07.02.2011 | pain in the left foot |
| 127 | Select a candidate disease | Dr. Moor | 11:58:30 07.02.2011 | angina |
| 127 | Query patient about symptoms | Dr. Moor | 12:01:11 07.02.2011 | sore throat? — yes |
| 127 | Query patient about symptoms | Dr. Moor | 12:08:21 07.02.2011 | white patches on the tonsils? — yes |
| 107 | Select a candidate disease | Dr. No | 12:10:31 07.02.2011 | broken leg |
| 107 | Query patient about symptoms | Dr. No | 12:11:01 07.02.2011 | swollen leg? — No |
| 107 | Select a candidate disease | Dr. No | 12:11:33 07.02.2011 | joint dislocation |
| 107 | Query patient about symptoms | Dr. No | 12:14:00 07.02.2011 | blood inflammation? — Yes |
| 107 | Make a diagnosis | Dr. No | 12:16:02 07.02.2011 | joint dislocation |
| 107 | END | Dr. No | 12:16:50 07.02.2011 | |
| 127 | Make a diagnosis | Dr. Moor | 12:34:01 07.02.2011 | angina |
| 127 | END | Dr. Moor | 12:34:55 07.02.2011 | |
| ... | ... | ... | ... | ... |

**Fig. 1.** Event Log

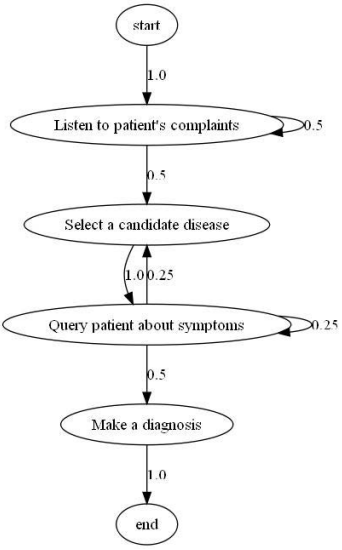



**Fig. 2.** Discovered Model

over other methods is its objectiveness — models discovered from real executions of a process are all about the real situation as it takes place, and not about how people think of the process, and how they wish the process would be like. In this case, the extracted knowledge about a business process may be used to reorganize the process to reduce its time and cost for the enterprise.

Figure 1 shows a typical event-log gathered during executions of the process to determine and identify a possible disease or disorder of a patient. In this paper, we assume that with every such an event-log there are associated:

- an identifier referring to the execution (the case) of the process that generated the event
- a unique timestamp indicating the particular moment when the event occurred
- an observable action of the event; we shall assume, that we are given only some rough information about the real actions.

and we shall forget about any additional information and attributes associated with an execution of a process. The first property says that we may divide a list of events on collections of events corresponding to executions of the process, and the second property let us linearly order each of the collections. If we use only information about the relative occurrences of two events (that is: which of the events was first, and which was second), then the log may be equally described as a finite list of finite sequences over the set $ObservableAction$ of all possible observable actions. Therefore we may think of the log as a finite sample of a probabilistic language over alphabet $ObservableAction$ — or more accurately — as the image of a finite sample of a probabilistic language

over $Action$ under a morphism $h\colon Action \to ObservableAction$. The morphism $h$ describes our imperfect information about the real actions. In the example from Figure 1 (here we use the first letter of the name of an action as abbreviate for the action)

$$ObservableAction = \{l, s, q, m\}$$

and the sample contains sequences

$$S = \{\langle l, l, l, s, q, q, m\rangle, \langle l, s, q, s, q, m\rangle\} \tag{1}$$

Figure 2 shows a model recognized from this sample. Here $Action = ObservableAction$ and $h$ is the identity morphism (there are no duplicated events).

### 1.2 A Survey of Most Successful Process Mining Methods

1. Algorithms $\alpha, \alpha^{++}, \beta$ [23][26][16]. They are able to mine models having single tasks only. These algorithms base on finding causalities of tasks.
2. Genetic algorithms [19][12]. Models are transition matrices of Petri nets. A crossing operation exchanges fragments of the involved matrices.
3. Algorithms based on prefix trees [4]. The prefix tree is built for a given set of executions of a process. Learning corresponds to finding a congruence on the tree.
4. Algorithms based on regular expressions [2]. Models are regular expressions. Learning corresponds to a compression of the regular expression.
5. Statistical methods based on recursive neural networks [4]. The model is represented by a three-layer neural network. The hidden layer corresponds to the states of discovered automaton.
6. Statistical methods based on Markov chains [4], or Stochastic Activation Graphs [10]. The set of executions of a process is assumed to be a trajectory of a Markov chain; such a Markov chain is then constructed and turned into finite state machine by pruning transitions that have small probabilities or insufficient support.

Skeletal algorithms reassembles and generalizes the idea from algorithms based on prefix trees and regular expressions, and makes the task of finding a congruence structured and less ad hoc. We will elaborate more on skeletal algorithms in the next section.

### 1.3 Organization of the Paper

We assume that the reader is familiar with basic mathematical concepts. The paper is structured as follows. In section 2 we shall briefly recall some crucial for this paper mathematical concepts, and introduce skeletal algorithms. Section 3 describes our approach to process mining via skeletal algorithms. In section 4 we show some examples of process mining. We conclude the paper in section 5.

## 2 Skeletal Algorithms

Skeletal algorithms are a new branch of evolutionary metaheuristics focused on data and process mining. The basic idea behind the skeletal algorithm is to express a problem in

terms of congruences on a structure, build an initial set of congruences, and improve it by taking limited unions/intersections, until a suitable condition is reached. Skeletal algorithms naturally arise in the context of data/process mining, where the skeleton is the "free" structure on initial data and a congruence corresponds to similarities in the data. In such a context, skeletal algorithms come equipped with fitness functions measuring the complexity of a model.

Skeletal algorithms, search for a solution of a problem in the set of quotients of a given structure called the skeleton of the problem. More formally, let $S$ be a set, and denote by $Eq(S)$ the set of equivalence relations on $S$. If $i \in S$ is any element, and $A \in Eq(S)$ then by $[i]_A$ we shall denote the abstraction class of $i$ in $A$ — i.e. the set $\{j \in S\colon jAi\}$. We shall consider the following skeletal operations on $Eq(S)$:

1. Splitting

   The operation $split\colon \{0,1\}^S \times S \times Eq(S) \to Eq(S)$ takes a predicate $P\colon S \to \{0,1\}$, an element $i \in S$, an equivalence relation $A \in Eq(S)$ and gives the largest equivalence relation $R$ contained in $A$ and satisfying: $\forall_{j\in[i]_A} iRj \Rightarrow P(i) = P(j)$. That is — it splits the equivalence class $[i]_A$ on two classes: one for the elements that satisfy $P$ and the other of the elements that do not (Figure 3).
2. Summing

   The operation $sum\colon S \times S \times Eq(S) \to Eq(S)$ takes two elements $i, j \in S$, an equivalence relation $A \in Eq(S)$ and gives the smallest equivalence relation $R$ satisfying $iRj$ and containing $A$. That is — it merges the equivalence class $[i]_A$ with $[j]_A$ (see Figure 4).
3. Union

   The operation $union\colon S \times Eq(S) \times Eq(S) \to Eq(S) \times Eq(S)$ takes one element $i \in S$, two equivalence relations $A, B \in Eq(S)$ and gives a pair $\langle R, Q\rangle$, where $R$ is the smallest equivalence relation satisfying $\forall_{j\in[i]_B} iRj$ and containing $A$, and dually $Q$ is the smallest equivalence relation satisfying $\forall_{j\in[i]_A} iQj$ and containing $B$. That is — it merges the equivalence class corresponding to an element in one relation, with all elements taken from the equivalence class corresponding to the same element in the other relation (see Figure 5).
4. Intersection

   The operation $intersection\colon S \times Eq(S) \times Eq(S) \to Eq(S) \times Eq(S)$ takes one element $i \in S$, two equivalence relations $A, B \in Eq(S)$ and gives a pair $\langle R, Q\rangle$, where $R$ is the largest equivalence relation satisfying $\forall_{x,y\in[i]_A} xRy \Rightarrow x, y \in [i]_B \vee x, y \notin [i]_B$ and contained in $A$, and dually $Q$ is the largest equivalence relation satisfying $\forall_{x,y\in[i]_B} xQy \Rightarrow x, y \in [i]_A \vee x, y \notin [i]_A$ and contained in $B$. That is — it intersects the equivalence class corresponding to an element in one relation, with the equivalence class corresponding to the same element in the other relation (see Figure 6).

Furthermore, we shall assume that there is also a fitness function $\Delta\colon H(S) \to R$. The general template of skeletal algorithm is shown on figure 7. There are many things that can be implemented differently in various problems.

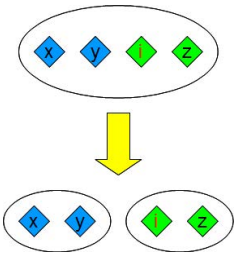



**Fig. 3.** Splitting

Equivalence class $[i]$ is split according to the predicate: blue elements satisfies the predicate, whereas green — not.

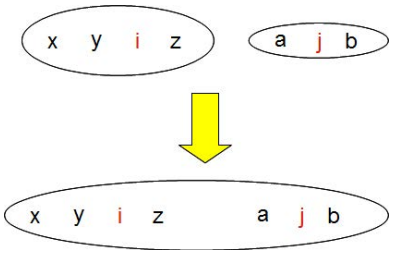



**Fig. 4.** Summing

Equivalence classes $[i]$ and $[j]$ are merged.

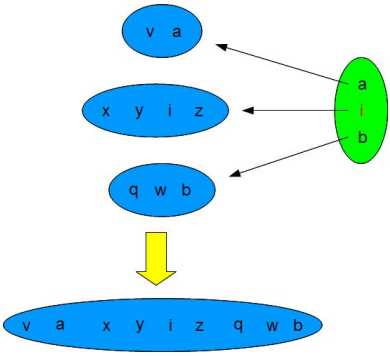



**Fig. 5.** Union

Merging equivalence classes in one relation along elements from the equivalence class $[i]$ in another relation.

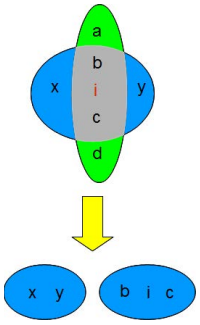



**Fig. 6.** Intersection

Spliting the equivalence class of $i$ in one relation along equivalence class of $i$ in another relation.

## 2.1 Construction of the Skeleton

As pointed out earlier, the skeleton of a problem should correspond to the "free model" build upon sample data. Observe, that it is really easy to plug in the skeleton some priori knowledge about the solution — we have to construct a congruence relation induced by the priori knowledge and divide by it the "free unrestricted model". Also, this suggests the following optimization strategy — if the skeleton of a problem is too big to efficiently apply the skeletal algorithm, we may divide the skeleton on a family of smaller skeletons, apply to each of them the skeletal algorithm to find quotients of the model, glue back the quotients and apply again the skeletal algorithm to the glued skeleton.

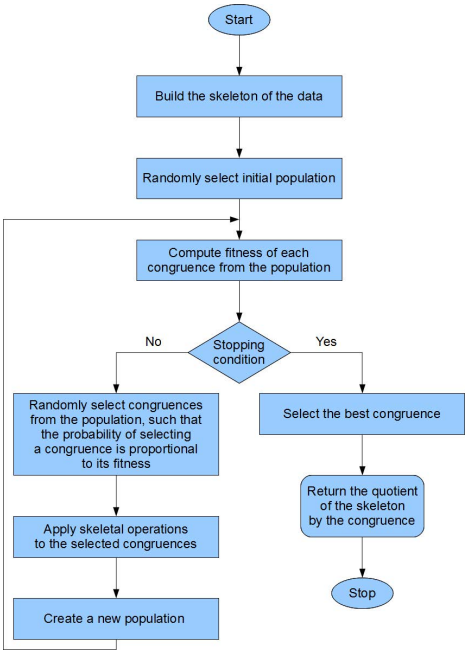



**Fig. 7.** Skeletal Algorithm

### 2.2 Construction of the Initial Population

Observe that any equivalence relation on a finite set $S$ may be constructed by successively applying $sum$ operations to the identity relation, and given any equivalence relation on $S$, we may reach the identity relation by successively applying $split$ operations. Therefore, every equivalence relation is constructible from *any* equivalence relation with $sum$ and $split$ operations. If no priori knowledge is available, we may build the initial population by successively applying to the identity relation both $sum$ and $split$ operations.

### 2.3 Selection of Operations

For all operations we have to choose one or more elements from the skeleton $S$, and additionally for a split operation — a splitting predicate $P\colon S \to \{0,1\}$. In most cases these choices have to reflect the structure of the skeleton — i.e. if our models have an algebraic or coalgebraic structure, then to obtain a quotient model, we have to divide the skeleton by an equivalence relation *preserving* this structure, that is, by a congruence. The easiest way to obtain a congruence is to choose operations that map congruences to congruences. Another approach is to allow operations that move out congruences from they class, but then "improve them" to congruences, or just punish them in the intermediate step by the fitness function.

### 2.4 Choosing Appropriate Fitness Function

Data nad process mining problems frequently come equipped with a natural fitness function measuring the total complexity of data given a particular model. One of the crucial conditions that such a function has to satisfy is the ability to easily adjust its value on a model obtained by applying skeletal operations.

### 2.5 Creation of Next Population

There is a room for various approaches. We have experimented most successful with the following strategy — append $k$-best congruences from the previous population to the result of operations applied in the former step of the algorithm.

## 3 Skeletal Algorithms in Process Mining

If we forget about additional information and attributes associated with an execution of a process, then the task of identifying a process reduces to the task of language recognition. The theory of language recognition that gives most negative results is "identification of a language in the limit" developed by Mark Gold [8]. The fundamental theorem published by Dan Angluin [1] says that a class of recursively indexed languages is (recursively) identifiable in the limit iff for every language $L$ from the class there exists an effectively computable finite "tell-tale" — that is: a subset $T$ of $L$ such that: if $T$ is a subset of any other language $K$ from the class, then $K \not\subseteq L$. An easy consequence of this theorem is that the set of regular languages is not identifiable in the limit. Another source of results in this context is the theory of PAC-learning developed by Leslie Valiant [17].

Although these results are fairly interesting, in the context of process mining, we are mostly given a very small set of sample data, and our task is to find the most likely hypothesis — the question: "if we were given sufficiently many data, would it have been possible to find the best hypothesis?" is not really practical.

### 3.1 Probabilistic Languages

A probabilistic language $L$ over an alphabet $\Sigma$ is any subset of $\Sigma^* \times [0,1]$ that satisfies the following condition: $\sum_{\langle w,p\rangle \in L} p = 1$. Note that probabilistic languages over $\Sigma$ are the same as probability distributions over $\Sigma^*$.

A probabilistic finite state automaton is a quadruple $A = \langle \Sigma, S, l, \delta \rangle$, where:

- $\Sigma$ is a finite set called the "alphabet of the automaton"
- $S$ is a finite set of states
- $l$ is a labeling function $S \to \Sigma \cup \{start, end\}$ such that $l^{-1}[start] = \{s_{start}\} \neq \{s_{end}\} = l^{-1}[end]$; state $s_{start}$ is called "the initial state of the automaton", and $s_{end}$ "the final state of the automaton"
- $\delta$ is a transition function $S \times S \to [0,1]$ such that:
  - $\forall_{s \in S} \sum_{q \in S} \delta(s,q) = 1$
  - $\forall_{s \in S} \delta(s, s_{start}) = 0$
  - $\delta(s_{end}, s_{end}) = 1$

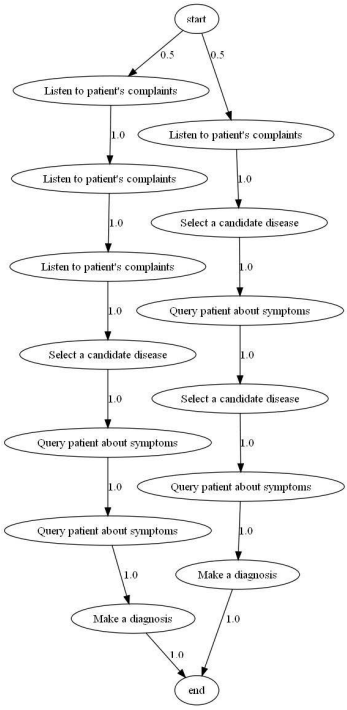



**Fig. 8.** Skeletal model of sample 1

A trace of an automaton $A$ starting in a state $s_0$ and ending in a state $s_k$ is a sequence $\langle s_1, \ldots, s_k \rangle \in S^*$. A full trace of an automaton is a trace starting in $s_{start}$ and ending in $s_{end}$.

In our setting models correspond to probabilistic finite automata, the distributions are induced by the probabilities of full traces of the automata, and morphisms map states to they actions (i.e. labels).

### 3.2 Skeleton

Given a list of sample data $K : n = \{0, \cdots, n-1\} \to \Sigma^*$, by a skeleton of $K$ we shall understand the automaton: $skeleton(K) = \langle \Sigma, S, l, \delta \rangle$, where:

- $S = \{\langle i, k \rangle : i \in n, k \in \{1, \ldots, |K(i)|\}\} \cup \{-\infty, \infty\}$
- $l(-\infty) = start, l(\infty) = end, l(\langle i, k \rangle) = K(i)_k$, where the subscript $k$ indicates the $k$-th element of the sequence
- $\delta(-\infty, \langle i, 1 \rangle) = 1, \delta(\infty, \infty) = 1, \delta(\langle i, |K(i)| \rangle, \infty) = 1, \delta(\langle i, k \rangle, \langle i, k+1 \rangle) = 1$

So the skeleton of a list of data is just an automaton corresponding to this list enriched with two states — initial and final. This automaton describes the situation, where all actions are different. Our algorithm will try to glue some actions that give the same output (shall search for the best fitting automaton in the set of quotients of the skeletal automaton). Figure 8 shows the skeletal automaton of the sample 1 from section 1.

Given a list of sample data $K\colon n \to \Sigma^*$, our search space $Eq(S)$ consists of all equivalence relations on $S$.

### 3.3 Skeletal Operations

1. Splitting
   For a given congruence $A$, choose randomly a state $\langle i,k\rangle \in skeleton(K)$ and make use of two types of predicates
   - split by output — $P(\langle j,l\rangle) = 1 \Leftrightarrow \exists_{\langle i',k'\rangle \in [\langle i,k\rangle]_A} \delta(\langle j,l\rangle, \langle i',k'\rangle)$
   - split by input — $P(\langle j,l\rangle) = 1 \Leftrightarrow \exists_{\langle i',k'\rangle \in [\langle i,k\rangle]_A} \delta(\langle i',k'\rangle, \langle j,l\rangle)$
2. Summing
   For a given congruence $A$, choose randomly two states $\langle i,k\rangle, \langle j,l\rangle$ such that $l(\langle i,k\rangle) = l(\langle j,l\rangle)$.
3. Union/Intersection
   Given two skeletons $A, B$ choose randomly a state $\langle i,k\rangle \in skeleton(K)$.

Let us note that by choosing states and predicates according to the above description, all skeletal operations preserve congruences on $skeleton(K)$.

### 3.4 Fitness

Let $v_0\colon v = \langle v_0, v_1, \cdots, v_k\rangle$ be a trace of a probabilistic automaton. Assuming that we start in node $v_0$, the probability of moving successively through nodes $v_1, \cdots, v_k$ is

$$P(v|v_0) = \prod_{i=1}^{k} \delta(v_{i-1}, v_i)$$

and it give us a probability distribution on $S^k$:

$$P(v) = \sum_{v_0 \in S} \mu(v_0) P(v|v_0)$$

where $\mu$ is any probability distribution on the states $S$ of the automaton. If we choose for $\mu$ a probability mass distribution concentrated in a single node $v_0$, then $P(v)$ would depend multiplicatively on probabilistic transitions $P(v_{i-1}, v_i)$. In this case any local changes in the structure of the automaton (like splitting or joining nodes) give multiplicative perturbations on the probability $P(v)$, so it is relatively easy (proportional to the number of affected nodes) to update the complexity of $v$.

Consider any full trace $v = \langle v_0 = start, v_1, \cdots, v_k = end\rangle$ of an automaton. According to our observation, we may associate with it the following probability:

$$P(v) = \prod_{i=1}^{k} \delta(v_{i-1}, v_i) = \prod_{x \in S} \prod_{a \in S} \delta(x,a)^{|\{i\colon\, x = v_i \wedge a = v_i + 1\}|}$$

where for every $x$ the term $\prod_{a \in S} \delta(x,a)^{|\{i:\ x=v_i \wedge a=v_i+1\}|}$ depends only on the number of pass to the state $a$. Hence, we may restrict our analysis to single states.

Let $s$ be such a state with $l$ output probabilistic transitions $a_1, \cdots, a_l$, and let us assume that the probability of passing the $j$-th arrow is $p_j$. Then the probability of consecutively moving through arrows $x = \langle a_{i_1}, \cdots, a_{i_k} \rangle$ when visiting node $s$ is:

$$p^s(x) = \prod_{j=1}^{k} p_{i_j} = \prod_{j=1}^{l} p_j^{c_j}$$

where $c_j$ is the number of occurences of $a_j$ in $x$. Thus, given a sample $x$ and a probabilistic node $s$ the optimal length of a code describing $x$ is about

$$log(\frac{1}{p^s(x)})$$

and the shortest code is achieved for $s$ having probabilities

$$p_1 = \frac{c_1}{k}, \cdots, p_k = \frac{c_l}{k}$$

Now, let us assume that we do not know probabilities at $s$. Then any code describing $x$ via $s$ has to contain some information about these probabilities. A "uniform approach" would look like follows: for a given sample $x$ chose the optimal probability node $s_x$, then $opt(x) = p^{s_x}(x)$ is not a probability on $l^k$ as it does not sum up to 1 (i.e. it does not contain information about choosing appropriate hypothesis $s_x$); however

$$\begin{aligned} mdl(x) &= \frac{opt(x)}{\sum_{x \in l^k} opt(x)} \\ &= (\sum_{r_1+\cdots+r_l=k} \binom{k}{r_1, \cdots, r_l} \prod_{i=1}^{l} r_i^{r_i})^{-1} \prod_{i=1}^{l} c_i^{c_i} \\ &= m \prod_{i=1}^{l} c_i^{c_i} \end{aligned} \tag{2}$$

is, where $\binom{k}{r_1,\cdots,r_l}$ is the multionomial $k$ over $r_1, \cdots, r_l$. One may take another approach based on Bayesian interpretation. Let us fix a meta-distribution $q$ on all probabilistic nodes $s$ having the same output arrows. This distribution chooses probabilities $p_1, \cdots, p_l$, that is — non-negative real numbers such that $p_1 + \cdots + p_l = 1$ — then for a given sample $x$ chose a node $s_{p_1,\cdots,p_l}$ with probability $q(s_{p_1,\cdots,p_l})$ and describe $x$ according to that node:

$$bayes(x) = \int_{p_1+\cdots+p_l=1, p_i \geq 0} p^{s_{p_1,\cdots,p_l}}(x) q(s_{p_1,\cdots,p_l})$$

If $q$ is a uniform distribution, then

$$\begin{aligned} bayes(x) &= \frac{\int_{p_1+\cdots+p_l=1, p_i \geq 0} \prod_{i=1}^{l} p_i^{c_i}}{Vol(\Delta_l)} \\ &= \frac{\Gamma(l) \prod_{i=1}^{l} \Gamma(c_i+1)}{\Gamma(\sum_{i=1}^{l}(c_i+1))} \\ &= \frac{\Gamma(l)}{\Gamma(k+l)} \prod_{i=1}^{l} c_i^{\underline{c_i}} \\ &= b \prod_{i=1}^{l} c_i^{\underline{c_i}} \end{aligned} \tag{3}$$

So, $mdl(x) = m \prod_{i=1}^{l} c_i^{c_i}$ and $bayes(x) = b \prod_{i=1}^{l} c_i^{\underline{c_i}}$, where $m, b$ are constants making $mdl$ and $bayes$ probability distributions. In fact, these distributions are really close — by using Striling's formula

$$n^{\underline{n}} \approx \sqrt{2\pi n}(\frac{n}{e})^n$$

we have

$$bayes(x) \approx b' \prod_{i=1}^{l} c_i^{c_i+\frac{1}{2}}$$

where $b' = be^{-n}(2\pi)^{l/2}$ is just another constant. We shall prefer the Bayesian distribution as it is much easier to compute and update after local changes, but we should be aware that it slightly more favors random sequences than the optimal (in the sense of minimum regret) distribution.

The total distribution on traces is then given by:

$$bayes^{trace}(v) = \prod_{s \in S} bayes^s(v \downarrow s)$$

where $bayes^s$ is the Bayesian distribution corresponding to the node $s$ and $v \downarrow s$ is the maximal subsequence of $v$ consisting of elements directly following $s$. And the corresponding complexity of $v$ is:

$$comp(v) = -\sum_{s \in S} log(bayes^s(v \downarrow s))$$

Although this complexity assumes that we do not know the exact probabilities of the automaton, it also assumes that we know all its other properities. Our research showed that the other aspects of the automaton are best described with two-parts codes. Thus, the fitness function for a congruence $A$ on the skeleton of sample data $K \colon n \to \Sigma^*$

would be proportional to the sum of the description (neglecting probabilities) of the quotient model $skeleton(K)/A$ and complexities of each $K(i)$ according to that model:

$$\Delta(A) = -|skeleton(K)/A| - \sum_{i=0}^{n-1} comp^{skeleton(K)/A}(K(i))$$

where $|skeleton(K)/A|$ may be tuned for particular samples. Our experience showed that choosing

$$clog(|S|)|\{\langle x, y\rangle \in S \times S \colon \delta(x, y) > 0\}|$$

for a small constant $c > 1$ behaves best.

## 4 Examples

### 4.1 Non-deterministic Automata

Given a non-deterministic automata like on figure 9 we generate sample of $n$ words by moving through each arrow outgoing from a state with equal probabilities. Figure 10 shows discovered model after seeing 4 samples and Figure 11 after seeing 16. Note, that the automaton is rediscovered with a great precision after seeing a relatively small sample data.

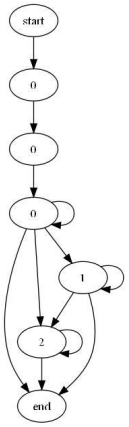



**Fig. 9.** Nondeterministic automaton

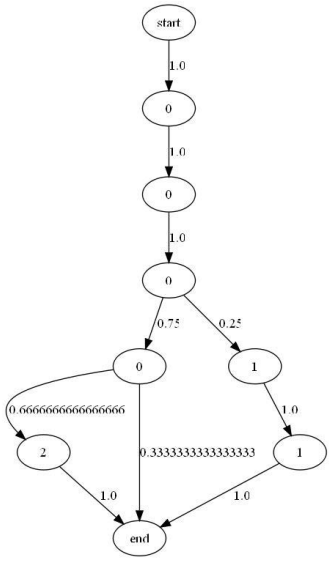



**Fig. 10.** Model after seeing 4 samples

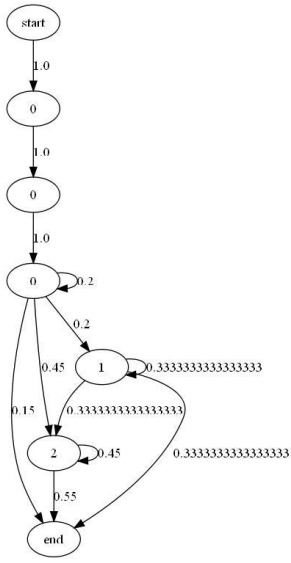



**Fig. 11.** Model after seeing 16 samples

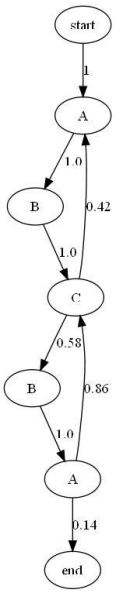



**Fig. 12.** Model discovered from sample L1

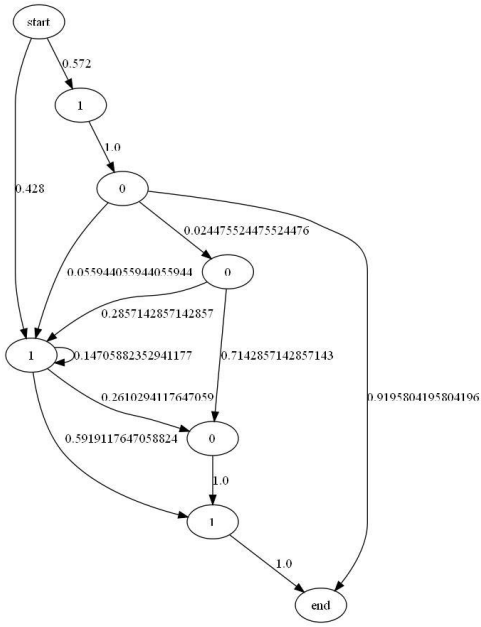



**Fig. 13.** Prime numbers

### 4.2 Testing Sample

In this example we use samples from [4]:

$$\begin{aligned} L1 = {} & A, B, C, A, B, C, B, A, C, B, A, C, A, \\ & B, C, B, A, C, B, A, C, A, B, C, B, A, \\ & C, A, B, C, A, B, C, B, A, C, B, A \end{aligned} \tag{4}$$

$$\begin{aligned} L2 = {} & A, B, C, D, C, E, F, G, H, G, I, J, \\ & G, I, K, L, M, N, O, P, R, F, G, I, \\ & K, L, M, N, O, P, Q, S \end{aligned} \tag{5}$$

Figures 12 and 14 show models discovered from sample $L1$ and $L2$ respectively. Model 12 corresponds to the model mined by KTAIL method, whereas model 14 outperforms underfitted RNET, MARKOV and KTAIL.

### 4.3 Prime Numbers

In this example we show how skeletal algorithms can learn from a probabilistic source $p$ that does not correspond to any model. We define $p$ to be non-zero only on prime numbers, and such that the probability for a given prime number is proportional to its numbers of bits in binary representation. Figure 13 shows discovered automaton from $500$ samples. Observe that it quite accuratly predicts all $5$-bits prime numbers.

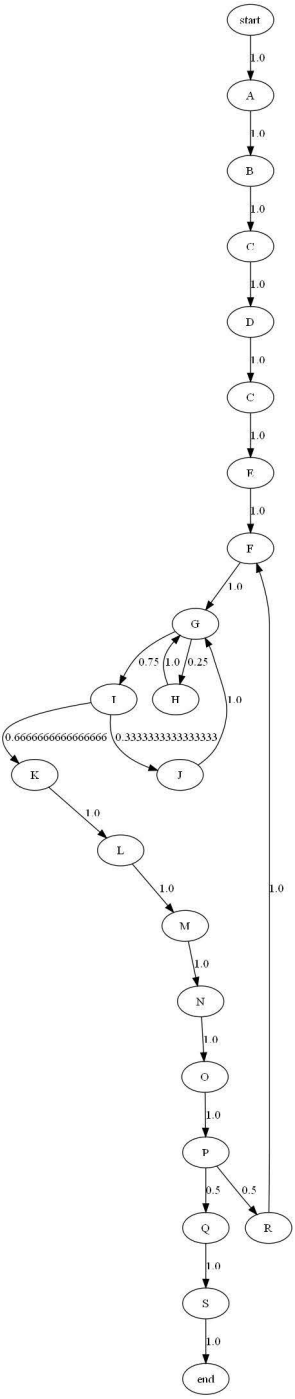



**Fig. 14.** Model discovered from sample L2

## 5 Conclusions

In this paper we introduced a new kind of evolutionary method — "skeletal algorithm", especially suitable in the context of data and process mining. In such a context "skeletal algorithms" come often equipped with a natural fitness function measuring the complexity of a model. We showed a sample application of "skeletal algorithms" to process mining and examined two naturally fitness functions — one based on Minimum Description Length Principle, and another based on Bayesian Interpretation. Although, obtained results are really promising, there are issues that should be addressed in future works. The main concern is to extend the concept of models — our models base on probabilistic automata, and so the algorithm is not able to mine nodes corresponding to parallel executions of a process (i.e. AND-nodes). Also, we are interested in applying various optimization techniques and investigate more industrial data.

## References

1. Angluin, D.: Inductive inference of formal languages from positive data. Information and Control 42 (1980)
2. Brazma, A.: Efficient Algorithm for Learning Simple Regular Expressions from Noisy Examples. In: Arikawa, S., Jantke, K.P. (eds.) AII 1994 and ALT 1994. LNCS, vol. 872, pp. 260–271. Springer, Heidelberg (1994)
3. Bremermann, H.J.: Optimization through evolution and recombination. In: Yovitts, M.C., et al. (eds.) Self-Organizing Systems 1962, pp. 93–106. Spartan Books, Washington (1962)
4. Cook, J., Woolf, A.: Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology 7(3) (1998)
5. de Medeiros, A., van Dongen, B., van der Aalst, W., Weijters, A.: Process mining: Extending the alpha-algorithm to mine short loops. BETA Working Paper Series, Eindhoven University of Technology, Eindhoven (2004)
6. Friedberg, R.M.: A learning machines part i. IBM Journal of Research and Development 2 (1956)
7. Friedberg, R.M., Dunham, B., North, J.H.: A learning machines part ii. IBM Journal of Research and Development 3 (1959)
8. Gold, E.: Language identification in the limit. Information and Control 10 (1967)
9. Grunwald, P.D., Rissanen, J.: The minimum description length principle. In: Adaptive Computation and Machine Learning Series. The MIT Press (2007)
10. Herbst, J.: A Machine Learning Approach to Workflow Management. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 183–194. Springer, Heidelberg (2000)
11. Holland, J.H.: Adaption in natural and artificial systems. The University of Michigan Press, Ann Arbor (1975)
12. Medeiros, A., Weijters, A., van der Aalst, W.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery 14(2) (2007)
13. Przybylek, A.: The Integration of Functional Decomposition with UML Notation in Business Process Modelling. In: Advances in Information Systems Development, pp. 85–99. Springer (2008)
14. Przybylek, M.: Skeletal Algorithms. In: International Conference on Evolutionary Computation Theory and Applications 2011, pp. 80–89 (2011)

15. Rechenberg, I.: Evolutions strategie —- optimierung technischer systeme nach prinzipien der biologischen evolution. PhD thesis (1971), reprinted by Fromman-Holzboog (1973)
16. Ren, C., Wen, L., Dong, J., Ding, H., Wang, W., Qiu, M.: A novel approach for process mining based on event types. In: IEEE SCC 2007, pp. 721–722 (2007)
17. Valiant, L.: A theory of the learnable. Communications of the ACM 27 (1984)
18. van der Aalst, W.: Process mining: Discovery, conformance and enhancement of business processes. Springer (2011)
19. van der Aalst, W., de Medeiros, A.A., Weijters, A.: Process equivalence in the context of genetic mining. BPM Center Report BPM-06-15, BPMcenter.org (2006a)
20. van der Aalst, W., Pesic, M., M.S.: Beyond process mining: From the past to present and future. BPM Center Report BPM-09-18, BPMcenter.org (2009)
21. van der Aalst, W., ter Hofstede, A.: Workflow patterns: On the expressive power of (petri-net-based) workflow languages. BPM Center Report BPM-02-02, BPMcenter.org (2002)
22. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. BPM Center Report BPM-00-02, BPMcenter.org (2000)
23. van der Aalst, W., van Dongen, B.F.: Discovering Workflow Performance Models from Timed Logs. In: Han, Y., Tai, S., Wikarski, D. (eds.) EDCIS 2002. LNCS, vol. 2480, pp. 45–63. Springer, Heidelberg (2002)
24. van der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: Discovering process models from event logs. BPM Center Report BPM-04-06, BPMcenter.org (2006b)
25. Weijters, A., van der Aalst, W.: Process mining: Discovering workflow models from event-based data. In: Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence, pp. 283–290. Springer, Maastricht (2001)
26. Wen, L., Wang, J., Sun, J.: Detecting Implicit Dependencies Between Tasks from Event Logs. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 591–603. Springer, Heidelberg (2006)
27. Wynn, M., Edmond, D., van der Aalst, W., ter Hofstede, A.: Achieving a general, formal and decidable approach to the or-join in workflow using reset nets. BPM Center Report BPM-04-05, BPMcenter.org (2004)

# Part II
# Fuzzy Computation Theory and Applications

# Handling Fuzzy Models in the Probabilistic Domain

Manish Agarwal*, Kanad K. Biswas, and Madasu Hanmandlu

Indian Institute of Technology, New Delhi, India
{magarwal,kkb}@cse.iitd.ernet.in, mhmandlu@ee.iitd.ernet.in

**Abstract.** This chapter extends the fuzzy models to the probabilistic domain using the probabilistic fuzzy rules with multiple outputs. The focus has been to effectively model the uncertainty in the real world situations using the extended fuzzy models. The extended fuzzy models capture both the aspects of uncertainty, vagueness and random occurence. We also look deeper into the concepts of fuzzy logic, possibility and probability that sets the background for laying out the mathematical framework for the extended fuzzy models. The net conditional probabilistic possibility is computed that forms the key ingredient in the extension of the fuzzy models. The proposed concepts are well illustrated through two case-studies of intelligent probabilistic fuzzy systems. The study paves the way for development of computationally intelligent systems that are able to represent the real world situations more realistically.

**Keywords:** Probabilistic fuzzy rules, Probability, possibility, Fuzzy models, Decision making.

## 1 Introduction

Zadeh [1] first coined the term possibility to represent the imprecision in information. This imprecision is quite different from the frequentist uncertainty represented by well developed probabilistic approach. But if we could appreciate the real world around us, there is a constant interplay between probability and possibilityeven though the two represent different aspects of uncertainty. Hence, if not all, in many a situation, the two are intricately interwoven in the linguistic representation of a situation or an event by a human brain. Often, it is possible to infer probabilistic information from possibilistic one and vice versa. Even though they are dissymmetrical and treated differently in literature, there is a need to make an effort towards exploring a unifying framework for their integration. We feel that these two different, yet complimentary formalisms can better represent practical situations, going hand in hand.

Besides the vast potential of this study in more closely representing the real world, we are also motivated by its roots in philosophy. Non-determinism is almost a constant feature in nature, and together probability and possibility can go farther in representing the real world situations. Even though, probability and possibility represent two different forms of uncertainty and are not symmetrical, but still both are closely related, and often needs to be transformed into one other, to achieve computational simplicity and efficiency. This transformation would pave the way for simpler methods for the

* Corresponding author.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 137–151.
DOI: 10.1007/978-3-642-35638-4_10 

computation of net possibility. The intelligent systems utilizing these transformations would represent the requirements and situations of the real world more truly and accurately. They would also be more computationally efficient in terms of speed, storage and accuracy in processing of the uncertain information.

Such transformations bridge two different facets of uncertainty, the probabilistic uncertainty and the imprecision on account of vagueness or lack of knowledge. Dubois et al. [2], [3] analyzed the transformations between the two and judged the consistency in the two representations.

This work is concerned with devising a novel approach for application of some of the research results to the field of fuzzy theory under probabilistic setting, and using the same to enhance the existing fuzzy models to better infer the value of possibility in the light of probabilistic information available. It also relooks at the relevant results along with their interpretations in the context of probabilistic fuzzy theory. This chapter basically addresses the following issues:

1. To amalgamate the field of fuzzy theory with the probability theory and to discover the possible linkages or connections between these two facets of uncertainty.
2. To apply the probabilistic framework on the existing fuzzy models for imparting the practical utility to them.
3. To devise an approach to calculate the output of the probabilistic fuzzy models and compare it with the outputs of conventional fuzzy rules.

This chapter is organized as follows. Section 2 explores the relationship between probability and possibility and sets the background. Section 3 presents mathematical relations to calculate the output of probabilistic fuzzy rules (PFRs). The utility and advantages of PFR are also discussed. An algorithm for computation of the net conditional possibility from probabilistic fuzzy rules is also presented. Section 4, illustrates the concepts by two case studies. Finally, Section 5 gives the conclusions and the scope of further research in the area.

## 2 The Two Facets of Uncertainty: A Relook

The possible links between the two facets of uncertainty: probability and possibility are explored on the basis of the key contributions in the area.

The celebrated example of Zadeh [1], "Hans ate X eggs for Breakfast", illustrates the differences and relationships between probability and possibility in one go. The possibility of Hans eating 3 eggs for breakfast is 1 whereas the probability that he may do so might be quite small, e.g. 0.1. Thus, a high degree of possibility does not imply a high degree of probability; though if an event is impossible it is bound to be improbable. This heuristic connection between possibility and probability may be called the possibility/probability consistency principle, stated as: If a variable x takes values $u_1, u_2, \ldots, u_n$ with respective possibilities $\Pi = (\pi_1, \pi_2, \ldots, \pi_n)$ and probabilities $P = (p_1, p_2, .., p_n)$ then the degree of consistency of the probability distribution P with the possibility distribution II is expressed by the arithmetic sum as

$$\gamma = \pi_1 p_1 + \pi_2 p_2 + \ldots + \pi_n p_n \tag{1}$$

Note that the above principle is not a precise law or a relationship that is intrinsic to the concepts of possibility and probability; rather it is an approximate formalization of the heuristic observation that a lessening of the possibility of an event tends to lessen its probability, not vice-versa. In this sense, the principle is applicable to situations in which we know the possibility of a variable x rather than its probability distribution. This principle forms the most conceptual foundation of all the works in the direction of probability/possibility transformations having wide practical applications [4].

Having deliberated on the consistency principle, we will look into: (i) Basic difference between possibility and probability, (ii) Inter-relation between possibility and probability and vice-versa, (iii) Infer probability from possibility and vice-versa, and (iv) Transformation of probability to possibility and vice-versa, with a view to tackle real life problems involving both probabilistic and possibilistic information.

### 2.1 Basic Difference between Possibility and Probability

In the perspective of the above example by Zadeh, possibility is the degree of ease with which Hans may eat u eggs whereas probability is the chances of actual reality; there may be significant difference between the two. This difference is now elucidated by noting that the possibility represents likelihood of a physical reality with respect to some reference whereas the probability represents the occurrences of the same [2]. To put it mathematically,

$$\Pi(A) = sup_{u \in A} \pi_x(u) \tag{2}$$

where A is a non fuzzy subset of U, $\Pi$ is possibility distribution of x, $\Pi(A)$ denotes the possibility measure of A in [0,1], $\pi_x(u)$ is the possibility distribution function of $\Pi_x$.

Let A and B be arbitrary fuzzy subsets of U. In view of (1) we can write that

$$\pi(A \cup B) = \pi(A) \vee \pi(B) \tag{3}$$

The corresponding relation for probability is written as

$$P(A \cup B) \leq P(A) + P(B) \tag{4}$$

### 2.2 Inter-relation between Possibility and Probability

Any pair of dual necessity/possibility functions $(N, \Pi)$ can be interpreted as the upper and lower probabilities induced from specific convex sets of probability functions.

Let $\pi$ be a possibility distribution inducing a pair of functions $(N, \Pi)$. Then we define

$$\mathcal{P}(\pi) = \{P, \forall A measurable, N(A) \leq P(A)\} = \{P, \forall A measurable, P(A) \leq \Pi(A)\} \tag{5}$$

The family,$\mathcal{P}(\pi)$, is entirely determined by the probability intervals it generates. Any probability measure is said to be consistent with the possibility distribution, $\pi$[2],[5]. That is

$$sup_{P \in \mathcal{P}(\pi)} P(A) = \Pi(A) \tag{6}$$

A relevant work in this direction was carried out in [6]. It is shown that the imprecise probability setting is capable of capturing fuzzy sets representing linguistic information.

### 2.3 Inference of Probability from Possibility and Vice-versa

In [1-3], [7] degrees of possibility can be interpreted as the numbers that generally stand for the upper probability bounds. The probabilistic view is to prepare interpretive settings for possibility measures. This enables us to deduce a strong interrelation between the two. Zadeh's consistency principle gives the degree of consistency or interrelation between possibility and probabilty associated with an event. From [14], and from the above properties of possibility and necessity measures, we know that maximizing the degree of consistency brings about two strong restrictive conditions

$$P_{oss}(A) < 1 \Rightarrow N_{ecc}(A) = 0; \qquad N_{ecc}(A) > 0 \Rightarrow P_{oss}(A) = 1 \tag{7}$$

There exists a relationship between probability and possibility. In the example in [1], possibility refers to the ease of eating eggs, and probability refers to the actual reality. Unless something is possible, it cant be probable and if something is more possible, it should be more probable. We take two more real life examples.

1. People who do regular exercise are likely to live long.
2. It is very cloudy. It is likely to rain heavily.

In both of these examples, possibility of action (of exercising, or cloudiness) is contributing towards inferring the probability of living long and rain. People who do regular exercise are certainly having better chances of a long life. Here, while long is a fuzzy and qualitative in nature, chances of having a long life is quantitative and probabilistic in nature. Similarly, if it is very cloudy, chances of rain are more than what it would have been under less cloudy conditions. So more possibility of cloudiness is associated with more probability of rain.

### 2.4 Transformation from Probability to Possibility

Any transformation from probability to possibility must comply with the following three basic principles [3].

1. Possibility-probability consistency: $\gamma = \pi_1 p_1 + \pi_2 p_2 + \ldots + \pi_n p_n$
2. Ordinal faithfulness: $\pi(u) > \pi(u')\ iff\ p(u) > p(u')$
3. Informativity: Maximization of information content of $\pi$

If P is a probability measure on a finite set U, statistical in nature [7], then, for a subset, E of U, its possibility distribution on U, $\Pi_E(u)$ is given by

$$\Pi_E(u) = \begin{cases} 1 & \text{if } u \in E, \\ 1 - P(E) & \text{otherwise} \end{cases} \tag{8}$$

Also $\Pi_E(A) \geq P(A), \forall A \subseteq U$.

In other words, $\Pi_E = x \in E$ with the confidence at least P(E). In order to have a meaningful possibility distribution, $\Pi_E$, care must be taken to balance the nature of complimentary ingredients in (8), i.e. E must be narrow and P (E) must be high.

There are quite a few ways, in which one can do it. The one used in [7] chooses a confidence threshold $\alpha$ so as to minimize the cardinality of E such that $P(E) \geq \alpha$.

Conversely, cardinality of E can be fixed and P (E) maximized. This way, a probability distribution P can be transformed into a possibility distribution $\pi^P$ [7] as shown here. Take $\Pi$ as the probability distribution on U and $X = \{x_1, x_2, .., x_n\}$ such that $\Pi = P(\{x_i\})$. Similarly possibility distribution $\Pi_i = \Pi(\{x_i\})$ and $p_1 \geq p_2 \geq \ldots \geq p_n$, then we have

$$\pi_i^P(u) = \sum_{j=i}^{n} p_j, \ \forall i = 1, n \tag{9}$$

For a continuous case, if the probability density function so obtained is continuous unimodal having bounded support $[a, b]$, say p, then p is increasing on $[a, x_0]$ and decreasing on $[x_0, b]$, where $x_0$ is the modal value of p. This set is denoted as D [8].

## 3 Probabilistic Fuzzy Modelling

A probabilistic fuzzy rule (PFR), first devised by Meghdadi [9], is an appropriate tool to represent a real world situation possessing both the features of uncertainty. In such cases, we often observe that for a set of inputs, there may be more than one possible output. The probability of occurrence of the outputs may be context dependent. In a fuzzy rule, there being only a single output, we are unable to accommodate this feature of the real world multiple outputs with different probabilities. This ability is afforded with PFR. The PFR with multiple outputs and their probabilities is defined as:
Rule $R_q$:
If x is $A_q$ then y is $O_1$ with probability $P_1$
& ...
& y is $O_j$ with probability $P_j$
& ...
& y is $O_q$ with probability $P_n$

$$where P_1 + P_2 + \ldots + P_n = 1 \tag{10}$$

Given the occurrence of the antecedent (an event) in (10), one of the consequents (output) would occur with the respective probability of occurrence, . Therefore, y is associated with both qualitative (in terms of membership function, O) and quantitative (in terms of probability of occurrence, P) information. Therefore y is both a stochastic and fuzzy variable at the same time. The real outcome is a function of the probability, while the quality of an outcome is a function of the respective membership function. The probability of an event is having a larger role to play since it is the one that determines the occurrence of the very event. More the probability of an output event, more are the chances of its certainty which in turn gives rise to the respective possibility of the event (in terms of membership function) determining the quality of the outcome.

The above example illustrates the fact that both these measures of uncertainty (probability and possibility) are indispensable in fuzzy modelling of real world multi-criteria decision making, and may lead to incomplete and misleading result if one of them is ignored. So the original fuzzy set theory, if backed by probability theory could go miles in better representing the decision making problems and deriving realistic solutions.

Here, one question that naturally arises is: how about treating probabilities in the antecedents? This aspect is taken into account by having more than one fuzzy rule and probabilistic outcome in the consequent which is sufficient to handle the frequentist uncertainty in the probabilistic fuzzy event. For example in (10), the antecedent could be :

$$\text{If } x_1 \text{ is } \mu_1 \text{ and } x_2 \text{ is } \mu_2$$

Now, the range of probable values of occurrence of inputs is either $Input_1$ or $Input_n$ etc. Thus for each occurrence of an antecedent condition, there is a corresponding probabilistic fuzzy consequent event in (10).

In this study, we would be considering similar PFRs with the same structure for a probabilistic fuzzy system under consideration. That is, any two PFRs would have the same order of probabilistic outputs.

$$\forall j, q, q' : O_{q_j} = O_{q'_j} = O_j \tag{11}$$

where, $q$ and $q'$ represent two PFRs $A_q$ and $A'_q$.
$O_{q_j}$ is $j_{th}$ output in $q_{th}$ rule; $O_{q'_j}$ is $j_{th}$ output in $q'_{th}$ rule $O_j$ is $j_{th}$ output that remains the same in any PFR.

The mathematical framework follows from [10]. Assuming two sample spaces, say X and Y, in forming the fuzzy events $A_i$ and $O_j$ respectively, the following equations hold good,

$$\forall x : \sum_i \mu_i(x) = 1, \forall y : \sum_j \mu_j(y) = 1 \tag{12}$$

If the above conditions are satisfied then X and Y are said to be well defined.

### 3.1 Conditional Output Probabilities with Given Fuzzy Antecedents

Given a set of S samples $(x_s, y_s)$, s = 1, ..., S from two well-defined sample spaces X, Y, the probability of $A_i$ can be calculated as

$$P(A_i) = \overline{f}_{A_i} = \frac{f_{A_i}}{S} = \frac{1}{S} \sum_{x_s} \mu_i(x_s) = \overline{\mu}_i \tag{13}$$

where, $A_i$ is the antecedent fuzzy event, which leads to one of the consequent events $O_1, \ldots, O_n$ to occur.
$\overline{f}_{A_i}$: Relative Frequency of fuzzy sample values $\mu_i(x_s)$ for the fuzzy event $A_i$
$f_{A_i}$ : Absolute Frequency of fuzzy sample values $\mu_i(x_s)$ for the fuzzy event $A_i$

The fuzzy conditional probability is given by,

$$P(O_j|A_i) = \frac{P(O_j \cap A_i)}{P(A_i)} \approx \frac{\sum_s \mu_j(y_s)\mu_i(x_s)}{\sum_s \mu_i(x_s)} \tag{14}$$

The density function, $p_j(y)$ can be approximated using the fuzzy histogram [11] as follows:

$$p_j(y) = \frac{P(O_j)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} \tag{15}$$

where denominator $\int_{-\infty}^{\infty} \mu_j(y)dy$ is a scaling factor.

### 3.2 Conditional Output Probability for Arbitrary Fuzzy Input

A input vector x, activates the firing of multiple fuzzy rules, q, with multiple firing rates $\mu_q(x)$, such that $\sum_q \mu_q(x) = 1$. In case this condition is true for a single rule, only one of the consequents $O_q$ will occur with the conditional probability $P(O_j|x)$. In the light of (14) and (15), we obtain

$$P(O_j|x) = \sum_{q=1} \frac{\mu_q(x)P(O_j|A_q)}{\int_{-\infty}^{\infty} \mu_j(x)dx} \tag{16}$$

Extending the conditional probability $P(O_j|x)$ to estimate the overall conditional probability density function p(y—x), using (15), we get

$$p(y|x) = \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} \tag{17}$$

where, proabilities $P(O_j|x)$ is calculated using (16).
In view of (6) and (9) we obtain,

$$\pi(y|x) = \sum_j \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} \tag{18}$$

This value for conditional probabilistic possibility can be used in the existing fuzzy models to obtain the defuzzified probabilistic output.

### 3.3 Obtaining Probabilistic Output

We now compute the probabilistic output using the conditional probabilistic possibility in the existing fuzzy models.

**Mamdani-Larsen Model**
Consider a rule of this model as: Rule q: If x is $A_q$ then y is $B_q$.

Here, fuzzy implication operator maps fuzzy subsets from the input space $A^q$ to the output space $B^q$ (with membership function $\varphi(y)$) [1] and generates the fuzzy output $B^q$ with the fuzzy membership.
Rule q: $\phi(y) = \mu(x) \rightarrow \varphi(y)$
The output fuzzy membership is:

$$\phi^0(y) = \phi^1(y) \vee \phi^2(y) \vee \ldots \vee \phi^k(y) \tag{19}$$

In Mamdani-Larsen (ML) model, the output of rule q is represented by $B^q(b_q, v_q)$, with centroid $b_q$ and the index of fuzziness $v_q$ given by

$$v_q = \int_y \phi(y)dy \tag{20}$$

$$b_q = \frac{\int_y y\phi(y)dy}{\int_y \phi(y)dy} \tag{21}$$

where $\phi(y)$ is output membership function for rule q. Now in the probabilistic fuzzy setting, the above expressions (20) and (21) need to be modified. Replacing the value of the output membership function from (9) into (20) and (21) we get

$$v_q = \int_y \sum_j \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} dy \tag{22}$$

$$b_q = \frac{\int_y y \sum_j \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} dy}{\int_y \sum_j \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} dy} \tag{23}$$

where $v_q$ is index of fuzziness and $b_q$ is the centroid.

The defuzzified output can be calculated in the ML model by applying the weighted average gravity method for the defuzzification. The defuzzified output value of $y^0$ is given by

$$y^0 = \frac{\int_y y\phi(y)dy}{\int_y \phi(y)dy} \tag{24}$$

where, $\phi(y)$ is the output membership function calculated using (19).

Also, the defuzzified output $y^0$ can be written as:

$$y^0 = \sum_{q=1}^{Q} \frac{\mu^q(x).v_q}{\sum_{q'=1}^{Q} \mu^{q'}(x).v_{q'}}.b_q \tag{25}$$

where $v_q$ and $b_q$ can be obtained using (22) and (23)

**Generalized Fuzzy Model**

The Generalized fuzzy model (GFM) by Azeem et al. in [12] generalizes both the ML model and the TS (Takagi- Sugeno) model. The output in GFM model has the properties of fuzziness (ML) around varying centroid (TS) of the consequent part of a rule. Let us consider a rule of the form

$$R^k\text{: if } x_k \text{ is } A_k \text{ then y is } B_k(f_k(x_k), v_k).$$

where $B_k$ is the output fuzzy set, $v_k$ is the index of fuzziness, $f_k$ is the output function.

Using (24), we can obtain the defuzzified output $y^0$ as

$$y^0 = \sum_{q=1}^{Q} \frac{\mu^q(x).v_q}{\sum_{q'=1}^{Q} \mu^{q'}(x).v_{q'}}.f^q(x) \tag{26}$$

where $f^q(x)$ is a varying singleton. It may be linear or non-linear. The linear form is: $f^q(x) = b_{q_0} + b_{q_0}x_1 + \ldots + b_{q_D}x_D$

Replacing the value of $b_q$ from (23) into (26) we get

$$y^0 = \sum_{q=1}^{Q} \frac{\mu^q(x).\int_y \sum_j \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} dy}{\sum_{q'=1}^{Q} \mu^{q'}(x).\int_y \sum_j \frac{P(O_j|x)\mu_j(y)}{\int_{-\infty}^{\infty} \mu_j(y)dy} dy}.f^q(x) \tag{27}$$

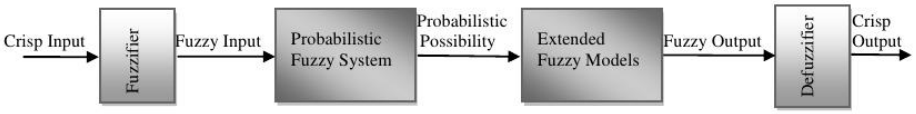



**Fig. 1.** Pictorial flow-chart of Probabilistic Fuzzy System

### 3.4 Computation of Probabilistic Possibility from Probabilistic Fuzzy Rules

The algorithm to compute the probabilistic possibility is presented, now . We refer to Fig. 1 to lay out the steps of the algorithm.

1. Pick appropriate linguistic terms to represent various inputs and output fuzzy sets.
2. Define the probabilistic fuzzy rules (PFRs).
3. Fuzzify crisp input by choosing appropriate support vector and grade (membership) vector for the chosen fuzzy sets.
4. Identify the PFRs that are applicable for the given test input x.
5. Evaluate the membership values for the applicable output fuzzy sets.
6. Calculate the conditional probability of each probabilistic output using (16).
7. Find the net output conditional possibility of the output using (18).
8. Compute the defuzzified output using (25) and (27). However this is an optional step and may be applied when all the values of parameters are available besides the possibility term (as computed in the above step).

## 4 Case-Studies

In this section, we take up two case-studies to illustrate the proposed concepts.

### 4.1 Probabilistic Fuzzy Air Conditioner

Let us contemplate the functioning of a fuzzy air conditioner example [13]. Design a fuzzy air conditioner control which takes in fuzzy input and has multiple probabilistic outputs. Let X be the input temperature (to be fuzzified) in Fahrenheit, and Y be the motor speed. Compute the net connditional output possibility when temperature is (1) $63^0F$ and (2) $68^0F$.

We perform the following steps to solve this case-study.

1. We pick the following inputs and output fuzzy sets, as shown in Fig. 2.
   - input fuzzy sets on X are: **Cold, Cool, Just Right, Warm,** and **Hot**
   - output fuzzy sets on Y are: **Stop, Slow, Medium, Fast,** and **Blast**
2. We define the following probabilistic fuzzy rules.
   If temperature is **cold** then motor speed is **stop** with probability 70%
   & motor speed is **slow** with probability 20%
   & motor speed is **medium** with probability 8%
   & motor speed is **fast** with probability 2%
   Similarly, other PFRs can also be constructed. The first column in Table 1 gives the antecedent value for each rule. The remaining columns give the values of the possible outputs for each rule.

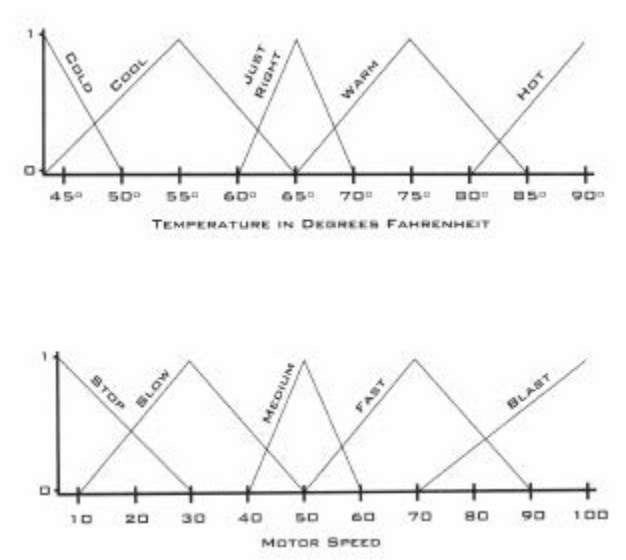



**Fig. 2.** Input fuzzy sets

**Table 1.** The Probabilistic Fuzzy Rule-set

| # | Temp(X) | $P_{Stop}$ | $P_{Slow}$ | $P_{Medium}$ | $P_{Fast}$ | $P_{Blast}$ |
|---|---|---|---|---|---|---|
| I | Cold | 0.7 | 0.2 | 0.08 | 0.02 | 0.0 |
| II | Cool | 0.1 | 0.7 | 0.1 | 0.08 | 0.02 |
| III | Just right | 0.05 | 0.1 | 0.7 | 0.1 | 0.05 |
| IV | Warm | 0.02 | 0.08 | 0.1 | 0.7 | 0.1 |
| V | Hot | 0.0 | 0.02 | 0.08 | 0.2 | 0.7 |

**Case: Input $63^0$F**

3. We compute the membership grades for the applicable input fuzzy sets after fuzzification of the input as shown in Fig. 3
   $\mu_0$(Just right): 0.8; $\mu_0$(Cool): 0.15
4. The applicable PFRs are nos. II and III.
5. The membership grades for the applicable output fuzzy sets are computed as shown in Fig. 3.
   $\mu_1$(Slow): 0.15; $\mu_1$(Medium): 0.80
6. The conditional probability of each output is computed referring to Table 1 and applying (16).
   $P(O_{Stop}|x) = (0.8 * 0.05) + (0.15 * 0.10) = 0.055$
   Similarly, $P(O_{Slow}|x) = 0.185$ $P(O_{Medium}|x) = 0.575$
   $P(O_{Fast}|x) = 0.092$ $P(O_{Blast}|x) = 0.043$
7. The net output conditional possibility is computed by applying (18)
   $\pi(y|x) = (0 + (0.185 * 0.15) + (0.575 * 0.8) + 0 + 0) = 0.48775$

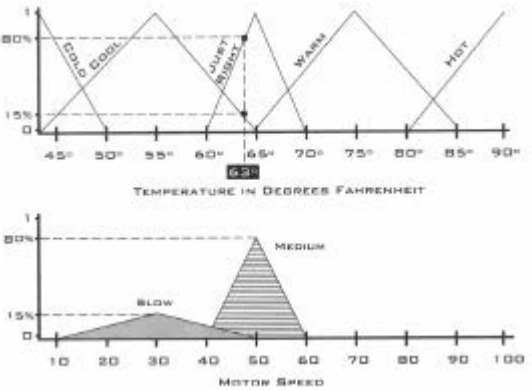



**Fig. 3.** Computation of membership grades when temperature is $63^0$ F

**Comparison of the Output with Basic Fuzzy Rules**

We now use the above algorithm to estimate the effect of the probabilistic output on the net output conditional possibility. The fuzzy rules of interest are as follows:

- If temperature is **cold**, motor speed is **stop**
- If temperature is **cool**, motor speed is **slow**
- If temperature is **just right**, motor speed is **medium**
- If temperature is **warm**, motor speed is **fast**
- If temperature is **hot**, motor speed is **blast**

The input and output fuzzy sets and their corresponding membership values are the same as above. The fuzzy rules that are fired are:

If temperature is **just right**, motor speed is **medium**
If temperature is **cool**, motor speed is **slow**

The output conditional probabilities are computed using (16) as

$\mathrm{P}(\mathrm{O}_{\mathrm{Stop}}|\mathrm{x}) = 0$ $\quad$ $\mathrm{P}(\mathrm{O}_{\mathrm{Slow}}|\mathrm{x}) = 0.15$
$\mathrm{P}(\mathrm{O}_{\mathrm{Medium}}|\mathrm{x}) = 0.8$ $\quad$ $\mathrm{P}(\mathrm{O}_{\mathrm{Fast}}|\mathrm{x}) = 0$
$\mathrm{P}(\mathrm{O}_{\mathrm{Blast}}|\mathrm{x}) = 0$

The net conditional possibility is found using (18) as

$$\pi(y|x) = (0 + (1 * 0.15) + (1 * 0.8) + 0 + 0) = 0.95$$

**Case: Input $68^0$F**

3. The fuzzy input and output membership values are:
   $\mu_0(\mathrm{Warm}) = 0.2$ $\quad$ $\mu_0(\mathrm{Justright}) = 0.55$
4. The applicable PFRs are nos. III and IV.
5. The membership grades for the applicable output fuzzy sets are:
   $\mu_1(\mathrm{Medium}) = 0.55$ $\quad$ $\mu_1(\mathrm{Fast}) = 0.2$

6. The conditional probability of each output is computed referring to Table 1 and applying (16)
   $\mathrm{P(O_{Stop}|x)} = 0.0315$ $\mathrm{P(O_{Slow}|x)} = 0.071$ $\mathrm{P(O_{Medium}|x)} = 0.525$
   $\mathrm{P(O_{Fast}|x)} = 0075$ $\mathrm{P(O_{Blast}|x)} = 0.0475$
7. The output conditional possibility is computed using (18)
   $\pi(y|x) = (0.55 * 0.525) + (0.2 * 0.075) = 0.303$

**Comparison of the Output with Basic Fuzzy Rules When Input is $68^0$F**

The conditional probabilities in the case of basic fuzzy rules can be computed as

$\mathrm{P(O_{Stop}|x)} = 0$ $\mathrm{P(O_{Slow}|x)} = 0$ $\mathrm{P(O_{Medium}|x)} = 0.55$
$\mathrm{P(O_{Fast}|x)} = 0.2$ $\mathrm{P(O_{Blast}|x)} = 0$

The net conditional possibility is found using (18) and is given here

$$\pi(y|x) = 0 + (1 * 0.55) + (1 * 0.2) + 0 + 0 = 0.75$$

It is pertinent to note that what we have here is the possibility in the probabilistic framework. So, in this example, the overall conditional possibility would converge to the sum of the individual possibilities, whereas in the case of probabilistic fuzzy rules, the conditional possibility is a factor of probabilities as well as possibilities.

### 4.2 Intelligent Oil Level Controller

Consider designing a fuzzy controller for the control of liquid level in a tank by varying its valve position [9]. The simple fuzzy controller, as shown in Fig. 4 employs $\Delta\mathrm{h}$ and dh/dt as inputs and $\mathrm{d}\alpha/\mathrm{dt}$ (rate of change of valve position $\alpha \in [0, 1]$) as the output, where h is the actual liquid level, $\mathrm{h_d}$ is the desired value of the level, and $\Delta\mathrm{h} = \mathrm{h_d} - \mathrm{h}$ is the error in the desired level.

Three Gaussian membership functions for three input fuzzy sets: **negative, zero, positive**, are applicable on the input variables $\Delta\mathrm{h}$ and dh/dt. The output fuzzy sets: **close-fast, close-slow, no-change, open-slow, open-fast** have triangular membership functions. The following fuzzy rules are selected using a human experts knowledge.

| | |
|---|---|
| I | If $\Delta\mathrm{h}$ is zero then $\mathrm{d}\alpha/\mathrm{dt}$ is **no-change** |
| II | If $\Delta\mathrm{h}$ is positive then $\mathrm{d}\alpha/\mathrm{dt}$ is **open-fast** |
| III | If $\Delta\mathrm{h}$ is negative then $\mathrm{d}\alpha/\mathrm{dt}$ is **close-fast** |
| IV | If $\Delta\mathrm{h}$ is zero and dh/dt is positive then $\mathrm{d}\alpha/\mathrm{dt}$ is **close-slow** |
| V | If $\Delta\mathrm{h}$ is zero and dh/dt is negative then $\mathrm{d}\alpha/\mathrm{dt}$ is **open-slow** |

In order to model the existing vagueness and uncertainty in our opinions, we may substitute each conventional rule with a probabilistic fuzzy rule with the output probability vector P defined such that the only output sets of the conventional fuzzy rules are the most probable from the probabilistic fuzzy rules. Also the neighbouring fuzzy sets in the PFR have smaller probabilities and the other fuzzy sets have zero probabilities. For example rule I in the above rule set may be modified as follows:

**Table 2.** The Probabilistic Fuzzy Rule-set for the Liquid Level Fuzzy Controller

| # | $Qty_1$ | $Val_1$ | $Qty_2$ | $Val_2$ | $P_{close-fast}$ | $P_{close-slow}$ | $P_{no-change}$ | $P_{open-slow}$ | $P_{open-fast}$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $\Delta h$ | 0 | | | 0 | 0.1 | 0.8 | 0.1 | 0 |
| 2 | $\Delta h$ | + | | | 0 | 0 | 0 | 0.2 | 0.8 |
| 3 | $\Delta h$ | - | | | 0.8 | 0.2 | 0 | 0 | 0 |
| 4 | $\Delta h$ | 0 | dh/dt | + | 0.1 | 0.8 | 0.1 | 0 | 0 |
| 5 | $\Delta h$ | 0 | dh/dt | - | 0 | 0 | 0.1 | 0.8 | 0.1 |

I. If $\Delta h$ is zero then $d\alpha/dt$ is no-change with probability 80%
& $d\alpha/dt$ is close-slow with probability 10%
& $d\alpha/dt$ is open-slow with probability 10%

The consequent part of the PFR can be thus expressed in a compact form using the output probabilities vector P. The sample probabilistic fuzzy rule set is given in Table 2. Let Input: $\Delta h = 0$. The PFRs for the given input are as follows:

I If $\Delta h$ is **zero** then $d\alpha/dt$ is **no-change** with probability 80%
& $d\alpha/dt$ is **close-slow** with probability 10%
& $d\alpha/dt$ is **open-slow** with probability 10%

IV If $\Delta h$ is **zero** and dh/dt is positive then $d\alpha/dt$ is **no-change** with probability 10%
& $d\alpha/dt$ is **close-slow** with probability 80%
& $d\alpha/dt$ is **close-fast** with probability 10%

V If $\Delta h$ is **zero** and dh/dt is negative then $d\alpha/dt$ is **no-change** with probability 10%
& $d\alpha/dt$ is **open-slow** with probability 80%
& $d\alpha/dt$ is **open-fast** with probability 10%

The membership values, $\mu_{Zero}(x)$, $\mu_{Positive}(x)$ and $\mu_{Negative}(x)$ for the given input are given as follows:

$\mu_{Zero}(\Delta h) : 1 \qquad \mu_{Positive}(\frac{dh}{dt}) : 1 \qquad \mu_{Negative}(\frac{dh}{dt}) : 1$

The membership grades for the output fuzzy sets are given as follows:

$\mu_{No-change}(\frac{d\alpha}{dt}) : 1 \qquad \mu_{Slow}(\frac{d\alpha}{dt}) : 0.15 \qquad \mu_{Fast}(\frac{d\alpha}{dt}) : 0.15$

The conditional probability is calculated using (16) for each probabilistic output in each fuzzy rule that is applicable, given the input value.

$P(O_{No-change}|x) = [(1 * 0.8) + (1 * 0.1) + (0 * 0.1)]/2 = 0.45$

Note:- The probability values are normalized by taking the number of the input fuzzy sets as denominator.

Similarly, $P(O_{Close-slow}|x) = 0.45 \qquad P(O_{Close-fast}|x) = 0.1$
$P(O_{Open-slow}|x) = 0.1 \qquad P(O_{Open-fast}|x) = 0$

The net conditional possibility for the output is computed using (18) as

$$\pi(y|x) = (0.45 * 1) + (0.45 * 0.15) + (0.1 * 0.15) + (0.1 * 0.15) + (0 * 0.15) = 0.5475$$

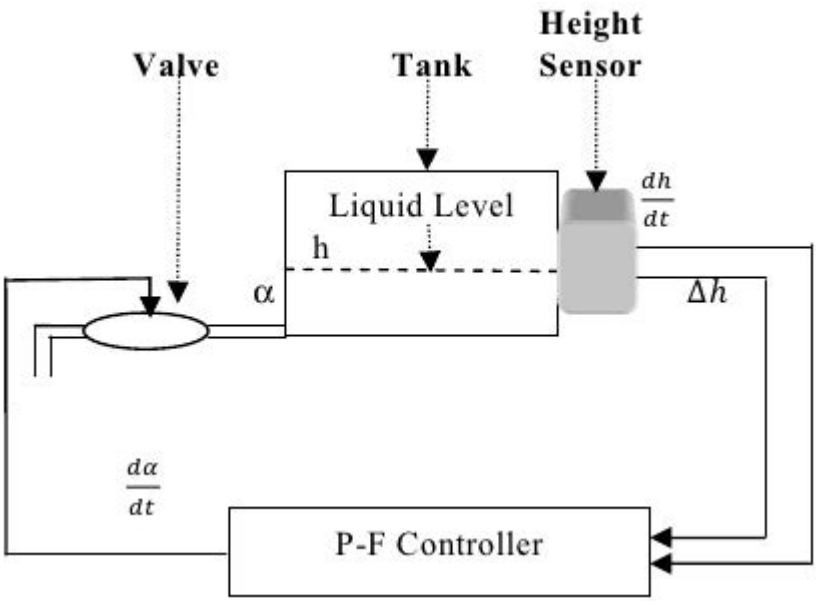



**Fig. 4.** Block Diagram for Intelligent Oil Level Controller

Thus having obtained the value of net membership, the same can be substituted in the ML and GFM models to obtain (vq, bq). It can also be noted that for the basic fuzzy rules the net conditional possibility for a given input is the sum of the memberships of the various output fuzzy sets that are applicable.

## 5 Conclusions

A probabilistic fuzzy framework, based upon probabilistic fuzzy rules, has been designed for modelling real world uncertainty. The probabilistic possibility is computed which is used to obtain probabilistic output by extending the existing fuzzy models. It is also shown that a probabilistic fuzzy framework is more flexible and convenient than the conventional methodology in representing the real world uncertainty. Its ability to represent fuzzy nature of situations along with corresponding probabilistic information brings it much closer to real-world. Two examples dealing with the practical applications of an air-conditioner and a liquid level controller are taken up to demonstrate the probabilistic fuzzy system. It is noticed that in the case of probabilistic fuzzy systems, the probabilities associated with various outputs affects the net output probabilistic possibility for an input. All the the applicable output fuzzy sets contribute towards the computation of the output probabilistic possibility.

The results are compared with those obtained throuh conventional fuzzy systems. A conventional fuzzy system is a special case of probabilistic fuzzy system in which there is only one output for a fuzzy rule that translates into 100 % probability for that particular output. The methodology proposed for calculating output probabilistic possibility

for PFRs fits well with basic fuzzy rules and leads to the intuitively acceptable result. The proposed work provides functionality to process the probabilistic fuzzy rules that are better equipped to represent the real-world situations.

Another feature of probabilistic fuzzy rules is the enhanced adaptability in view of the outputs with varying probabilities. This is borne out of the fact that the outputs in the fuzzy rules are context- dependent hence vary accordingly. The proposed approach to calculate the possibility from probability can be tailored to a specific application depending upon the output membership functions and their probabilities. This can also be extended to represent probabilistic rough fuzzy sets and other types of fuzzy sets so as to increase its utility in capturing the higher forms of uncertainty. The probabilistic information adds enormous potential to the existing possibility based fuzzy models in decision making in the real-world situations, as shown in this study. The proposed framework addresses the uncertainty arising from fuzziness and vagueness in the wake of their random occurrences.

## References


1. Zadeh, L.A.: Fuzzy Sets as a Basis for a Theory of Possibility. Fuzzy Sets and Systems 1, 3–28 (1978)
2. Dubois, D., Prade, H.: When upper probabilities are possibility measures. Fuzzy Sets and Systems 49, 65–74 (1992)
3. Dubois, D., Prade, H., Sandri, S.: On possibility/probability transformations. In: Lowen, R., Roubens, M. (eds.) Fuzzy Logic, pp. 103–112 (1993)
4. Roisenberg, M., Schoeninger, C., Silva, R.R.: A hybrid fuzzy-probabilistic system for risk analysis in petroleum exploration prospects. Expert Systems with Applications 36, 6282–6294, 103–112 (2009)
5. De Cooman, G., Aeyels, D.: Supremum-preserving upper probabilities. Inform. Sci. 118, 173–212 (1999)
6. Walley, P., de Cooman, G.: A behavioural model for linguistic uncertainty. Inform. Sci. 134, 1–37 (1999a)
7. Dubois, D., Prade, H.: On several representations of an uncertain body of evidence. In: Gupta, M.M., Sanchez, E. (eds.) Fuzzy Information and Decision Processes, pp. 167–181. North-Holland (1982)
8. Dubois, D.: Possibility theory and statistical reasoning. Computational Statistics & Data Analysis 51(1), 47–69 (2006)
9. Meghdadi, A.H., Akbarzadeh-T, M.-R.: Probabilistic fuzzy logic and probabilistic fuzzy systems. In: The 10th IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1127–1130 (2001)
10. Van den Berg, J., Van den Bergh, W.M., Kaymak, U.: Probabilistic and statistical fuzzy set foundations of competitive exception learning. In: The 10th IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1035–1038 (2001)
11. Van den Bergh, W.M., Kaymak, U., Van den Berg, J.: On the data-driven design of Takagi-Sugeno probabilistic furzy systems. In: Proceedings of the EUNlTE Conference, Portugal (2002)
12. Azeem, M.F., Hanmandlu, M., Ahmad, N.: Generalization of adaptive neuro-fuzzy inference systems. IEEE Transactions on Neural Networks 11(6), 1332–1346 (2000)
13. Kosko, B.: Fuzzy Thinking: The New Science of Fuzzy Logic. Hyperion (1993)
14. Klir, G.J.: Fuzzy Sets: An Overview of Fundamentals, Applications and Personal Views. Beijing Normal University Press, Beijing (2000)

# A Root-Cause-Analysis Based Method for Fault Diagnosis of Power System Digital Substations

Piao Peng[1], Zhiwei Liao[1], Fushuan Wen[2] and Jiansheng Huang[3,*]

[1] School of Electrical Engineering, South China University of Technology, Guangzhou, China
ggpengpiao@163.com, epliao@scut.edu.cn
[2] School of Electrical Engineering, Zhejiang University, Hangzhou, China
fushuan.wen@gmail.com
[3] School of Computing and Mathematics, University of Western Sydney, Sydney, Australia
j.huang@uws.edu.au

**Abstract.** Fault Diagnosis of power systems has attracted great attention in recent years. In the paper, the authors present a Cause-Effect fault diagnosis model, which takes into account the structure and technical features of a digital substation and performs root-cause-analysis so as to identify the exact reason of a power system fault occurred in the monitored district. The Dempster/Shafer evidence theory has been employed to integrate different types of fault information in the diagnosis model aiming at a hierarchical, systematic and comprehensive diagnosis based on the logic relationship between the parent fault node and the child nodes like transformers, circuit-breakers, and transmission lines, and between the root and child causes. An actual fault scenario is investigated in the case study to demonstrate the capability of the developed model in diagnosing malfunctions of protective relays and/or circuit breakers, miss or false alarms, and other faults often encountered at modern digital substations of a power system.

**Keywords:** Digital substation, Fault diagnosis, Root cause analysis, Dempster/Shafer theory, Fishbone diagram.

## 1 Introduction

Fault diagnosis and accident management in substations have become a major challenge in reinforcing power systems' safety and reliability. Many integrated substation diagnosis models and methods have been proposed to address this challenge by using information obtained from protective relays and circuit breakers and employing technologies such as expert systems [1-3], artificial neural networks [4-5], Petri networks [6-7], agent technology [8], and rough sets [9-10]. In addition, substation diagnosis models and methods may rely on a single transmission or transformation equipment such as done by the transformer diagnosis model based on three chromatographic level correlation analyses [11], and wavelet theory based transmission line fault diagnosis model using fault recorders [12]. It is observed that current substation fault diagnosis models only take into account information of protective relays and circuit

* Corresponding author.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 153–165.
DOI: 10.1007/978-3-642-35638-4_11 

breakers, or fault features of a single device. In other words, the existing models and methods, due to employing only local information, are inadequate to diagnose complex faults with uncertainties, including multiple consecutive failures, malfunctions of protective relays and/or circuit breakers, missing or false alarms, and sensor errors, to name a few [13].

With the advent of new technologies and tools such as intelligent primary/secondary equipment and IEC61850 communication standard, applications of digital technologies have become the trend in substation automation, calling for novel fault diagnosis methods and models with information sharing and interoperability of monitoring and controlling devices.

In the paper, the authors propose a Root Cause Analysis (RCA) based Cause-Effect (fishbone diagram) fault diagnosis model for digital substations of power systems. The fusion rule of the well-developed Dempster/Shafer (D-S) evidence theory is used to integrate different types of fault information obtained through monitoring status of substations, protective relays and circuit breakers. Based on the logic relationship between the parent and the transformer/circuit breaker/transmission line child nodes, and between the root and the child causes, in the diagnosis model, an hierarchical, systematic, and comprehensive diagnosis can then be performed. A software package has been developed to implement the proposed fault diagnosis model, which has been deployed in Xingguo Substation, the first digital substation in the State Grid of Jiangxi Province, China.

## 2 Basic Principles of RCA

The RCA, originally applied in organization management [14], is a hierarchical and systematic approach to identify and analyze the root cause of an incident and develop the countermeasures accordingly. A large size substation comprises many components with interactions among them. Through analyzing these interactions, a novel fault diagnosis model for substation transmission and transformation systems can be built up according to the structure and data/information flows of a digital substation. Based on the theory of RCA, the fault diagnosis model is formulated to explain the linkage chain of accident causes in order to identify what really happened and what the applicable countermeasures could be.

The research tools of RCA include Cause&Effect/Fishbone Diagram, Brain Storm and WHY-WHY Diagram. There are three types of Fishbone Diagrams: arrangement-based, cause-based and solution-based. As shown in Fig. 1, the cause-based fishbone diagram is adopted to explain the philosophy of applying the RCA in fault diagnosis of digital substations. The following explains each component of the diagram:

- $F$ : a problem node to be solved as a specified fault in a substation .
- $c_i$ : a child cause of $F$ and a basic reason of a specified fault. $p(c_i)$ denotes the fault probability caused by $c_i$ . $S(F) = \{c_1, c_2, \cdots, c_i\}$ is the set of child causes which could trig $F$ .
- $r_j$ : a root cause of $F$ and a fundamental reason of a specified fault in the power system. $p(r_j \mid c_i)$ denotes the conditional fault probability caused by $r_j$ with given $c_i$ and $G(c_i)=\{r_1,r_2,\ldots,r_j|c_i\}$ is the root cause set.

- $FN$ : the only parent node for the diagnosis system. $FN=(D,M,O)$ , composed of three elements $D$ , $M$ and $O$ , denotes the basic diagnosis functions. $D$ represents the composition of the access modes to obtain the required information from the source, $D \subseteq D_e=\{d_1,d_2,\cdots,d_n\}$ , and $D_e$ is the collection of all the *n* available modes. M= $\{met_1, met_2,\ldots, met_p\}$, denotes the $p$ fault diagnosis methods applicable at the node. $O=\{[c_i, p(c_i)]\,|\,(i=1,2,\cdots,q)\}$ is the diagnosis output, where $c_i \in O$ , $q$ is the number of the reasons $\{ c_i \}$, and $p(c_i)$ denotes the fault probability caused by $c_i$ .

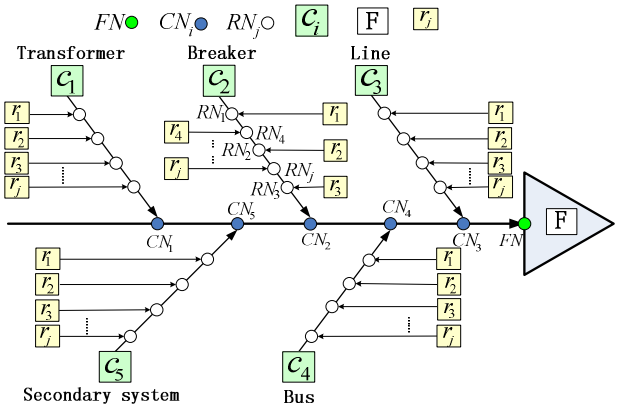



**Fig. 1.** Framework of RCA-based fault diagnosis system for digital substations

- $CN_i$ and $RN_j$ : the child nodes and the root nodes. Like $FN$ , they are constituted by the three elements $D$ , $M$ , $O$ . Furthermore, they can give a more detailed diagnosis based on that of $FN$ . Thereinto, $S(CN)=\{CN_1, CN_2, \ldots, CN_i\} \subseteq FN$, with $S(CN)$ denoting the set of all the child nodes belonging to $FN$ ; $S(RN|CN_k)=\{RN_1, RN_2, \ldots, RN_i\} \subseteq CN_k$ is the set of root node $RN$ belonging to the child node $CN_k$ .
- In light of above, it can be seen that all nodes, including $FN$ , $CN_i$ and $RN_j$ , are independent in obtaining the information needed by the diagnosis, selecting the appropriate diagnosis methods, and analyzing fault reasons of each node.

## 3 Fault Diagnosis of Digital Substation

A root cause analysis based fault diagnosis system for digital substations is as shown in Fig. 1. $S(CN)=\{CN_1,CN_2,CN_3,CN_4,CN_5\}$ and $S(F)=\{c_1,c_2,c_3,c_4,c_5\}$ denote the child nodes and the child causes of transformer, circuit breaker, line, bus and secondary system (DC power supply, network communications and security devices).

### 3.1 The Mode to Obtain Information

The Substation Configuration Description Language (SCL) is used to describe IEC61850 standard based IED configuration and related parameters, communication

system configuration, substation system structure, and the relationship among them for information exchanging. Logical node LN is the basic function unit of a digital substation to obtain the needed information. Part of the logic nodes required in the designed fault diagnosis is listed in Table 1.

**Table 1.** Main logic nodes in SCL

| | Logical node<br>Explain |
|---|---|
| 1. | Pxyz (Protective relay)<br>Protection operation |
| 2. | XCBR (Circuit breaker)<br>Switch position |
| 3. | RREC (Reclosing)<br>Reclosing operation |
| 4. | XSWI (Knife switch)<br>Knife position |
| 5. | SMIL (Online monitoring information of transformer oil chromatography)<br>Monitoring value |
| 6. | SCBR (Online monitoring information of circuit breaker )<br>Monitoring value |

For a comprehensive analysis and diagnosis of an accident, the required diagnosis information should also include various electrical and chemical test results of the equipment. The diagnosis information is divided into three types as following:

- The location variant information, i.e., the remote information with time stamps.
- The section information, including the information of remote communication and remote measurement at a certain time point.
- The data files, including various electrical and chemical test results of equipment, chemical experiment results, overhaul history, waveform files of breaker and recorded faults via on-line monitoring.

### 3.2 The Parent and Child Nodes

The information access mode $D = \{d_1\}$ of a parent node $FN = (D, M, O)$ is a passive one in obtaining location variant information of Pxyz(Protective relay), XCBR(Circuit breaker), RREC(Reclosing) and secondary equipment. The diagnosis method $M = \{m_1\}$ based on the optimization algorithm developed in [14] is to diagnose substation faults with information obtained from protective relays, circuit breakers and secondary equipment. The output of the diagnosis is $O = \{[c_1, p(c_1)], [c_2, p(c_2)], [c_3, p(c_3)], [c_4, p(c_4)], [c_5, p(c_5)]\}$ where $c_1, c_2, c_3, c_4, c_5$ respectively denote transformer faults, malfunction of protective relays and/or circuit breakers, line faults, bus faults, and malfunctions of secondary equipment (Fig. 2). The child nodes such as $CN_1$, $CN_2$ and $CN_3$ are defined below:

- The child node of transformer $CN_1$

The information access mode $D_1 = \{d_2\}$ of child node $CN_1 = (D_1, M_1, O_1)$ is an active mode in obtaining online monitoring information of the transformer oil

chromatography. $M = \{m_1\}$ is the method to analyze the gas in the oil to diagnose transformer faults using the improved three-ratio method. $O_1 = \{[r_j, p(r_j \mid c_1)] \mid j = 1, 2, \cdots, 9\}$ is the output of the diagnosis, where $r_1$ is partial discharge, $r_2$ is type-1 low-temperature overheating(below 150°C), $r_3$ is type-2 low-temperature overheating (150°C-300°C), $r_4$ is medium-temperature overheating, $r_5$ is high-temperature overheating, $r_6$ is low-energy discharge, $r_7$ is low-energy discharge and overheating, $r_8$ is arc discharge, and $r_9$ is arc discharge and overheating. $p(r_j \mid c_1)$ is the fault probability caused by $r_j$ with given $c_1$.

- The child node of circuit breaker $CN_2$

1) $D_2 = \{d_1, d_2, d_3\}$ is the information access mode of child node $CN_2 = (D_2, M_2, O_2)$, and $d_1$ is a passive mode in obtaining location variant information of XCBR, $d_2$ is an active mode in obtaining online monitoring information of SCBR, $d_3$ is an active FTP mode in obtaining online monitoring waveform files of circuit breakers. Based on the Dempster's Fusion Rule and the expert knowledge-base, $M = \{m_2\}$is the method to establish the set of state sign with online monitoring information, including switching coil current, switch waveform file, storage time of energy-storage motor, and current curves. The method diagnoses faults of circuit breakers according to coil switching current RMS and elapsed time, energy-storage motor storage time, total distance of a circuit breaker's operation, instantaneous and average switching speed of a circuit breaker.

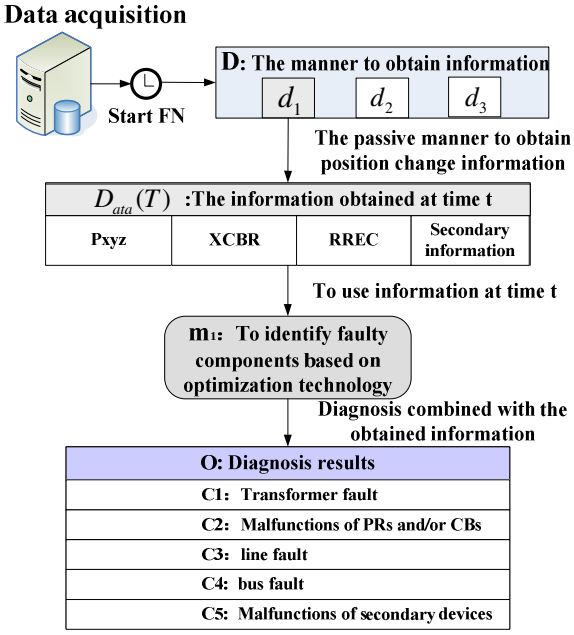



**Fig. 2.** The functional diagram of FN

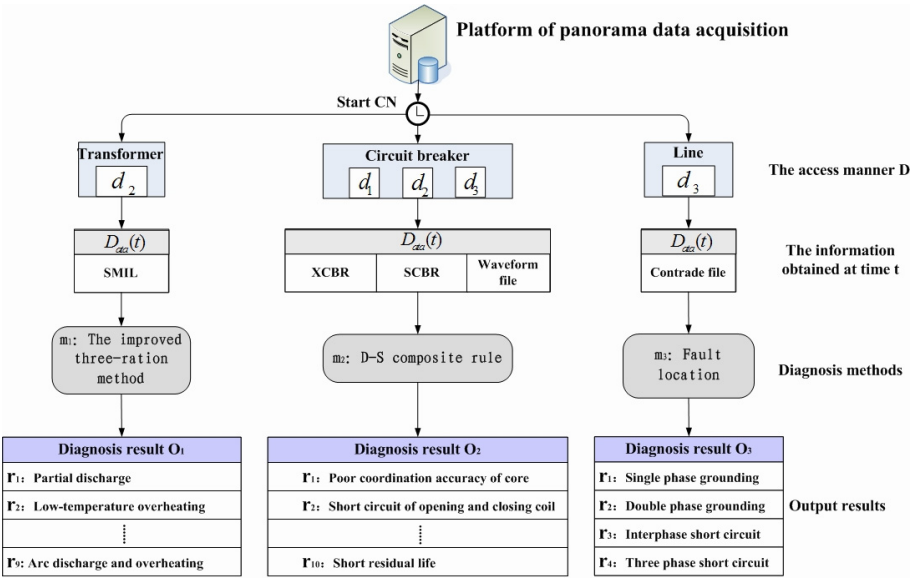



**Fig. 3.** The functional diagram of CN

$O_2 = \{[r_j, p(r_j \mid c_2)] \mid j = 1,2,\cdots,10\}$ is the output of the diagnosis, $r_1$: mismatch of the switching coil core and over resistance of switch operation, $r_2$: short circuit of the switching coil, $r_3$: burn or break of the switching coil, $r_4$: deformation or displacement of latch and valve connected to the core mandrel, $r_5$: poor contact and operation of auxiliary switch and closing contactor, $r_6$: fault of DC power or system auxiliary power, $r_7$: fault of operating mechanism, $r_8$: fault of energy-storage motor, $r_9$: mechanical failure, such as deformation and displacement of linkage unit, and latch failure, and $r_{10}$: short residual life. $p(r_j \mid c_2)$ denotes the fault probability caused by $r_j$ with given $c_2$.

- The child node of line *CN3*

$D_3 = \{d_3\}$ is the information access mode of child node $CN_3 = (D_3, M_3, O_3)$, $d_3$ is the active FTP mode to obtain recorded line fault files (Contrade). $M = \{m_3\}$ is the method that utilizes the sudden-change of the phase current difference to select the phase and then locate the fault by estimating the distance with the sampled data from the recorded fault curve. $O_3 = \{[r_j, p(r_j \mid c_2)] \mid j = 1,2,3,4\}$ is the output of the diagnosis, with $r_1$ as single phase grounding fault, $r_2$ as double phase grounding fault, $r_3$ as inter-phase short circuit fault, $r_4$ as three phase short circuit fault. $p(r_j \mid c_3)$ denotes the fault probability caused by $r_j$ with given $c_3$. In addition, the identified fault location is included in the diagnosis output (Fig. 3).

# 4 The Fault Diagnosis Flowchart Based on RCA

As illustrated by the flowchart in Fig. 4, the RCA based fault diagnosis includes two cases.

## 4.1 Without Operation of Protective Relaying, Circuit Breaking and Reclosing

This case is mainly for monitoring and evaluating the status of transmission and transformation equipment. Each child node ( $D$ , $M$ and $O$ ) is started periodically with a timer interval $t_{\text{int}erval}$ . According to the output of child nodes $CN_1$, $CN_2$, $CN_3$ and $CN_5$, the states of transmission and transformation equipment of the substation is evaluated, with the evaluation results $O_1$, $O_2$, $O_3$, $O_5$ and $R$ as given in Eqn. (1).

$$R = O_1 \cup O_2 \cup O_3 \cup O_5 = \begin{Bmatrix} [(r_j \mid c_1), p(r_j \mid c_1)] \\ [(r_j \mid c_2), p(r_j \mid c_2)] \\ [(r_j \mid c_3), p(r_j \mid c_3)] \\ [(r_j \mid c_5), p(r_j \mid c_5)] \end{Bmatrix} \tag{1}$$

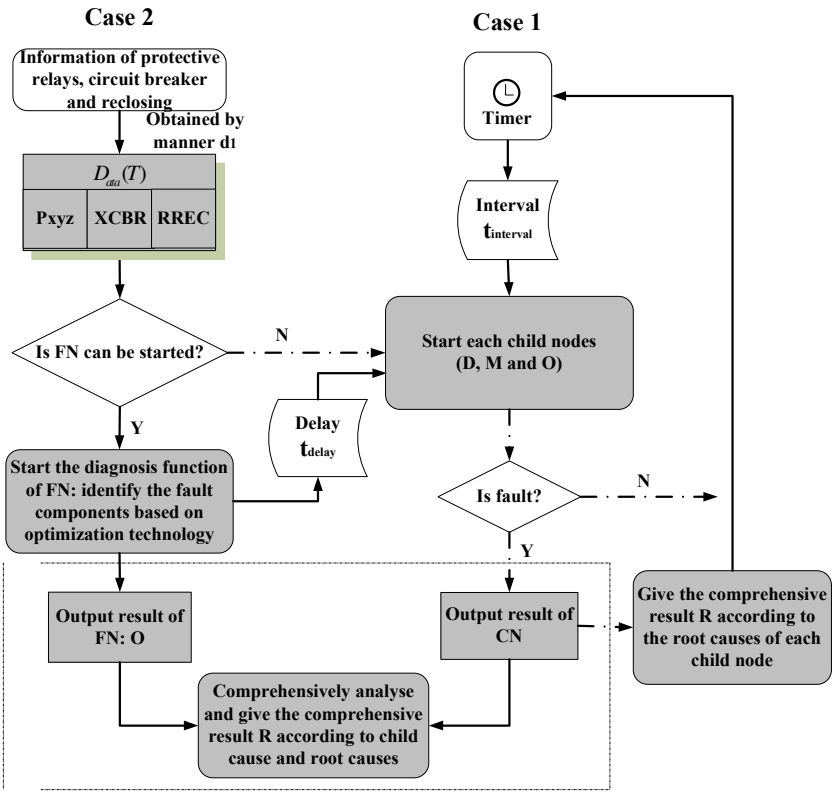



**Fig. 4.** The RCA based fault diagnosis

### 4.2 With Operation of Protective Relaying, Circuit Breaking and Reclosing

The major analysis and diagnosis procedure of this case is as following:

- Once the protective relays, circuit breakers and re-closers operate, the diagnosis *M* of *FN* is started, to obtain the location variant information of Pxyz, XCBR, RREC and secondary equipment by mode $d_1$. In the diagnosis, the optimization technology is employed to identify the faulty components and provide the child cause set of fault as shown in Eqn. (2).

$$\begin{aligned} S(F) = O = \{&[c_1, p(c_1)], [c_2, p(c_2)], \\ &[c_3, p(c_3)], [c_4, p(c_4)], [c_5, p(c_5)]\} \end{aligned} \tag{2}$$

- Due to the lag in acquiring various waveform files compared with obtaining the location variant information, a delay of $t_{delay}$ is introduced to start each child node.

- To avoid conflicts between the parent node and child node due to potential errors existing in information source, the set of all possible faults is used as the basis of the diagnosis, and each symptom of faults is used as the evidence in conducting the comprehensive analysis to the output of *FN*, $O_1$, $O_2$, $O_3$, $O_4$, and $O_5$. The frame of discernment is the basic concept of D-S evidence theory. For a judgment problem, all possible results that can be recognized are expressed by Θ, a non-empty set known as the frame of discretion. The frame consists of a number of mutually exclusive and exhaustive elements. $\Theta = \{q_1, q_2, q_3, q_4, q_5\}$, where $q_1$ is a transformer fault, $q_2$ is malfunction of protective relays and/or circuit breakers, $q_3$ is a line fault, $q_4$ is a bus fault and $q_5$ is malfunction of secondary equipment. If $m(q_i)$, the assigned value to function *m* by proposition $q_i$, meets the following conditions:

$$m(\Phi) = 0 \tag{3}$$

$$\forall q_i \in \Theta,\ m(q_i) \geq 0, \text{且} \sum_{q_i \in \Theta} m(q_i) = 1 \tag{4}$$

$m(q_i)$ is known as the basic probability assignment function (BPAF) of $q_i$, which reflects the belief to the accuracy of $q_i$, i.e., the direct support to $q_i$ but no support to any subset of $q_i$. Furthermore, $m(q_i)$ is defined as the focus element of evidence if $q_i$ is a subset of Θ and $m(q_i)$>0. Φ represents an empty set in Eqn. (3). Here the diagnosis result of *FN* is taken as Evidence-1 corresponding to the BPAF $m_1(q_k)$, and the diagnosis result of *CN* is taken as Evidence-2 corresponding to the BPAF $m_2(q_l)$. $m_1(q_k)$ and $m_2(q_l)$ are supposed to be the two BPAF of independent evidence in the same frame of discernment Θ. While $m_1$ is the BPAF of Evidence-1 with $m_1(q_k) = p(c_i)$, $m_2$ is the BPAF of Evidence-2 with $m_2(q_l) = p(r_j \mid c_i)$.

The D-S Fusion Rule is to reflect the joint effect of the evidences in the same frame of discernment through calculating a single BPAF with the BPAFs of different evidences. By applying the rule, the joint effect of Evidence-1 and Evidence-2 is evaluated in Eqn. (5).

$$m(q)=\frac{\sum_{q_k\cap q_l=q} m_1(q_k)m_2(q_l)}{\sum_{q_k\cap q_l\neq\Phi} m_1(q_k)m_2(q_l)}=m_1(q_k)\oplus m_2(q_l) \tag{5}$$

where $m(q)$ is the orthogonal sum of $m_1(q_k)$ and $m_2(q_l)$, denoted by $m=m_1\oplus m_2$.

$$\sum_{q_k\cap q_l\neq\Phi} m_1(q_k)m_2(q_l)=1-\sum_{q_k\cap q_l=\Phi} m_1(q_k)m_2(q_l)=1-k \tag{6}$$

where $k=\sum_{q_k\cap q_l=\Phi} m_1(q_k)m_2(q_l)$ expressing the conflict degree resulted in the fusion course of the evidences, and $0\leq k\leq 1$. In general, the larger the $k$, the more intense conflicts are among the evidences.

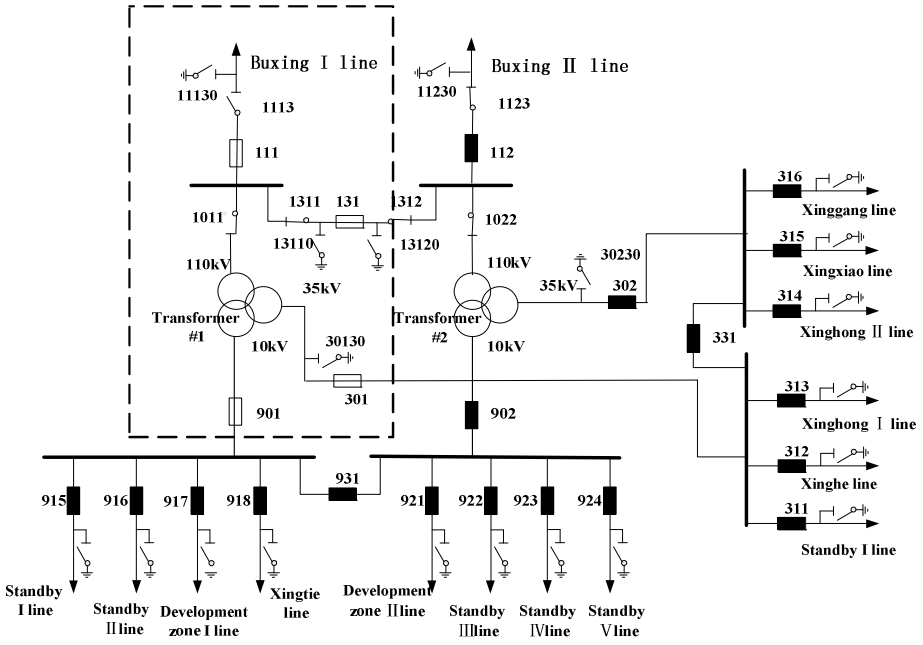



**Fig. 5.** The main connection scheme of the 110 kV Xingguo digital substation

Case Study

The developed software package has been applied in the 110 kV Xingguo Substation, the first digital substation in Jiangxi Province, China. The power outage region due to a fault is circled by the dotted lines as shown in Fig. 5. The fault diagnosis is carried out as follows:

- The location variant information is obtained by mode $d_1$ (Table 2). $D_{ata}(T)$ denotes the information obtained from 2009-12-20 15:20:12 50ms to 2009-12-20 15:20:13 383ms.

- The diagnosis function $M$ of $FN$ is started to identify the faulty components and then provide the child cause set of the fault $S(F) = O = \{[c_1, 0.4], [c_2, 0], [c_3, 0.6], [c_4, 0], [c_5, 0]\}$ which reveals that the probability is 0.4 for a transformer fault, and is 0.6 for a line fault.
- The online monitoring information of transformer oil chromatography is obtained by mode $d_2$ (Table 3). The coil current and switch signal waveform of the circuit breaker numbered 111 is obtained by mode $d_3$ (Fig. 6). The recorded fault curve of Line Buxing-I is obtained by mode $d_3$ (Fig. 7). A 10s delay $t_{delay}$ is set to start the child nodes $CN_1$, $CN_2$, $CN_3$ and $CN_5$. Finally the root cause has been identified as a single phase grounding fault in Line Buxing-I $O_3 = \{[r_1, 1]\}$.
- According to the D-S Fusion Rule, the diagnosis result is obtained as given in Table 4:

**Table 2.** Location variant information obtained by mode $d_1$

| Time | Alarm ID | Alarm value | Alarm description |
|---|---|---|---|
| 2009-12-20 15:20:12 50ms | PCOS_PZB1H/Q0PTOC3$ST$Op $general | 1 | Operation of overcurrent, segment-2, 1st time, limit of high reserve for transformer 1# |
| 2009-12-20 15:20:13 150ms | PCOS_P110LINE1/Q0XCBR1$S T$Pos$stVal | 1 | Operation of circuit breaker numbered 111 |
| 2009-12-20 15:20:13 260ms | PCOS_PZB1L/Q0XCBR1$ST$Po s$stVal | 1 | Operation of circuit breaker numbered 901 |
| 2009-12-20 15:20:13 327ms | PCOS_PZB1M/Q0XCBR1$ST$P os$stVal | 1 | Operation of circuit breaker numbered 301 |
| 2009-12-20 15:20:13 383ms | PCOS_ P110LINE3/Q0XCBR1$ST$Pos$s tVal | 1 | Operation of circuit breaker numbered 131 |

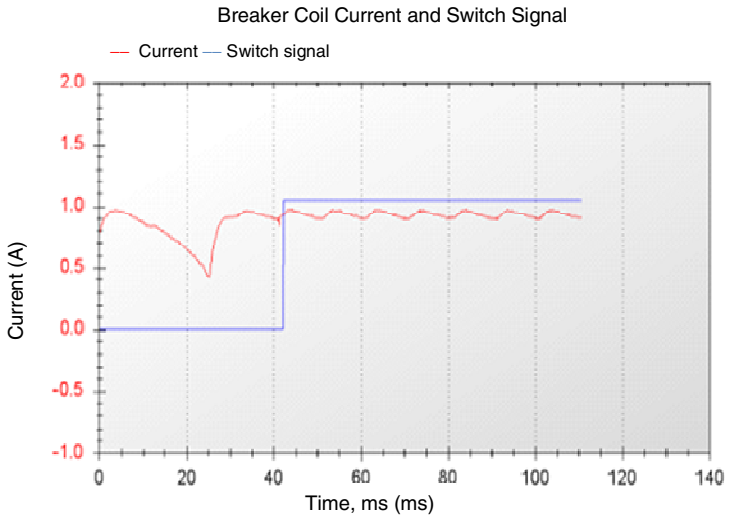



**Fig. 6.** Coil current and switch signal waveform of circuit breaker numbered 111 obtained by mode $d_3$

**Table 3.** Online monitoring information of transformer oil chromatography obtained by mode $d_2$

| Time | Alarm ID | Alarm value | Alarm description |
|---|---|---|---|
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$H2$mag$f | 35 | Hydrogen measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$CH4$mag$f | 12 | Methane measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H4$mag$f | 15 | Ethylene measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H2$mag$f | 0 | Acetylene measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H6$mag$f | 8 | Ethane measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$CO$mag$f | 406 | Carbon monoxide measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$CO2$mag$f | 120 | Carbon dioxide measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$THC$mag$f | 35 | THC measurement of transformer 1#(uL/L) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$H2AbsRte$mag$f | 1 | Absolute gas production rate of hydrogen of transformer 1#(uL/d) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H2$mag$f | 0 | Absolute gas production rate of methane of transformer 1#(uL/d) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H2$mag$f | 0.5 | Absolute gas production rate of ethene of transformer 1#(uL/d) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H6$mag$f | 0 | Absolute gas production rate of acetylene of transformer 1# |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H6$mag$f | 0 | Absolute gas production rate of ethane of transformer 1#(uL/d) |
| 2009-12-20 15:20:23 | PCOS_YSP1/Q0SIML0$MX$C2H6$mag$f | 0.5 | Absolute gas production rate of THC of transformer 1#(uL/d) |

**Table 4.** The composite results

| BPAF of nodes | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ |
|---|---|---|---|---|---|
| $m_1$ | 0.4 | 0 | 0.6 | 0 | 0 |
| $m_2$ | 0 | 0 | 1 | 0 | 0 |
| $m = m_1 \oplus m_2$ | 0 | 0 | 1 | 0 | 0 |

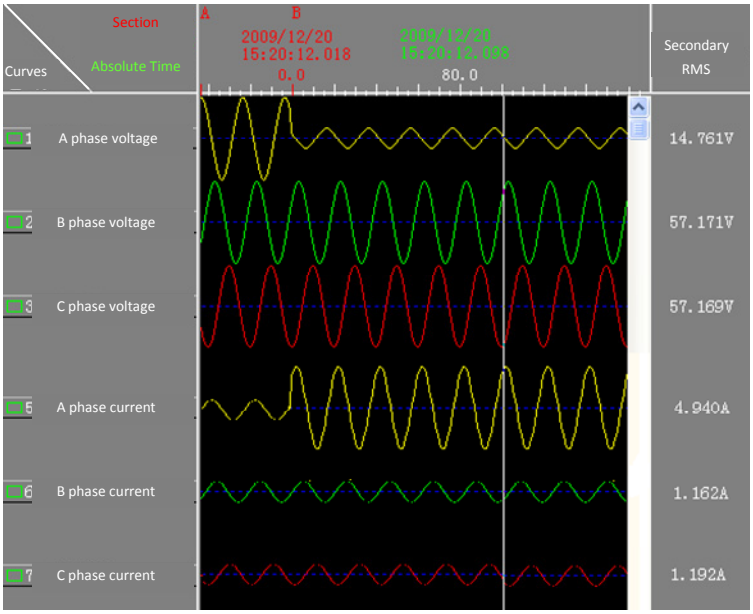



**Fig. 7.** The fault recording curves of Buxing I line obtained by mode $d_3$

$$k = \sum_{q_k \cap q_l = \Phi} m_1(q_k) m_2(q_l) = 0.4 \times 1 = 0.4$$

$$m(q_3) = \frac{\sum_{q_k \cap q_l = q_3} m_1(q_k) m_2(q_l)}{\sum_{q_k \cap q_l \neq \Phi} m_1(q_k) m_2(q_l)} = \frac{0.6}{1 - 0.4} = 1$$

Before the fusion, the parent node's supporting is 0.4 to $q_1$ and 0.6 to $q_3$. The parent node does not support $q_2$, $q_4$, and $q_5$. The child nodes support only $q_3$. Once combined, both of the parent node and the child nodes support only $q_3$. The fusion result supports the common part of the diagnosis results, and discards the conflicting ones. The fusion result, i.e., the single phase grounding fault of Line Buxing-I, agrees with the actual fault of the substation.

## 5 Conclusions

By taking into account the structure and technical features of a digital substation, the authors develop a Root Cause Analysis based approach to diagnose faults of transmission and transformation equipment in the controlled area of the substation. The D-S evidence theory is applied to analyze thoroughly the comprehensive fault information obtained with different modes to find the exact root cause or causes. The developed fault diagnosis system can be used to diagnose various faults commonly encountered

in a substation, including malfunctions of protective relays and/or circuit breakers, and miss or false alarms. The diagnosis system can be implemented in a hierarchical structure for multi-level information integration. A real fault scenario was used in the case study to demonstrate the effectiveness of the proposed fault diagnosis system and the performance of the developed software package.

## References


1. Lee, H.J.B., Ahn, S., Park, Y.M.: A fault diagnosis expert system for distribution substations. IEEE Transactions on Power Delivery 15(1), 92–97 (2000)
2. Jung, J.W., Liu, C.C., Hong, M.G., Gallanti, M., Tornielli, G.: Multiple hypotheses and their credibility in on-line fault diagnosis. IEEE Transactions on Power Delivery 16(2), 225–230 (2001)
3. Huang, Y.C.: Fault section estimation in power systems using a novel decision support system. IEEE Transactions on Power Systems 17(2), 438–444 (2002)
4. Yang, H.T., Chang, W.Y., Huang, C.L.: A new neural network approach to on-line fault section estimation using information of protective relays and circuit breakers. IEEE Trans. on Power Delivery 9(1), 220–230 (1994)
5. Cardoso, G.J., Rolim, J.G., Zurn, H.H.: Application of neural-network modules to electric power system fault section estimation. IEEE Transactions on Power Delivery 19(3), 1024–1041 (2004)
6. Huang, J.B., Mu, L.H.: Fault diagnosis of substation based on Petri Nets technology. In: Proceedings of 2006 International Conference on Power System Technology, pp. 1–5 (October 2006)
7. Lo, K.L., Ng, H.S., Trecat, J.: Power systems fault diagnosis using Petri nets. IEE Proceedings: Generation, Transmission and Distribution 144(3), 231–236 (1997)
8. Dong, H.Y., Xue, J.Y.: An approach of substation remote cooperation fault diagnosis based on multi-agents. In: Proceedings of the Fifth World Congress on Intelligent Control and Automation, pp. 5170–5174 (June 2004)
9. Hor, C.L., Crossley, P.A.: Building knowledge for substation-based decision support using rough sets. IEEE Transactions on Power Delivery 22(3), 1372–1379 (2007)
10. Dong, H.Y., Zhang, Y.B., Xue, J.Y.: Hierarchical fault diagnosis for substation based on rough set. In: Proceedings of 2002 International Conference on Power System Technology, pp. 2318–2321 (October 2002)
11. Michel, D., James, D.J.: Improving the reliability of transformer gas-in-oil diagnosis. IEEE Electrical Insulation Magazine 21(4), 21–27 (2005)
12. Silva, K.M., Souza, B.A., Brito, N.S.D.: Fault detection and classification in transmission lines based on wavelet transform and ANN. IEEE Trans. Power Delivery 21(4), 2058–2063 (2006)
13. Nelms, C.R.: The problem with root cause analysis. In: Proceedings of 2007 IEEE 8th Human Factors and Power Plants and HPRCT 13th Annual Meeting, pp. 253–258 (August 2007)
14. Guo, W.X., Wen, F.S., Liao, Z.W., Wei, L.H., Xin, J.B.: An Analytic Model-Based Approach for Power System Alarm Processing Employing Temporal Constraint Network. IEEE Transactions on Power Delivery 25(4), 2435–2447 (2010)

# Generating Fuzzy Partitions from Nominal and Numerical Attributes with Imprecise Values

J.M. Cadenas, M.C. Garrido, and R. Martínez

University of Murcia, Faculty of Informatic, Campus of Espinardo, 30100 Murcia, Spain
{jcadenas,carmengarrido,raquel.m.e}@um.es

**Abstract.** In areas of Data Mining and Soft Computing is important the discretization of numerical attributes because there are techniques that can not work with numerical domains or can get better results when working with discrete domains. The precision obtained with these techniques depends largely on the quality of the discretization performed. Moreover, in many real-world applications, data from which the discretization is carried out, are imprecise. In this paper we address both problems by proposing an algorithm to obtain a fuzzy discretization of numerical attributes from input data that show imprecise values in both numerical and nominal attributes. To evaluate the proposed algorithm we analyze the results on a set of datasets from different real-world problems.

**Keywords:** Fuzzy partition, Imperfect information, Fuzzy random forest ensemble, Imprecise data.

## 1 Introduction

The construction of fuzzy intervals in which a numerical domain is discretized supposes an important problem in the areas of data mining and soft-computing due to the determinate of these intervals can deeply affect the performance of the different classification techniques [1].

Although there are a lot of algorithms to discretization, most of them have not considered that sometimes the information available to construct the partitioning is not as precise and accurate as desirable. However, imperfect information inevitably appears in realistic domains and situations. Instrument errors or corruption from noise during experiments may give rise to information with incomplete data when measuring a specific attribute. In other cases, the extraction of exact information may be excessively costly or unfeasible. Moreover, it might be useful to complement the available data with additional information from an expert, which is usually elicited by imperfect data (interval data, fuzzy concepts, etc). In most real-world problems, data have a certain degree of imprecision. Sometimes, this imprecision is small enough for it to be safely ignored. On other occasions, the imprecision of the data can be modeled by a probability distribution. However, there is a third kind of problems, where the imprecision is significant, and a probability distribution is not the most natural way to model it. This is the case of certain practical problems where the data are inherently fuzzy [2,6,8,10].

When we have imperfect data, we have two options: the first option is to transform the original data for another kind of data which the algorithm can work; the second

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 167–182.
DOI: 10.1007/978-3-642-35638-4_12 

one is to work directly with original data without carrying out any transformation in data. When we choose the first option, we can lose information and therefore, we can lose accuracy. For this reason, it is necessary to incorporate the handling of information with attributes which may present missing and imprecise values in the discretization algorithms.

In this paper we present an algorithm, which we call EOFP (Extended Optimized Fuzzy Partitions) that obtains fuzzy partitions from imperfect information. This algorithm extends the OFP_CLASS algorithm [4] to incorporate the management of imprecise values (intervals and fuzzy values) in numerical attributes, set-valued nominal attributes (nominal attributes with imprecise values) and set-valued classes (imprecise values for the attribute class).

EOFP Algorithm follows the steps of a top-down discretization process with four iterative stages [9]: 1.- All kind of numerical values in the dataset to be discretized are ordered. 2.- The best cut point for partitioning attribute domains is found. 3.- Once the best cut point is found, the domain of each attribute is divided into two partitions. 4.- Finally, we check whether the stopping criterion is fulfilled, and if so the process is terminated.

To implement the above general discretization process, EOFP Algorithm is divided in two stages. In the first stage, we carry out a search of the best cut points for each attribute. In the second stage, based on these cut points, we use a genetic algorithm which optimizes the fuzzy sets formed from the cut points.

The structure of this study is as follows. In Section 2, we are going to present the EOFP Algorithm. In addition, in this section we are going to extend a fuzzy decision tree, which is used as base in the first stage of EOFP algorithm. This tree is able to work with imprecise information both in the values of the attributes and in the class values. Later, in Section 3, we will show various experimental results which evaluate our proposal in comparison with previously existing proposals. For these experiments we will use datasets with imprecision. In Section 4, we will show the conclusions of this study. Finally, we include Appendix A with a brief description of the combination methods used at work.

## 2 Designing the Algorithm

In this section we are going to present the EOFP Algorithm which is able to work with imprecise data. The EOFP Algorithm builds fuzzy partitions which guarantees for each attribute:

- Completeness (no point in the domain is outside the fuzzy partition), and
- Strong fuzzy partition (it verifies that $\forall x \in \Omega_i, \sum_{f=1}^{F_i} \mu_{B_f}(x) = 1$ where $B_1, .., B_{F_i}$ are the $F_i$ fuzzy sets for the partition of the $i$ numerical attribute with $\Omega_i$ domain and $\mu_{B_f}(x)$ are its functions membership).

The domain of each $i$ numerical attribute is partitioned in trapezoidal fuzzy sets, $B_1$, $B_2..$, $B_{F_i}$, so that:

$$\mu_{B_1}(x) = \begin{cases} 1 & b_{11} \leq x \leq b_{12} \\ \frac{(b_{13}-x)}{(b_{13}-b_{12})} & b_{12} \leq x \leq b_{13} \\ 0 & b_{13} \leq x \end{cases} \quad ; \quad \mu_{B_2}(x) = \begin{cases} 0 & x \leq b_{12} \\ \frac{(x-b_{12})}{(b_{13}-b_{12})} & b_{12} \leq x \leq b_{13} \\ 1 & b_{13} \leq x \leq b_{23} \\ \frac{(b_{24}-x)}{(b_{24}-b_{23})} & b_{23} \leq x \leq b_{24} \\ 0 & b_{24} \leq x \end{cases} ;$$

$$\cdots \quad ; \quad \mu_{B_{F_i}}(x) = \begin{cases} 0 & x \leq b_{(F_i-1)3} \\ \frac{(x-b_{(F_i-1)3})}{(b_{(F_i-1)4}-b_{(F_i-1)3})} & b_{(F_i-1)3} \leq x \leq b_{(F_i-1)4} \\ 1 & b_{F_i3} \leq x \end{cases}$$

The EOFP Algorithm is composed for two stages: in the stage 1 we use a fuzzy decision tree. In this stage we get possible cut points to different attributes. In the stage 2 we carry out the process by which we optimize the cut points and make fuzzy partitions. The objective is to divide the numerical domains in fuzzy sets which will be competitive and effective to obtain a good accuracy in the classification task.

Before to describe the EOFP Algorithm, we are going to present a fuzzy decision tree witch is be able to work with imprecise data.

### 2.1 Fuzzy Decision Tree

In this section, we describe a fuzzy decision tree that we will use as base classifier in a Fuzzy Random Forest ensemble to evaluate fuzzy partitions generated and whose basic algorithm will be modified for the first stage of the EOFP Algorithm, as we will see later. This tree is an extension of the fuzzy decision tree that we presented in [4], to incorporate the management of imprecise values.

The tree is built from a set of examples E which are described by attributes which may be nominal expressed with crisp values and with a set of domain values (set-valued nominal attributes) and/or numerical expressed with crisp, interval and fuzzy values where there will be at least one nominal attribute which will act as class attribute. In addition, the class attribute can be expressed with a set of classes (set-valued class). Thus, the class also may be expressed in an imprecise way.

The fuzzy decision tree is based on the ID3 algorithm, where all numerical attributes have been discretized by means of a series of fuzzy sets. An initial value equal to 1 ($\chi_{root}(e_j) = 1$, where $\chi_N(e_j)$ is the membership degree of $e_j$ to node $N$ and $e_j$ is $j$-th example from dataset) is assigned to each example $e_j$ used in the tree learning, indicating that initially $e_j$ is only in the root node of the tree. This value will continue to be 1 as long as the example $e_j$ does not belong to more than one node during the tree construction process. In a classical tree, an example can only belong to one node at each moment, so its initial value (if it exists) is not modified throughout the construction process. But in a fuzzy decision tree, this value is modified when the test in a node is based on an attribute with missing, interval or fuzzy values, or with a set-valued nominal attribute.

**An Attribute with Missing Values.** When the example $e_j$ has a missing value in an attribute $i$ which is used as a test in a node $N$, the example descends to each child node

$N_h, h = 1, ..., H_i$ with a modified value proportionately to the weight of each child node. The modified value for each $N_h$ is calculate as:

$$\chi_{N_h}(e_j) = \chi_N(e_j) \cdot \frac{T\chi_{N_h}}{T\chi_N}$$

where $T\chi_N$ is the sum of the weights of the examples with known value in the attribute $i$ at node $N$ and $T\chi_{N_h}$ is the sum of the weights of the examples with known value in the attribute $i$ that descend to the child node $N_h$.

**An Attribute with Interval and Fuzzy Values.** When the test of a node $N$ is based on attribute $i$ which is numerical, each example in $N$ modifies its weight according to the membership degree of that example to different fuzzy sets of the partition. In this case, the example $e_j$ descends to those child nodes to which it belongs with a degree greater than 0 ($\mu_{B_f}(e_j) > 0; f = 1, ..., F_i$). Due to the characteristics of the partitions we use, the example may descend to two child nodes at most. In this case, $\chi_{N_h}(e_j) = \chi_N(e_j) \cdot \mu_{B_f}(e_j);\ \forall f|\ \mu_{B_f}(e_j) > 0;\ h = f$.

When the test of a node $N$ is based on a numerical attribute $i$ and the value to attribute $i$ in $e_j$ is a fuzzy value different from the set of partitions of the attribute, or an interval value, we need to extend the function that measures the membership degree of these type of data. This new function (denoted $\mu_{simil}(\cdot)$) captures the change in the value $\chi_N(e_j)$, when $e_j$ descends in the fuzzy decision tree. For this reason, the membership degree of $e_j$ is calculated using a similarity measure ($\mu_{simil}(e_j)$) between the value of attribute $i$ in $e_j$ and the different fuzzy sets of the partition of attribute $i$. Therefore, the example $e_j$ can descend to different child nodes. In this case, $\chi_{N_h}(e_j) = \chi_N(e_j) \cdot \mu_{simil}(e_j)$.

Function $\mu_{simil}(e_j)$ is defined, for $f = 1, \ldots, F_i$, as:

$$\mu_{simil}(e_j) = \frac{\int(\min\{\mu_{e_j}(x), \mu_f(x)\})dx}{\sum_{f=1}^{F_i}\int(\min\{\mu_{e_j}(x), \mu_f(x)\})dx} \tag{1}$$

where

- $\mu_{e_j}(x)$ represents the membership function of the fuzzy or interval value of the example $e_j$ in the attribute $i$.
- $\mu_f(x)$ represents the membership function of the fuzzy set of the partition of the attribute i.
- $F_i$ is the cardinality of the partition of the attribute i.

**A Set-Valued Nominal Attribute.** When the test on a node of the tree is based on a nominal attribute, and some examples of that node have a set of domain values for that attribute as value, each of these examples will descend to each child node, according to a set values, with a weight proportional to the weight of each value in the set. So, if in the example $e_j$ of node $N$, the nominal attribute $x$, with domain $\{X_1, X_2, \ldots, X_t\}$, has the value $(P_1\ X_1, P_2\ X_2, \ldots, P_t\ X_t)$ and $x$ is the test in the node, $e_j$ descends to each child node $N_h,\ h = 1, \ldots, t$ with weight $\chi_{N_h}(e_j) = \chi_N(e_j) \cdot P_h$.

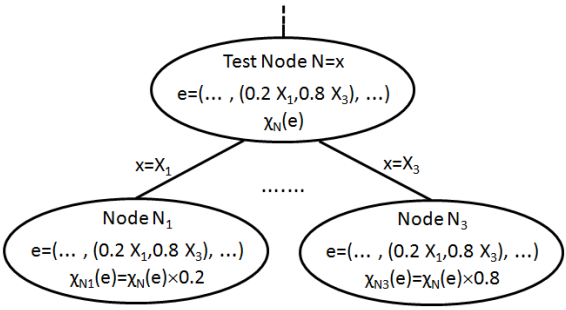



**Fig. 1.** Management of set-valued nominal attributes

For example, let $x$ be a nominal attribute with domain $\{X_1, X_2, X_3\}$. The value of $x$ in an example $e_j$ can be expressed by $(0.2\ X_1, 0.8\ X_3)$ indicating that there are uncertainty in the value and we assign a certainty degree of 0.8 to $x = X_3$ and a certainty degree of 0.2 to $x = X_1$. Figure 1 shows, in an illustrative way, the management of this example in the fuzzy decision tree.

In a general way, we can say that the $\chi_N(e_j)$ value indicates the degree with which the example fulfills the conditions that lead to node $N$ on the tree.

**Calculating the Information Gain in an Extended Fuzzy Decision Tree.** Another aspect of this extended fuzzy decision tree is the way to calculate the information gain when node N (node which is being explored at any given moment) is divided using the attribute $i$ as test attribute. This information gain $G_i^N$ is defined as:

$$G_i^N = I^N - I^{S_{V_i}^N} \tag{2}$$

where:

- $I^N$: Standard information associated with node $N$. This information is calculated as follows:
  1. For each class $k = 1, ..., |C|$, the value $P_k^N$, which is the number of examples in node $N$ belonging to class $k$, is calculated:

$$P_k^N = \sum_{j=1}^{|E|} \chi_N(e_j) \cdot \mu_k(e_j) \tag{3}$$

  where $\chi_N(e_j)$ is the membership degree of example $e_j$ to node $N$ and $\mu_k(e_j)$ is the membership degree of example $e_j$ to class $k$.
  2. $P^N$, which is the total number of examples in node $N$, is calculated.

$$P^N = \sum_{k=1}^{|C|} P_k^N$$

  3. Standard information is calculated as: $I^N = -\sum_{k=1}^{|C|} \frac{P_k^N}{P^N} \cdot \log \frac{P_k^N}{P^N}$

– $I^{S^N_{V_i}}$ is the product of three factors and represents standard information obtained by dividing node $N$ using attribute $i$ adjusted to the existence of missing values in this attribute.

$$I^{S^N_{V_i}} = I_1^{S^N_{V_i}} \cdot I_2^{S^N_{V_i}} \cdot I_3^{S^N_{V_i}}$$

where:

- $I_1^{S^N_{V_i}}$ = 1 - $\frac{P^{N_{m_i}}}{P^N}$, where $P^{N_{m_i}}$ is the weight of the examples in node $N$ with missing value in attribute $i$.
- $I_2^{S^N_{V_i}} = \frac{1}{\sum_{h=1}^{H_i} P^{N_h}}$, $H_i$ being the number of descendants associated with node $N$ when we divide this node by attribute $i$ and $P^{N_h}$ the weight of the examples associated with each one of the descendants.
- $I_3^{S^N_{V_i}} = \sum_{h=1}^{H_i} P^{N_h} \cdot I^{N_h}$, $I^{N_h}$ being the standard information of each descendant $h$ of node $N$.

On the other hand the stopping criterion is the same that we described in [4] which is defined by the first condition reached out of the following: (a) pure node, (b) there aren't any more attributes to select, (c) reaching the minimum number of examples allowed in a node. Besides, it must be pointed out, that once an attribute has been selected as a node test, this attribute will not be selected again due to the fact that all the attributes are nominal or are partitioned.

Having constructed the fuzzy decision tree, we use it to infer an unknown class of a new example. The inference process is as follow:

Given the example $e$ to be classified with the initial value, for instance, $\chi_{root}(e) = 1$, go through the tree from the root node. After obtain the leaf set reached by $e$. For each leaf reached by $e$, calculate the support for each class. The support for a class on a given leaf $N$ is obtained according to the expression (3). Finally, obtain the tree's decision, $c$, from the information provided by the leaf set reached and the value $\chi$ with which example $e$ activates each one of the leaves reached.

With the fuzzy decision tree presented at the moments, we have incorporated numerical attributes with imprecise values which are described by an interval or fuzzy values. Also, we have incorporated nominal attributes expressed by a set of values. In the next subsection we consider the modifications which are necessary to carry out in the phases of learning and classification to incorporate the treatment of examples whose class attribute is set-valued.

**Evaluating Data with Set-Valued Classes.** In the previous section, we have said that the initial weight of one example $e$ may be equal to 1 ($\chi_{root}(e) = 1$) but this value depends on if the example has a single class or it has a set-valued class. In the first case, if the example $e$ has a single class, the initial weight is 1 and in the second case the initial weight will depend on the number of classes that example has. Therefore, if the example $e$ has a set-valued class with $n_{classes}$ classes, the example will be replicated $n_{classes}$ times and each replicate of the example $e$ will have associated the weight $1/n_{classes}$.

In this case, when we perform a test of the tree to classify a dataset with set-valued classes, we can follow the decision process:

If class(e)==class$_{tree}$(e)$\wedge$size(class(e))==1 then successes++
else
If class(e)$\cap$ class$_{tree}$(e)$\neq \emptyset$ then success_or_error++ else errors++

where $class_{tree}$ is the class that fuzzy decision tree provides as output and $class(e)$ is the class value of the example $e$.

As result of this test, we obtain the interval $[min\_error, max\_error]$ where $min\_error$ is calculated considering only errors indicated in the variable errors from the previous process and $max\_error$ is calculated considering as errors $errors + success\_or\_error$.

With this way to classify, the tree receives an imprecise input and its output is imprecise too, because it's not possible to determine exactly a unique error.

One, we have described the fuzzy decision tree that we will use to classify, and that with some modifications, we will use in the stage 1 of the discretization algorithm, we are going to expose such algorithm. As we said earlier, the discretization algorithm EOFP is composed by two stages which we are going to present.

### 2.2 First Stage: Searching for Cut Points

In this stage, a fuzzy decision tree is constructed whose basic process is that described in Subsection 2.1, except that now a procedure based on priority tails is added and there are attributes that have not been discretized. To discretize these attributes, the first step is look for the cut points which will be the border between different partitions. In previous section, we are expose that to discretize attributes, we must order the values. If all data are not crisp, we need a function to order crisp, fuzzy and interval values. To order data, we use the same function that to search for the possible cut points.

To deal with non-discretized attributes, the algorithm follows the basic process in C4.5. The thresholds selected in each node of the tree for these attributes will be the split points that delimit the intervals. Thus, the algorithm that constitutes this first stage is based on a fuzzy decision tree that allows nominal attributes, numerical attributes discretized by means of a fuzzy partition, non-discretized numerical attributes described with crisp, interval and fuzzy values and furthermore it allows the existence of missing values in all of them. Algorithm 1 describes the whole process.

In the step 1, all examples in the root node have an initial weight equal to 1, less the examples with set-valued class whose weight will be initialize as we indicate in the Section 2.1. The tail is a priority tail, ordered from higher to lower according to the total weight of the examples of nodes that form the tail. Thus the domain is guaranteed to partition according to the most relevant attributes.

In the step 3, when we expand a node according to an attribute:

1. If the attribute is already discretized, the node is expanded into many children as possible values the selected attribute has. In this case, the tree's behaviour is similar to that described in the Subsection 2.1.
2. If the attribute is not previously discretized, its possible descendants are obtained. To do this, as in C4.5, the examples are ordered according to the value of the attribute in question. To carry out the order of data with crisp, fuzzy and interval values, we need an ordering index, [13]. Therefore, we have a representative value for each interval

**Algorithm 1.** Search of cut points

**SearchCrispIntervals**($in: E, Fuzzy\ Partition; out: Cut\ points$)
**begin**
(a) Start at the root node, which is placed in the initially empty priority tail. Initially, in the root node is found the set of examples $E$ with an initial weight.
(b) Extract the first node from the priority tail.
(c) Select the best attribute to split this node using information gain expressed in (2) as the criterion. We can find two cases: The first case is where the attribute with the highest information gain is already discretized, either because it is nominal, or else because it had already been discretized earlier by the $Fuzzy\ Partition$. The second case arises when the attribute is numerical and non-discretized. In this case it is necessary to obtain the corresponding cut points.
(d) Having selected the attribute to expand node, all the descendants generated are introduced in the tail.
(e) Go back to step two to continue constructing the tree until there are not nodes in the priority tail or until another stopping condition occurs, such as reaching nodes with a minimum number of examples allowed by the algorithm.
**end**

and fuzzy value and we can order all values of the non-discretized attributes. The index used is calculated as in (4). Let $A_i$ be a fuzzy set (or interval) in the attribute $i$ of the example $e$:

$$Y(A_i) = \int_0^1 M(A_{i\alpha}) d\alpha \tag{4}$$

where $Y(A_i)$ is the representative value of the fuzzy or interval data of the attribute $i$ and $M(A_{i\alpha})$ is the mean value of the elements of $A_{i\alpha}$.

This index determines for each fuzzy or interval value a number with which we order all values. Using the crisp and the representative values, we find the possible cut points as a C4.5 tree. The intermediate value between value of the attribute for example $e_j$ and for example $e_{j+1}$ is obtained. The value obtained will be that which provides two descendants for the node and to which the criterion of information gain is applied. This is repeated for each pair of consecutive values of the attribute, searching for the value that yields the greatest information gain. The value that yields the greatest information gain will be the one used to split the node and will be considered as a cut point for the discretization of this attribute. When example $e$ descend to the two descendants, the process carries out is the same that we explain in Section 2.1 and if the value of the attribute is fuzzy or interval, we apply the function (1) to determine the membership of this example $e$ to the descendant nodes, because we only use the representative value of these kind of values to order and to get cut points, but when we need use these values to do some estimates, we use the original value and not the representative value.

### 2.3 Second Stage: Optimizing Fuzzy Partitions with Imprecise Data

In the second stage of the EOFP Algorithm, we are going to use a genetic algorithm to get the fuzzy sets that make up the partitioning of nondiscretized attributes. We have

decide to use a genetic algorithm, because these algorithms are very powerful and robust, as in most cases they can successfully deal with an infinity of problems from very diverse areas and specifically in Data Mining [5]. These algorithms are normally used in problems without specialized techniques or even in those problems where a technique does exist, but is combined with a genetic algorithm to obtain hybrid algorithms that improve results [7].

The genetic algorithm takes as input the cut points which we have obtained in the first stage, but it is important to mention that the genetic algorithm will decide what cut points are more important to construct the fuzzy partitions, so it is possible that many cut points are not used to obtain the optimal fuzzy partitions. Maximum if the first stage gets $F$ cut points for the attribute $i$, the genetic algorithm can make up $F_i+1$ fuzzy partitions for the attribute $i$. However, if the genetic algorithm considers that the attribute $i$ doesn't have a lot of relevance in the dataset, this attribute won't be partitioned. The different elements which compose this genetic algorithm are as follows:

**Encoding.** An individual will consist of two arrays $v_1$ and $v_2$. The array $v_1$ has a real coding and its size will be the sum of the number of cut points that the fuzzy tree will have provided for each attribute in the first stage. Each gene in array $v_1$ represents the quantity to be added to and subtracted from each attribute's split point to form the partition fuzzy. On the other hand, the array $v_2$ has a binary coding and its size is the same that the array $v_1$. Each gene in array $v_2$ indicates whether the corresponding gene or cut point of $v_1$ is active or not. The array $v_2$ will change the domain of each gene in array $v_1$. The domain of each gene in array $v_1$ is an interval defined by $[0, min(\frac{p_r-p_{r-1}}{2}, \frac{p_{r+1}-p_r}{2})]$ where $p_r$ is the $r$-th cut point of attribute $i$ represented by this gene except in the first ($p_1$) and last ($p_u$) cut point of each attribute whose domains are, respectively: $[0, min(p_1, \frac{p_2-p_1}{2}]$ and $[0, min(\frac{p_u-p_{u-1}}{2}, 1-p_u]$.

When $F_i = 2$, the domain of the single cut point is defined by $[0, min(p_1, 1-p_1]$. The population size will be 100 individuals.

**Initialization.** First the array $v_2$ in each individual is randomly initialized, provided that the genes of the array are not all zero value, since all the split points would be deactivated and attributes would not be discretized. Once initialized the array $v_2$ , the domain of each gene in array $v_1$ is calculated, considering what points are active and which not. After calculating the domain of each gene of the array $v_1$, each gene is randomly initialized generating a value within its domain.

**Fitness Function.** The fitness function of each individual is defined according to the information gain defined in [1]. Algorithm 2 implements the fitness function, where:

- $\mu_{if}$ is the membership function corresponding to fuzzy set $f$ of attribute $i$. Again, we must emphasize that this membership function depends on the kind of attribute. Where if the attribute is numerical or belonging to a known fuzzy partition, the membership function is calculated as we have indicated in 2. On the contrary if the attribute is fuzzy or interval, the membership function is calculated as we show in function (1).
- $E_k$ is the subset of examples of $E$ belonging to class $k$.

  This fitness function, based on the information gain, indicates how dependent the attributes are with regard to class, i.e., how discriminatory each attribute's partitions

**Algorithm 2.** Fitness Function

**Fitness**($in:\ E,\ out: ValueFitness$)
**begin**

1. For each attribute $i = 1, ..., |A|$:
   1.1 For each set $f = 1, ..., F_i$ of attribute $i$
   For each class $k = 1, ..., |C|$ calculate the probability $P_{ifk} = \frac{\sum_{e \epsilon E_k} \mu_{if}(e)}{\sum_{e \epsilon E} \mu_{if}(e)}$
   1.2 For each class $k = 1, ..., |C|$ calculate the probability $P_{ik} = \sum_{f=1}^{F_i} P_{ifk}$
   1.3 For each $f = 1, ..., F_i$ calculate the probability $P_{if} = \sum_{k=1}^{|C|} P_{ifk}$
   1.4 For each $f = 1, ..., F_i$ calculate the information gain of attribute $i$ and set $f\ I_{if} = \sum_{k=1}^{|C|} P_{ifk} \cdot \log_2 \frac{P_{ifk}}{P_{ik} \cdot P_{if}}$
   1.5 For each $f = 1, ..., F_i$ calculate the entropy $H_{if} = -\sum_{k=1}^{|C|} P_{ifk} \cdot \log_2 P_{ifk}$
   1.6 Calculate the $I$ and $H$ total of attribute $i$
   $I_i = \sum_{f=1}^{F_i} I_{if}$ and $H_i = \sum_{f=1}^{F_i} H_{if}$
2. Calculate the fitness as : $ValueFitness = \frac{\sum_{i=1}^{|A|} I_i}{\sum_{i=1}^{|A|} H_i}$

**end**

are. If the fitness we obtain for each individual is close to zero, it indicates that the attributes are totally independent of the classes, which means that the fuzzy sets obtained do not discriminate classes. On the other hand, as the fitness value moves further away from zero, it indicates that the partitions obtained are more than acceptable and may discriminate classes with good accuracy.

**Selection.** Individual selection is by means of tournament, taking subsets with size 2.

**Crossing.** The crossing operator is applied with a probability of $0.3$, crossing two individuals through a single point, which may be any one of the positions on the vector. Not all crossings are valid, since one of the restrictions imposed on an individual is that the array $v_2$ should not has all its genes to zero. When crossing two individuals and this situation occurs, the crossing is invalid, and individuals remain in the population without interbreeding. If instead the crossing is valid, the domain for each gene of array $v_1$ is updated in individuals generated.

**Mutation.** Mutation is carried out according to a certain probability at interval $[0.01, 0.1]$, changing the value of one gene to any other in the possible domain. First, the gene of the array $v_2$ is mutated and then checked that there are still genes with value 1 in $v_2$. In this case, the gene in $v_2$ is mutated and, in addition, the domains of this one and its adjacent genes are updated in the vector $v_1$. Finally, the mutation in this same gene is carried out in the vector $v_1$.

If when a gene is mutated in $v_2$ all genes are zero, then the mutation process is not produced.

**Stopping.** The stopping condition is determined by the number of generations situated at interval $[100, 150]$.

The genetic algorithm should find the best possible solution in order to achieve a more efficient classification.

In the next section we want to show with some computational experiments that it is important construct fuzzy partitions from real data versus transform them because we will lost information and accuracy.

## 3 Experiments

In this section we are going to show different experiments to evaluate if the fuzzy partitions which are constructed without making any transform of data (EOFP Algorithm) are better than fuzzy partitions which are constructed making certain transformation on imprecise data to convert them in crisp data (OFP_CLASS algorithm). All partitions are evaluated classifying with a Fuzzy Random Forest ensemble (FRF) [3] which is able to handle imperfect data into the learning and the classification phases.

The experiments are designed to measure the behavior of fuzzy partitions used in the FRF ensemble using datasets and results proposed in [11,12] where the authors use a fuzzy rule-based classifier to classify datasets with imprecise data such as missing or interval. Also they use uniform partitions to evaluate the datasets and we are going to show how the results are better when the partitions are fuzzy although they are constructed using the modified dataset instead of the original dataset. Also we are going to show how the results in classification are still better if we don't modify data to construct the fuzzy partitions. Due to we are going to compare with results of [11,12], we define the experimental settings quite similar to those proposed by them.

### 3.1 Datasets and Parameters for FRF Ensemble

To evaluate fuzzy partitions, we have used real-world datasets about medical diagnosis and high performance athletics [11,12], that we describe in Table 1.

**Table 1.** Datasets

| Dataset | \|E\| | \|M\| | I | Dataset | \|E\| | \|M\| | I |
|---|---|---|---|---|---|---|---|
| 100ml-4-I | 52 | 4 | 2 | Dyslexic-12 | 65 | 12 | 4 |
| 100ml-4-P | 52 | 4 | 2 | Dyslexic-12-01 | 65 | 12 | 3 |
| Long-4 | 25 | 4 | 2 | Dyslexic-12-12 | 65 | 12 | 3 |

Table 1 shows the number of examples ($|E|$), the number of attributes ($|M|$) and the number of classes ($I$) for each dataset. "Abbr" indicates the abbreviation of the dataset used in the experiments.

All FRF ensembles use a forest size of 100 trees. The number of attributes chosen at random at a given node is $\log_2(|\cdot|+1)$, where $|\cdot|$ is the number of available attributes at that node, and each tree of the FRF ensemble is constructed to the maximum size (node pure or set of available attributes is empty) and without pruning.

### 3.2 Results

These experiments were conducted to test the accuracy of FRF ensemble when it uses fuzzy partitions constructed from real-world datasets with imperfect values

using EOFP Algorithm. These results are compared with the ones obtained by the GFS classifier proposed in [11], which uses uniform partitions and with the results obtained by FRF ensemble when uses fuzzy partitions constructed with the OPF_CLASS Algorithm.

It is important to clarify that OPF_CLASS Algorithm doesn't work with imperfect data. For this reason, to get the fuzzy partitions of these datasets we have modified the original data. The interval or fuzzy values have been changed by their average values. Therefore we have transformed the interval and fuzzy values in crisp values and of this way the OPF_CLASS Algorithm can work with these datasets.

In these experiments we have used the available datasets in "http://sci2s.ugr.es/ keel/" and the available results in [11,12]. There are datasets from two different real-world problems. The first one is related to the composition of teams in high performance athletics and the second one is a medical diagnosis problem. A more detailed description of these problems may be found in [11,12].

**High Performance Athletics.** The score of an athletics team is the sum of the individual scores of the athletes in the different events. It is the coach's responsibility to balance the capabilities of the different athletes in order to maximize the score of a team according to the regulations. The variables that define each problem are as follows:

- There are four indicators for the long jump that are used to predict whether an athlete will pass a given threshold: the ratio between the weight and the height, the maximum speed in the 40 meters race, and the tests of central (abdominal) muscles and lower extremities;
- There are also four indicators for the 100 meters race: the ratio between weight and height, the reaction time, the starting or 20 m speed, and the maximum or 40 m speed.

The datasets used in this experiment are the following: "Long-4" (25 examples, 4 attributes, 2 classes, no missing values and all attributes are interval-valued), "100ml-4-I" and "100ml-4-P" (52 examples, 4 attributes, 2 classes, no missing values and all attributes are interval-valued).

As in [11], we have used a 10 fold cross-validation design for all datasets. Table 2 shows the results obtained in [11] and the ones obtained by the FRF ensemble with the six combination methods which are explained in detail in [3]. In Appendix 4 we present a brief intuitive description of each of them. Except for the crisp algorithm proposed in [11], in Table 2, the interval $[mean\_min\_error, mean\_max\_error]$ obtained for each dataset according to the decision process described in Section 2.1, is shown. For each dataset, the best results obtained with each algorithm are underlined.

The results obtained in classification by the extended GPS proposed in [11] and FRF ensemble, are very promising because we are representing the information in a more natural and appropriate way, and in this problem, we are allowing the collection of knowledge of the coach by ranges of values and linguistic terms.

The results of FRF ensemble are very competitive with all fuzzy partitions but the fuzzy partitions obtained with EOFP Algorithm are the best.

**Table 2.** Comparative results for datasets of high performance athletics

| | | Dataset | | | | | |
|---|---|---|---|---|---|---|---|
| | | 100ml-4-I | | 100ml-4-P | | Long-4 | |
| | Technique | Train | Test | Train | Test | Train | Test |
| EOFP Fuzzy partition | $FFR_{SM1}$ | [0.107,0.305] | [0.130,0.323] | [0.043,0.235] | [0.093,0.290] | [0.191,0.484] | [0.083,0.349] |
| | $FRF_{SM2}$ | [0.110,0.306] | [0.150,0.343] | [0.045,0.237] | [0.110,0.307] | [0.165,0.449] | [0.083,0.349] |
| | $FRF_{MWL1}$ | [0.070,0.265] | [0.073,0.267] | [0.032,0.224] | [0.060,0.257] | [0.085,0.364] | [0.033,0.299] |
| | $FRF_{MWL2}$ | [0.060,0.254] | [0.113,0.306] | [0.043,0.235] | [0.060,0.257] | [0.111,0.391] | [0.083,0.349] |
| | $FRF_{MWLT1}$ | [0.070,0.267] | [0.073,0.267] | [0.032,0.224] | [0.060,0.257] | [0.085,0.364] | [0.033,0.299] |
| | $FRF_{MWLT2}$ | [0.060,0.252] | [0.093,0.286] | [0.038,0.231] | [0.060,0.257] | [0.107,0.386] | [0,083,0.349] |
| OFP_CLASS Fuzzy partition | $FFR_{SM1}$ | [0.139,0.331] | [0.150,0.343] | [0.098,0.291] | [0.133,0.310] | [0.120,0.404] | [0.200,0.467] |
| | $FRF_{SM2}$ | [0.141,0.333] | [0.150,0.343] | [0.096,0.288] | [0.093,0.290] | [0.115,0.391] | [0.200,0.467] |
| | $FRF_{MWL1}$ | [0.077,0.269] | [0.093,0.287] | [0.075,0.269] | [0.073,0.270] | [0.116,0.396] | [0.100,0.417] |
| | $FRF_{MWL2}$ | [0.060,0.252] | [0.093,0.287] | [0.077,0.269] | [0.073,0.270] | [0.102,0.382] | [0.100,0.367] |
| | $FRF_{MWLT1}$ | [0.077,0.269] | [0.093,0.287] | [0.075,0.267] | [0.073,0.270] | [0.107,0.387] | [0.150,0.417] |
| | $FRF_{MWLT2}$ | [0.062,0.254] | [0.093,0.287] | [0.077,0.269] | [0.073,0.270] | [0.094,0.373] | [0.067,0.333] |
| | Crisp [11] | 0.259 | 0.384 | 0.288 | 0.419 | 0.327 | 0.544 |
| | GGFS [11] | [0.089,0.346] | [0.189,0.476] | [0.076,0.320] | [0.170,0.406] | [0.000,0.279] | [0.349,0.616] |

**Diagnosis of Dyslexic.** Dyslexia is a learning disability in people with normal intellectual coefficient, and without further physical or psychological problems explaining such disability. A more detailed description of this problem can found in [11,12].

In these experiments, we have used three different datasets. Their names are "Dyslexic-12', "Dyslexic-12-01" and "Dyslexic-12-12". Each dataset has 65 examples and 12 attributes. The output variable for each dataset is a subset of the labels that follow: - No dyslexic; - Control and revision; - Dyslexic; and - Inattention, hyperactivity or other problems.

These three datasets differ only in their outputs:

- "Dyslexic-12" comprises the four mentioned classes.
- "Dyslexic-12-01" does not make use of the class "control and revision", whose members are included in class "no dyslexic".
- "Dyslexic-12-12" does not make use of the class "control and revision", whose members are included in class "dyslexic".

All experiments are repeated 100 times for bootstrap resamples with replacement of the training set. The test set comprises the "out of the bag" elements.

In Table 3, we show the results obtained when we run FRF ensemble with fuzzy partitions obtained with OFP_CLASS and fuzzy partitions obtained with EOFP for datasets "Dyslexic-12", "Dyslexic-12-01" and"Dyslexic-12-12'.

Also, in Table 3, we compare these results with the best ones obtained in [12] ($(*)$: partition used - four labels; $(**)$ partition used - five labels). Again, in this table, the interval $[mean\_min\_error, mean\_max\_error]$ obtained for each dataset according to

**Table 3.** Comparative results for datasets of dyslexia

| | | Dataset | | | | | |
|---|---|---|---|---|---|---|---|
| | | Dyslexic-12 | | Dyslexic-12-01 | | Dyslexic-12-12 | |
| | Technique | Train | Test | Train | Test | Train | Test |
| EOFP Fuzzy partition | $FRF_{SM1}$ | [0.000,0.238] | [0.000,0.398] | [0.022,0.223] | [0.039,0.377] | [0.001,0.263] | [0.035,0.422] |
| | $FRF_{SM2}$ | [0.000,0.228] | [0.000,0.399] | [0.008,0.184] | [0.022,0.332] | [0.009,0.245] | [0.032,0.411] |
| | $FRF_{MWL1}$ | [0.000,0.270] | [0.000,0.406] | [0.017,0.231] | [0.045,0.383] | [0.001,0.273] | [0.019,0.430] |
| | $FRF_{MWL2}$ | [0.000,0.270] | [0.000,0.407] | [0.020,0.241] | [0.056,0.385] | [0.001,0.267] | [0.026,0.406] |
| | $FRF_{MWLT1}$ | [0.000,0.263] | [0.000,0.402] | [0.012,0.216] | [0.038,0.365] | [0.000,0.265] | [0.019,0.427] |
| | $FRF_{MWLT2}$ | [0.000,0.266] | [0.000,0.404] | [0.015,0.221] | [0.049,0.373] | [0.000,0.262] | [0.024,0.422] |
| OFP_CLASS Fuzzy partition | $FRF_{SM1}$ | [0.000,0.320] | [0.002,0.511] | [0.000,0.282] | [0.000,0.413] | [0.000,0.405] | [0.000,0.477] |
| | $FRF_{SM2}$ | [0.000,0.327] | [0.001,0.515] | [0.000,0.253] | [0.000,0.389] | [0.000,0.402] | [0.000,0.469] |
| | $FRF_{MWL1}$ | [0.000,0.261] | [0.003,0.419] | [0.000,0.264] | [0.000,0.400] | [0.000,0.335] | [0.000,0.422] |
| | $FRF_{MWL2}$ | [0.000,0.270] | [0.003,0.423] | [0.000,0.276] | [0.000,0.407] | [0.000,0.343] | [0.000,0.414] |
| | $FRF_{MWLT1}$ | [0.000,0.264] | [0.004,0.419] | [0.000,0.243] | [0.000,0.386] | [0.000,0.331] | [0.000,0.422] |
| | $FRF_{MWLT2}$ | [0.000,0.267] | [0.003,0.417] | [0.000,0.259] | [0.000,0.394] | [0.000,0.343] | [0.000,0.418] |
| (*) | Crisp $CF^0$ | 0.444 | [0.572,0.694] | 0.336 | [0.452,0.533] | 0.390 | [0.511,0.664] |
| (*) | GGFS | – | [0.421,0.558] | – | [0.219,0.759] | – | [0.199,0.757] |
| (*) | GGFS $CF^0$ | [0.003,0.237] | [0.405,0.548] | [0.005,0.193] | [0.330,0.440] | [0.003,0.243] | [0.325,0.509] |
| (**) | Crisp $CF^0$ | 0.556 | [0.614,0.731] | 0.460 | [0.508,0.605] | 0.485 | [0.539,0.692] |
| (**) | GGFS | – | [0.490,0.609] | – | [0.323,0.797] | – | [0.211,0.700] |
| (**) | GGFS $CF^0$ | [0.038,0.233] | [0.480,0.621] | [0.000,0.187] | [0.394,0.522] | [0.000,0.239] | [0.393,0.591] |

the decision process described in Section 2.1, is shown. For each dataset, the best results obtained with each algorithm are underlined.

As comment about all experiments, we see that FRF ensemble with EOFP fuzzy partitions obtains better results in test than FRF with OFP_CLASS fuzzy partitions. FRF ensemble is a significant improvement over the crisp GFS. In these experiments we can see that when the partitions are obtained with the original data using the EOFP algorithm, the accuracy is higher (the intervals of error are closer to 0 and they are less imprecise). As also discussed in [12] is preferable to use an algorithm which is able of learning with low quality data than removing the imperfect information and using a conventional algorithm.

## 4 Conclusions

In this paper we have presented the EOFP Algorithm for fuzzy discretization of numerical attributes. This algorithm is able to work with imperfect information. We have performed several experiments using imprecise datasets, obtaining better results when working with the original data. Besides, we have presented a fuzzy decision tree which can work with imprecise information.

Our final conclusion, as many papers in the literature are indicating, is that it is necessary to design classification techniques so they can manipulate original data that can be imperfect in some cases. The transformation of these imperfect values to (imputed) crisp values may cause undesirable effects with respect to accuracy of the technique.

**Acknowledgements.** Partially supported by the project TIN2011-27696-C02-02 of the MINECO of Spain. Thanks to the Funding Program for Research Groups of Excellence (04552/ GERM/06) and the scholarship FPI of Raquel Martínez granted by the "Agencia Regional de Ciencia y Tecnología - Fundación Séneca", Murcia, Spain.

## References

1. Au, W.-H., Chan, K.C., Wong, A.: A fuzzy approach to partitioning continuous attributes for classification. IEEE Tran., Knowledge and Data Engineering 18(5), 715–719 (2006)
2. Bonissone, P.P.: Approximate reasoning systems: handling uncertainty and imprecision in information systems. In: Motro, A., Smets, P. (eds.) Uncertainty Management in Information Systems: From Needs to Solutions, pp. 369–395. Kluwer Academic Publishers (1997)
3. Bonissone, P.P., Cadenas, J.M., Garrido, M.C., Díaz-Valladares, R.A.: A fuzzy random forest. Int. J. Approx. Reasoning 51(7), 729–747 (2010)
4. Cadenas, J.M., Garrido, M.C., Martínez, R., Muñoz, E.: OFP_CLASS: An Algorithm to Generate Optimized Fuzzy Partitions to Classification. In: 2nd International Conference on Fuzzy Computation, pp. 5–13 (2010)
5. Cantu-Paz, E., Kamath, C.: On the use of evolutionary algorithms in data mining. In: Abbass, H.A., Sarker, R.A., Newton, C.S. (eds.) Data Mining: A Heuristic Approach, pp. 48–71. Ideal Group Publishing (2001)
6. Casillas, J., Sánchez, L.: Knowledge extraction from data fuzzy for estimating consumer behavior models. In: IEEE Confer. on Fuzzy Systems, pp. 164–170 (2006)
7. Cox, E.: Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration. Morgan Kaufmann Publishers (2005)
8. Garrido, M.C., Cadenas, J.M., Bonissone, P.P.: A classification and regression technique to handle heterogeneous and imperfect information. Soft Computing 14(11), 1165–1185 (2010)
9. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: an enabling technique. Journal of Data Mining and Knowledge Discovery 6(4), 393–423 (2002)
10. Otero, A.J., Sánchez, L., Villar, J.R.: Longest path estimation from inherently fuzzy data acquired with GPS using genetic algorithms. In: International Symposium on Evolving Fuzzy Systems, pp. 300–305 (2006)
11. Palacios, A.M., Sánchez, L., Couso, I.: Extending a simple genetic coopertative-competitive learning fuzzy classifier to low quality datasets. Evolutionary Intelligence 2, 73–84 (2009)
12. Palacios, A.M., Sánchez, L., Couso, I.: Diagnosis of dyslexia with low quality data with genetic fuzzy systems. Int. J. Approx. Reasoning 51, 993–1009 (2010)
13. Wang, X., Kerre, E.E.: Reasonable propierties for the ordering of fuzzy quantities (I-II). Journal of Fuzzy Sets and Systems 118, 375–405 (2001)

# Appendix

## Combination Methods

We present, with a brief intuitive description, the combination methods used in this paper. These methods are described with more details in [3].

- Method SM1: In this method, each tree of the ensemble assigns a simple vote to the most voted class among the reached leaves by the example. The FRF ensemble classifies the example with the most voted class among the trees.
- Method SM2: The FRF ensemble classifies the example with the most voted class among the reached leaves by the example.
- Method MWL1: This method is similar to SM1 method but the vote of each reached leaf is weighted by the weight of the leaf.
- Method MWL2: In this case, each leaves reached assigns a weight vote to the majority class. The ensemble decides the most voted class.
- Method MWLT1: This method is similar to MWL1 method but the vote of each tree is weighted by a weight assigned to each tree.
- Method MWLT2: Each leaf reached vote to the majority class with a weighted vote with the weight of the leaf and the tree to which it belongs.

# A New Way to Describe Filtering Process Using Fuzzy Logic: Towards a Robust Density Based Filter

Philippe Vautrot[1], Michel Herbin[2], and Laurent Hussenet[2]

[1] CReSTIC EA 3804, University of Reims Champagne Ardenne, Department of Informatique, IUT Info, rue des Crayères, BP 1035, 51687 Reims Cedex 2, France
[2] CReSTIC EA 3804, University of Reims Champagne Ardenne, Department of Informatique, IUT RCC, Chaussée du port, BP 541, 51012 Châlons-en-Champagne, France
{philippe.vautrot,michel.herbin,laurent.hussenet}@univ-reims.fr

**Abstract.** Image denoising is a well-known preprocessing step that can help for further processing tasks. With the increase of acquisition device performance, multicomponent images tend now to be widely used. To deal with, this paper proposes to describe usual noise reduction methods in the scope the fuzzy logic. The denoising process can be describe by a fuzzification step, some aggregations and a defuzzification step. To illustrate the concept, the bilateral filter is reformulated in the field of fuzzy logic. It is then extended to take into account impulse noise by using a density based function in the fuzzification step. This leads to a robust filter against outliers.

## 1 Introduction

In the framework of image processing, one of the first tasks consists in removing or reducing noise from the images [1]. The improvement of acquisition devices increases the need for processing multicomponent images obtained from different channels [2,3]. The independent processing of image components turns out to be inappropriate and leads to strong artifacts [4]. Thus the noise reduction of multicomponent images is an active field of research in satellite remote sensing, robot guidance, electron microscopy, medical imaging, color processing and real-time applications [5,6,7]. This paper focuses on this preprocessing step for reducing both additive Gaussian noise and impulse noise. Additive Gaussian noise corrupts images because of the imprecision of acquisition devices. Impulse noise is generally produced by the transmission devices [3].

The noise reduction consists in filtering the image, classically by computing a barycenter within a window. The selection of barycentric coordinates is the main key of noise reduction methods. The fuzzy techniques also addresses this issue of noise reduction [8,9,10]. In this paper, we consider that the filtering window is a fuzzy set. First we determine these fuzzy sets associated to each pixel. This step corresponds to a fuzzification of the pixels. Second the estimation of the filtered value corresponds to a defuzzification [11]. Moreover the pixels of a multi-component image have both 2-dimensional spatial coordinates and n-dimenional photometric coordinates associated with the n components of the image. The bilateral filtering is a classical way taking into account both the spatial aspect and the photometric aspect of images in image processing. Bilateral filter of Tomasi and Manduchi [12] is the archetype of such

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 183–195.
DOI: 10.1007/978-3-642-35638-4_13 

bilateral approach. Thanks to the agregation operators [13], the fuzzy logic enables us to generalize the bilateral approach of filtering. Unfortunately Bilateral filter is not robust against outliers. Thus this paper proposes a new bilateral filter based on density estimation that provides robustness against outliers.

The paper is organized as follows: Section 2 presents the general framework selecting fuzzy neighborhood of each pixel for image filtering. Section 3 is devoted to the defuzzification step for estimating the filtered value of a pixel. In Section 4 we study the combination of fuzzy neighborhood improving the classical bilateral filtering [12]. This approach is applied to reduce Gaussian noise and impulse noise in color images. The last Section proposes a discussion and concludes this paper.

## 2 Fuzzy Neighborhood of a Pixel

Let $p$ be a pixel of a multicomponent image $I$ with $d$ components. Let $I(p)$ be its photometric vector. Reducing the noise, $I(p)$ is replaced by the filtered value $I^*(p)$ which is estimated within a window $W_p$ centered on $p$. Let $p_1, p_2, ...p_N$ be the $N$ pixels of $W_p$ ($N = n \times n$). $I^*(p)$ is usually a barycenter of $I(p_1)$, $I(p_2)$,... $I(p_N)$ defined by:

$$I^*(p) = \frac{1}{\sum\limits_{1 \leq i \leq N} \mu(i)} \sum_{1 \leq i \leq N} \mu(i) I(p_i) \tag{1}$$

where $\mu(i)$ are the barycentric coordinates of $I^*(p)$.

In the fuzzy logic frame, $\mu(i)$ becomes the membership value of the pixel $p_i$ to a fuzzy set $\widetilde{p}$. This fuzzy set has its support in $W_p$. Then the first step of the filtering procedure consists in selecting this fuzzy neighborhood of $p$. This fuzzification step is detailed in the following subsections.

### 2.1 Fuzzy Spatial Neighborhood

When the membership values $\mu(i)$ depend only on the spatial locations of the pixels $p_i$, then a fuzzy spatial neighborhood $\widetilde{p}_{spat}$ is defined for filtering. Gaussian filter is the archetype of these spatial filters. The membership values $\mu_{spat}(i)$ are defined by:

$$\mu_{spat}(i) = \exp\left( -\frac{dist^2_{spat}(p, p_i)}{2\sigma^2_{spat}} \right) \tag{2}$$

where $dist_{spat}$ is the Euclidean distance and $\sigma_{spat}$ is the standard deviation of the Gaussian filter. Note that these fuzzy neighborhoods are normalized fuzzy sets [14] and their largest membership values are equal to 1.

### 2.2 Fuzzy Photometric Neighborhood

When the membership values depend only on the closeness between the photometric values $I(p_i)$ and $I(p)$, then the fuzzy neighborhood of $p$ is designed in the photometric space. Rank filter or vector median filters [15] give examples of such photometric filters. They are obtained by ordering the vectors $I(p_1)$, $I(p_2)$,... $I(p_N)$. The estimation of $I^*(p)$ is based on the ranks of $I(p_i)$ vectors. In such cases, the membership values $\mu_{phot}(i)$ of the fuzzy photometric neighborhood $\widetilde{p}_{phot}$ ignore the spatial location of the pixels $p_i$.

By analogy to the fuzzy spatial neighborhood, the Gaussian distribution also permits to give another definition of $\widetilde{p}_{phot}$. The support of the fuzzy set remains $W_p$. But the distance $dist_{phot}$ is computed in the photometric space (e.g. Euclidean distance). Then the membership function is defined by:

$$\mu_{phot}(i) = \exp\left(-\frac{dist^2_{phot}(I(p), I(p_i))}{2\sigma^2_{phot}}\right) \tag{3}$$

where $\sigma_{phot}$ is the standard deviation of the Gaussian distribution in the photometric domain.

Because of the noise, $I(p)$ could be inappropriate as the center of a photometric neighborhood. Therefore we propose another approach for defining a fuzzy photometric neighborhood of $p$.

### 2.3 Fuzzy Neighborhood Based on Density

For each pixel $q$ in $W_p$ $\widetilde{p}^q_{phot}$ is a fuzzy neighborhood of $p$ centered on $I(q)$. The membership functions of $\widetilde{p}^q_{phot}$ are defined by:

$$\mu^q_{phot}(i) = \exp\left(-\frac{dist^2_{phot}(I(q), I(p_i))}{2\sigma^2_{phot}}\right) \tag{4}$$

where $q \in W_p$. These $N$ fuzzy sets are aggregated using the arithmetic mean of their membership functions. Then the function $\mu_{dens}$ we obtain corresponds to a local estimation of a probability density function (PDF). Improving PDF estimation we preserve against outliers and noise by ruling out $\widetilde{p}^p_{phot}$ (i.e. $\widetilde{p}_{phot}$) when estimating the density [16]. The membership function of this new fuzzy set based on density is defined by:

$$\mu_{dens}(i) = \frac{1}{C} \sum_{q \in W_p, q \neq p} \mu^q_{phot}(i) \tag{5}$$

where $C$ is a normalization coefficient. This paper proposes this approach through robust density estimation to define a new fuzzy photometric neighborhood.

### 2.4 Bilateral Approach of Fuzzy Neighborhood

To keep the advantage of both spatial and photometric approaches, the t-norms [14] (i.e. a conjunction operator) permit to combine the fuzzy spatial neighborhood and the

fuzzy photometric neighborhood. Tomasi and Manduchi [12] use the algebraic t-norm for computing their bilateral filter. Then the membership values of $\widetilde{p}_{bilat}$ is defined by:

$$\mu_{bilat}(i) = \mu_{spat}(i) \ \times \ \mu_{phot}(i) \tag{6}$$

In this paper, we use the classical minimum operator as t-norm combining both spatial and photometric density-based neighborhoods. The fuzzy bilateral neighborhood $\widetilde{p}_{bidens}$ we propose is the conjunction of these two fuzzy sets. Therefore the membership function $\mu_{bidens}$ is defined by:

$$\mu_{bidens}(i) = \min\Big(\mu_{spat}(i),\ \mu_{dens}(i)\Big) \tag{7}$$

## 3 Defuzzification

The goal of this section is to estimate the filtered value $I^*(p)$ from the fuzzy neighborhoods of $p$. This step corresponds to a defuzzification process (see a review of the defuzzification methods in [11]). The defuzzification is obtained using two stages: the first one operates in the spatial domain and the second one operates in the photometric domain.

The most classical defuzzification method is based on the maximum of membership values. In the context of multicomponent images, the maxima method in the spatial domain consists in selecting the pixel $p_i$ for which the membership value $\mu(i)$ is maximal. Let $\overline{p}$ be this pixel defined by:

$$\overline{p} = arg\ max_{p_i \in W_p}\Big(\mu(i)\Big) \tag{8}$$

In this paper, the membership function $\mu_{bidens}$ is used to determine $\overline{p}$. Therefore $\overline{p}$ corresponds to the mode of our density estimation.

Another usual defuzzification method consists in computing the center of gravity of a fuzzy set where the weights are the membership values. This method is used in the photometric domain. $I(\overline{p})$ is considered as the center of the fuzzy photometric neighborhood of $p$. Then the filtered value $I^*(p)$ is defined by:

$$I^*(p) = \frac{1}{\sum\limits_{1 \leq i \leq N} \mu^{\overline{p}}_{phot}(i)} \quad \sum_{1 \leq i \leq N} \mu^{\overline{p}}_{phot}(i) I(p_i) \tag{9}$$

Indeed this barycenter inside the window $W_p$ is the filtered value we propose to reduce noise in multicomponent images.

## 4 Application to Color Images

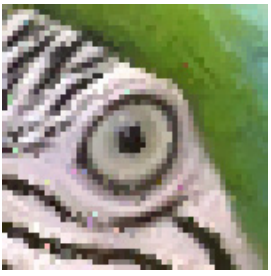
To assess our method, we use color images with three components: Red, Green and Blue. Images are corrupted with two kinds of independent and identically

(a) Reference Image (Parrots)

(b) Noised Image

(c) Vector Median Filter

(d) Bilateral Filter



(e) Density-based Filter

**Fig. 1.** Comparison of noise reduction filters: (a) Reference image (Parrots), (b) Part of Noised image (Noised), (c) Vector Median filtered image (VM), (d) Bilateral filtered image (BILAT), (e) Fuzzy Density-based filtered image (DENS)

distributed noise. A low level noise is designed through additive Gaussian noise, and high level noise is modeled by impulse noise. The goal is to reduce both low level noise and high level noise by filtering corrupted images.

The classical mean squared error (MSE) evaluates the results by averaging the squared differences of filtered and reference images. In this context $MSE$ is defined by:

$$MSE(I^*) = \frac{1}{\#I} \quad \sum_{p \in I} dist_{phot}(I^*(p), I(p))^2 \tag{10}$$

where $\#I$ is the number of pixels of the images. We separate MSE into two parts $MSE^-$ and $MSE^+$. $MSE^-$ is defined by:

$$MSE^-(I^*) = \frac{1}{\#I^-} \sum_{\delta(p) \leq T} \delta^2(p) \tag{11}$$

where $I^- = \{p : \delta(p) \leq T\}$, and $MSE^+$ is defined by:

$$MSE^+(I^*) = \frac{1}{\#I^+} \sum_{\delta(p) > T} \delta^2(p) \tag{12}$$

where $I^+ = \{p : \delta(p) > T\}$. In this paper, the threshold $T = 10$ is used to separate low level noise and high level noise.

**Table 1.** Assessments of noised image (Noised), vector median filtered image (VM), bilateral filtered image (BILAT) and fuzzy density-based filtered image (DENS) using $MSE$ with 388, 112 pixels, $MSE^-$ with $N^-$ pixels, and $MSE^+$ with $N^+$ pixels ($N^+ + N^- = 388, 112$)

| Image | $MSE$ | $N^-$ | $MSE^-$ |
|---|---|---|---|
| Noised | 2873.6 | 256,921 | 46.1 |
| VM | 98.2 | 338,773 | 32.9 |
| BILAT | 2803.0 | 336,865 | 28.6 |
| DENS | 69.0 | 370,588 | 21.3 |

| Image | $MSE$ | $N^+$ | $MSE^+$ |
|---|---|---|---|
| Noised | 2873.6 | 131,191 | 8410.9 |
| VM | 98.2 | 49,339 | 545.6 |
| BILAT | 2803.0 | 51,247 | 21040.3 |
| DENS | 69.0 | 17,524 | 1077.5 |

In this paper, the filtering windows has $5 \times 5$ pixels. Estimating the density in the photometric space, a large value of $\sigma_{phot}$ is prefered for smoothing PDF estimation. Then we use $\sigma_{phot} = 30.0$. In the spatial domain, $\sigma_{spat}$ is empirically determined ($\sigma_{spat} = 0.5$). In the defuzzification process, $\sigma_{phot}$ value controls the smoothing when filtering. The value which gives the best results is $\sigma_{phot} = 10$.

Evaluating the results, we compare a corrupted image (Noised), a classical bilateral filtered image (BILAT), a vector median filtered image (VM) and our fuzzy density-based filtered image (DENS). Table 1 gives the mean square errors obtained when assessing the noise reduction. These results show that the bilateral filter is inappropriate in the case of high level noise (i.e. outliers) and vector median filter cannot smooth enough the image for reducing low level noise when preserving the edges. Figure 1 confirm these results.

The defuzzification process uses a weighted mean of the photometric vectors which permits to smooth the image. The higher $\sigma_{phot}$ value, the smoother the image. If $\sigma_{phot}$ is too small, then the filter does not smooth the filtered image. Therefore it does not enough reduce low level noise. If $\sigma_{phot}$ is too large, the fine details could disappear

(a) Reference (b) $\sigma_{phot} = 5$

(c) $\sigma_{phot} = 10$ (d) $\sigma_{phot} = 20$

**Fig. 2.** Reducing noise and level of texture: (a) part of a reference image and fuzzy density-baser filtered images with (b) $\sigma_{phot} = 5$, (c) $\sigma_{phot} = 10$, (d) $\sigma_{phot} = 20$

when filtering because of a too large smoothing. Figure 2 displays the results obtained with $\sigma_{phot} = 5$, $\sigma_{phot} = 10$, and $\sigma_{phot} = 20$. The value $\sigma_{phot} = 10$ gives convenient results between smoothing for reducing low level noise and preserving details.

We have compared our fuzzy density-based filter (DENS) with two recent fuzzy filters. The first one is the fuzzy rank-ordered differences (FROD) statistic based filter [10] which uses a fuzzy metric to decide if a pixel is an outlier or not. This filter is only adapted to impulse noise. The second one is the fuzzy peer group (FPG) filter [9] which extends the concept of peer group in the fuzzy setting. This filter is able to process impulse noise as well as Gaussian noise. To evaluate the performance of these filters, the test images Caps, Flower, Motorbikes and Parrots in Fig. 3 have been used. In one hand, the images have been only corrupted with impulse noise. The noise appearance probability is denoted by $p$ and is successively 0.05, 0.1 and 0.2. The parameters of each filter have been set to optimize the Peak Signal to Noise Ratio (PSNR). The PSNR and the normalized colour difference (NCD) have been used to assess the performance of the filters (Tables 2 - 4). Experimental results show that the proposed method exhibits almost the same performance as the FPG filter. On the Fig. 4, we can see that the images obtained using the FPG and DENS filters have a slightly more pronounced contrast than those obtained by the FROD filter, which explains best PSNR and NCD results. This experiment shows that our filter is as well robust as existing state-of-the-art filters when dealing with impulse noise.

In a second experiment, the same set of images is now corrupted with both impulse and Gaussian noise. We denote $\sigma$ the standard deviation of the Gaussian noise. Tables 5 - 7 show the experimental results. As expected, the FROD filter obtains the same worst


(a) (b)

(c) (d)

**Fig. 3.** Test images: (a) Caps (b) Flower (c) Motorbikes (d) Parrots

**Table 2.** Comparison of the performance measured in terms of PSNR and NCD ($\times 10^2$) using images corrupted with $p = 0.05$ impulse noise

| Image | | Noisy | VM | FROD | FPG | DENS |
|---|---|---|---|---|---|---|
| Caps | PSNR | 22.28 | 34.56 | 38.02 | 41.19 | 40.61 |
| | NCD | 5.83 | 2.08 | 0.56 | 1.49 | 0.27 |
| Flower | PSNR | 22.54 | 34.25 | 36.31 | 39.74 | 38.47 |
| | NCD | 5.84 | 2.51 | 0.79 | 1.78 | 0.40 |
| Motorbikes | PSNR | 21.56 | 26.88 | 29.46 | 32.85 | 31.96 |
| | NCD | 6.90 | 6.24 | 2.03 | 4.26 | 1.08 |
| Parrots | PSNR | 21.68 | 35.63 | 36.92 | 39.85 | 40.61 |
| | NCD | 5.74 | 1.58 | 0.48 | 1.55 | 0.21 |

results as the VM filter since it is not adapted to Gaussian noise. Our proposed filter slightly outperforms the FPG filter, in particular when the noise amount is high. This is mainly due to the smoothing ability which derives from the adaptation of the bilateral filter. On Fig. 5, the background of the DENS result image is smoother than the FPG result one whereas the details are preserved.

These experiments illustrate the ability of the proposed filter to adapt itself to impulse and Gaussian kind of noise.

**Table 3.** Comparison of the performance measured in terms of PSNR and NCD ($\times 10^2$) using images corrupted with $p = 0.1$ impulse noise

| Image | | Noisy | VM | FROD | FPG | DENS |
|---|---|---|---|---|---|---|
| Caps | PSNR | 19.31 | 33.82 | 36.08 | 38.38 | 37.61 |
| | NCD | 11.71 | 2.22 | 0.91 | 1.71 | 0.51 |
| Flower | PSNR | 19.52 | 33.12 | 33.74 | 37.35 | 36.16 |
| | NCD | 11.69 | 2.79 | 1.73 | 1.77 | 0.70 |
| Motorbikes | PSNR | 18.53 | 26.25 | 27.84 | 30.71 | 29.52 |
| | NCD | 13.77 | 6.67 | 3.17 | 4.45 | 1.95 |
| Parrots | PSNR | 18.69 | 34.47 | 35.78 | 37.76 | 37.68 |
| | NCD | 11.41 | 1.72 | 0.66 | 1.69 | 0.41 |

**Table 4.** Comparison of the performance measured in terms of PSNR and NCD ($\times 10^2$) using images corrupted with $p = 0.2$ impulse noise

| Image | | Noisy | VM | FROD | FPG | DENS |
|---|---|---|---|---|---|---|
| Caps | PSNR | 16.30 | 31.87 | 32.65 | 34.79 | 34.85 |
| | NCD | 23.53 | 2.61 | 1.76 | 2.07 | 2.3 |
| Flower | PSNR | 16.53 | 30.90 | 31.53 | 33.52 | 33.29 |
| | NCD | 23.46 | 3.44 | 2.42 | 2.90 | 3.57 |
| Motorbikes | PSNR | 15.54 | 24.83 | 25.47 | 27.70 | 27.10 |
| | NCD | 27.60 | 32.14 | 32.61 | 34.45 | 35.06 |
| Parrots | PSNR | 15.69 | 32.14 | 32.61 | 34.45 | 35.06 |
| | NCD | 22.79 | 2.13 | 1.52 | 1.96 | 2.00 |

**Table 5.** Comparison of the performance measured in terms of PSNR and NCD ($\times 10^2$) using images corrupted with $p = 0.05$ impulse and $\sigma = 5$ Gaussian noise

| Image | | Noisy | VM | FROD | FPG | DENS |
|---|---|---|---|---|---|---|
| Caps | PSNR | 22.02 | 32.97 | 32.97 | 36.95 | 36.86 |
| | NCD | 14.77 | 6.43 | 6.44 | 4.67 | 4.39 |
| Flower | PSNR | 22.28 | 32.59 | 32.60 | 36.20 | 36.02 |
| | NCD | 14.47 | 6.91 | 5.07 | 4.53 | |
| Motorbikes | PSNR | 21.35 | 26.45 | 28.17 | 31.41 | 30.85 |
| | NCD | 20.50 | 12.3 | 15.17 | 8.34 | 8.36 |
| Parrots | PSNR | 21.46 | 33.65 | 33.67 | 36.93 | 37.10 |
| | NCD | 13.55 | 5.47 | 5.47 | 4.14 | 3.75 |

(a) (b)

(c) (d)

**Fig. 4.** Filter outputs of the Flower image corrupted with $p = 0.1$ impulse noise: (a) VM filter (b) FROD filter (c) FPG filter (d) DENS filter

**Table 6.** Comparison of the performance measured in terms of PSNR and NCD ($\times 10^2$) using images corrupted with $p = 0.1$ impulse and $\sigma = 10$ Gaussian noise

| Image | | Noisy | VM | FROD | FPG | DENS |
|---|---|---|---|---|---|---|
| Caps | PSNR | 18.80 | 29.96 | 32.41 | 33.02 | |
| | NCD | 28.55 | 11.62 | 8.99 | 7.97 | |
| Flower | PSNR | 19.04 | 29.46 | 29.47 | 31.97 | 32.47 |
| | NCD | 28.56 | 12.13 | 9.01 | 8.14 | |
| Motorbikes | PSNR | 18.13 | 25.04 | 25.11 | 28.34 | 27.88 |
| | NCD | 39.45 | 19.68 | 26.97 | 15.34 | 13.37 |
| Parrots | PSNR | 18.28 | 30.21 | 30.24 | 32.49 | 33.36 |
| | NCD | 26.43 | 10.18 | 10.19 | 7.78 | 7.06 |

(a) (b)

(c) (d)

(e) (f)

**Fig. 5.** Filter outputs of the (a) Parrots image, (b) image corrupted with $p = 0.2$ impulse and $\sigma = 20$ Gaussian noise and outputs from (c) VM filter (d) FROD filter (e) FPG filter (f) DENS filter

**Table 7.** Comparison of the performance measured in terms of PSNR and NCD ($\times 10^2$) using images corrupted with $p = 0.2$ impulse and $\sigma = 20$ Gaussian noise

| Image | | Noisy | VM | FROD | FPG | DENS |
|---|---|---|---|---|---|---|
| Caps | PSNR | 15.41 | 24.98 | 24.98 | 26.56 | 27.84 |
| | NCD | 53.22 | 23.08 | 23.08 | 18.26 | 15.73 |
| Flower | PSNR | 15.59 | 24.70 | 24.70 | 26.30 | 27.47 |
| | NCD | 53.10 | 23.12 | 23.12 | 18.26 | 15.67 |
| Motorbikes | PSNR | 14.79 | 22.11 | 22.11 | 23.81 | 24.52 |
| | NCD | 72.19 | 35.11 | 35.12 | 27.71 | 24.38 |
| Parrots | PSNR | 14.93 | 24.91 | 24.91 | 26.51 | 27.93 |
| | NCD | 50.25 | 20.59 | 20.59 | 16.90 | 14.33 |

## 5 Conclusions

This paper adapts the classical fuzzy scheme for data analysis in the framework of noise reduction for multicomponent images. This scheme consists in a data fuzzification following by a defuzzification allowing the decision. The approach we propose is based on first the selection of adaptive fuzzy neighbourhoods of the pixels (i.e. the fuzzification) and second a defuzzification taking into account both spatial and photometric aspects of images. This fuzzy logic approach allows us to model the most classical filters used in the framework of image processing. Therefore this fuzzy scheme offers new angles for noise reduction of multicomponent images. The new density-based filter we propose reduces both high level noise (impulse noise) and low level noise (Gaussian noise). Like Bilateral filter, our filter reduces low level noise preserving details because its anisotropic nature. But it is also as robust against outliers (i.e. high level noise) as the vector median based filters are. Therefore the fuzzy scheme permits us to design a new filter taking into account the advantages of two classic filters for reducing both high and low level noise.

## References

1. Gonzales, R., Woods, R.: Digital Image Processing. Addison-Wesley, USA (1992)
2. Kotropoulos, C., Pitas, I.: Nonlinear model-based image/video processing and analysis. Wiley, New York (2001)
3. Bovik, A.: Handbook of image and video processing. Academic Press, San Diego (2000)
4. Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N.: Vector sigma filters for noise detection and removal in color images. Journal of Visual Communication and Image Representation 17, 1–26 (2006)
5. Lin, R., Hsueh, Y.: Multichannel filtering by gradient information. Signal Processing 80, 279–293 (2000)
6. Wong, W., Chung, A., Yu, S.: Trilateral filtering for biomedical images. IEEE Proceedings (2004)

7. Gallegos-Funes, F., Ponomaryov, V.: Real-time image filtering scheme based on robust estimators in presence of impulsive noise. Real-Time Imaging 10, 69–80 (2004)
8. Ville, D.V.D., Nachtegael, M., der Weken, D.V., Kerre, E., Philips, W., Lemahieu, I.: Noise reduction by fuzzy image filtering. IEEE Transaction on Fuzzy Systems 11, 429–436 (2003)
9. Morillas, S., Gregori, V., Hervás, A.: Fuzzy peer groups for reducing mixed gaussian-impulse noise from color images. IEEE Transactions on Image Processing 18, 1452–1466 (2009)
10. Camarena, J.G., Gregori, V., Morillas, S., Sapena, A.: Two-step fuzzy logic-based method for impulse noise detection in colour images. Pattern Recognition Letters 31, 1842–1849 (2010)
11. Leekwijck, W.V., Kerre, E.: Defuzzification: criteria and classification. Fuzzy Sets System 108, 159–178 (1999)
12. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of IEEE Conference on Computer Vision, Bombay, India (1998)
13. Detyniecki, M.: Mathematical aggregation operators and their application to video querying. Research Report, LIP6, Paris (2001)
14. Bouchon-Meunier, B.: La logique floue et ses applications. Addison-Wesley, Paris (1995)
15. Astola, J., Haavisto, P., Neuovo, Y.: Vector median filters. IEEE Proceedings 78, 678–689 (1990)
16. Herbin, M., Bonnet, N.: A new adaptive kernel density estimation. In: Information Processing and Management of Uncertainty (IPMU), Annecy (2002)

# Being Healthy in Fuzzy Logic: The Case of Italy*

Tindara Addabbo[1], Gisella Facchinetti[2], and Tommaso Pirotti[1]

[1] Dipartimento di Economia Politica, Facoltà di Economia,
Università di Modena e Reggio Emilia, Viale Berengario 51, Modena (MO), Italy
[2] Dipartimento di Scienze Economiche e Matematico Statistiche, Università del Salento,
Centro Ecotekne Pal. C - S.P. 6, Lecce-Monteroni, Lecce (LE), Italy
{tindara.addabbo,tommaso.pirotti}@unimore.it,
gisella.facchinetti@unisalento.it

**Abstract.** Health is a crucial dimension of individual well being and it is in itself multidimensional and complex. In this paper we tackle this complexity by using fuzzy expert logic that allows us to keep its complexity and at the same time to get crisp indicators that can be used to measure its status. The applied analysis refers to a country, Italy, that shows a high regional variability in health achievements related to the different diffusion and quality of health services across Italy. The source of data used for this purpose is the Italian National Statistical Institute (ISTAT) survey on health conditions. We proceed with a comparison of the results of the application of fuzzy logic to health measurement to a more standard methodology (SF-12) outlining the advantages of using fuzzy logic. The obtained fuzzy measure of health is then analyzed by means of multivariate analysis that confirms regional variability, lower health achievements for women, elderly and lower educated individuals. People in nonstandard working positions (like temporary contract) or unemployed show a lower health achievement too.

**Keywords:** Fuzzy logic, Health, Capabilities, Gender perspective.

## 1 Introduction

This paper presents the initial results of a wider research project supported by the Italian Ministry of Health on gender and health. It is made up of six projects, each dealing with different aspects from a gender perspective. One of these projects, that developed by the research unit of the University of Modena & Reggio Emilia, is concerned with the socio-economic determinants of health from a gender perspective. We thank the expert group on health (Sivana Borsari, Maria Cristina Florini and Erica Villa) for their comments on the construction of the model used to measure health; Anna Maccagnan for her elaborations of the microdata and the other participants in the project for their comments on a previous version of this paper. This idea is supported by the increasing attention given in recent years to gender differences and inequalities, which no longer

* This paper is part of the research activities of the University of Modena & Reggio Emilia research unit within the broader project funded by the Italian Ministry of Health: "La medicina di genere come obiettivo strategico per la sanità pubblica: l'appropriatezza della cura per la salute della donna".

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 197–212.
DOI: 10.1007/978-3-642-35638-4_14 

come down to mere biological factors also seen within a wider perspective that includes the concept of women's capability of living a healthy life. Nonetheless, in Italy we observe a systematic lack of appreciation of "gender-oriented health", fundamental to guaranteeing equity and planning efficient health and social services. In our group's project, the evaluation of the gender factor will refer to four dimensions: access to services, objective and subjective health, life styles and states of well-being. The classical definition of a country's welł-being is usually connected with GDP measurements. The need to take the health dimension into account in the evaluation of well-being in order to go beyond GDP and towards an extended measurement of human development has been widely recognized in the literature [1], [2] leading to the proposal of indicators that measure human development and explicitly include measures of the health dimension such as the Human Development Index [3]. Here we follow Sen's capability approach [4] by measuring well-being in its multidimensional setting devoting special attention to one dimension: the capability of living a healthy life. In defining this capability we are aware of its complexity stemming from various dimensions (physical vs mental health; subjective vs objective) and bound up in the social environment that affects its development. In order not to lose its complexity while measuring it, we adopt fuzzy logic. Fuzzy logic is ideal, in our opinion, since it allows us to get to the heart of the development process of the capability without losing the various dimensions that interact to define it. An attempt to exploit fuzzy logic to measure healthy living has previously been undertaken by Pirotti (2006) using Italian microdata (yet with a limited number of variables to define health) and by Addabbo et al. (2010a) to measure the capability of living a healthy life in the Modena district. However, this is the first attempt to implement a fuzzy inference system on the definition of living a healthy life with a large number of dimensions at national level in Italy. Due to the different methodology adopted, this work differs in methodological terms, from other previously published papers, dealing with the issue of health from a capability perspective [6]. The fuzzy technique in fact allows us to preserve the complexity of the measuring issue and, at the same time through a system of rules,to make explicit the relationships between the variables that help to assess the degree of capability development. The presence in the project of experts in health problems has helped us in fuzzy inference building, in the fuzzification of inputs and in the rule construction. But our purpose is also to compare our "non main stream" approach with a classical method to look at differences, faults and values. So we have looked at the SF12 questionnaire, which is an instrument adoptedto measure the "health level" widely used (in over 40 countries) and validated by the international scientific community. It has been in use since 1992, when the Medical Outcomes Study (MOS) developed a standardized survey form made up of 115 items synthesized in 12 steps. The MOS was a two year study of patients with chronic conditions that set up to measure the quality of life, including physical, mental and general health. The SF-12 requires only two minutes to be filled in, and it can be self-reported. It can be added to larger surveys or printed on a single sheet and easily faxed away. Due to its handiness, yet still being of great meaningfulness as stated before, during the last decade the use of SF-12 has spread throughout the world. Even the Italian National Institute of Statistics (ISTAT) decided to add an SF-12 section to its 2005 national health survey. So we carried out our analysis using the Italian Statistical National

Institute survey on health conditions in 2004-2005, which provides a set of variables well-suited to the information needs for the treatment of the topic in question. Particularly relevant for the purposes of this work is the information on the measurement of health-related elements of quality of life, such as obesity, certain diseases, disabilities, on specialist visits and visits to the ER. Moreover, the survey contains information on factors that may affect the capability of living a healthy life and/or its conversion in functionings. Amongst them, we may identify in the light of Sen's capability approach: *Socio/Institutional factors*. These refer to the presence of social services in the region where the individual lives. In a further extension, we will also include data on the health structure available. In this specification of the model, we do take into account these factors by including regional dummies (given the uneven presence of health services in the Italian regions). *The individual factors:*age, gender, educational level and employment conditions. We expect to find a negative correlation between age and health status due to the worsening of health conditions experienced by the elderly. As regards the level of education, it is now documented extensively in the international literature that higher education is usually associated to a better health. This is due to a greater awareness of the importance of lifestyles on health and also to improved access to health services [7]. Furthermore a higher education level allows for a wider choice about of jobs that individuals may take and access to posts characterized by healthier conditions as well as a higher income, which may improve access to health services. Individuals employment status may be considered a crucial individual conversion factor: some contractual arrangements, like temporary work contracts, given their high level of instability, may have a negative effect on health, mainly due to the stress induced by the uncertainty linked to the job security [8]. Individual health status, as experienced in literature, is also influenced by *familiar conversion factors* such as parents' education level, marital status, parents' level of health, family income and housing conditions. These factors can affect lifestyles, for example, or access to health services.The fuzzy approach we propose provides all the values of the knowledge-based systems. Everything is transparent; the rule blocks, which translate the weights proposed by the experts, are readable and always justified and may be changed if necessary. SF12 applied to the ISTAT 2005 national survey on health is not able to produce this effect as its results are based on a weighted average. Moreover, the weights used to compute the weighted average were evaluated in 1994 using data based on the 1992 MOS survey for the USA [9]; thus, one may question the validity of the same weights years later and in a different country. On the other hand, though affected by the reliability of experts and the need to use a more complex methodology, fuzzy logic with its tree structure of the inference system, allows us to understand the inputs that produce the final result and to improve the final outcome by devising policies in those areas that appear to be less developed.

## 2 The Use of a Fuzzy Inference System within a Capability Approach Framework

Fuzzy logic has been previously used to measure poverty and well-being by Cheli & Lemmi (1995) and, by following the capability approach, by Chiappero Martinetti (2000). However the method they follow is different from the one adopted in our contribution. In fact they use a mix of probability theory and fuzzy logic and data are used

to build variables distributions similar to aleatory distributions, while the aggregation functions are similar to weighted averages, explained on the basis of weights that are determined ex-ante. In this method the creation of the membership functions relies on the distribution of every single variable in the population of reference. In our contribution, we use fuzzy logic following more heuristic methods, which, in our opinion, are more effective and able to reflect the multidimensionality of the issue of measuring capabilities without depending on current data. The system is constructed by following experts' judgments and rules based on their experience and/or on the literature. The experts start by choosing the 'input' variables, they then propose their aggregation with 'intermediate' variables and then to an output variable. The latter is interpreted as the final evaluation of the development of the functionings of the capability under analysis. Experts are also responsible for identifying the membership functions of the initial variables; therefore, unlike the method followed by Chiappero Martinetti (2000) the latter do not depend on the current available data, but are set by the experts on the basis of their experience. Experts suggest how to aggregate input variables by using only linguistic rules and attributes without seing the data in advance. The experts' linguistic rules are translated formally by mathematicians. The proposed system of rules is then explicitly described 'rule by rule', allowing us to understand to what extent the results depend on the ratings determined by the experts. The method we apply here to measure of the capability of living a healthy life has already been used on an experimental basis for the measurement of well-being within the capability approach, [12,13,14,15] and specifically for the measurement of the capability of living a healthy life by Pirotti (2006) and by Addabbo, Chiarolanza, Fuscaldo and Pirotti (2010).

## 3 The Short Form 12 (SF-12)

This questionnaire is a set of 12 questions relating to the condition perceived over the four weeks prior to the interview, allowing us to compile two indexes: the Physical Component Summary (PCS) (index of physical health) and Mental Component Summary (MCS) (index of mental health), with values from 0 to 100. Because of its brevity and simplicity, it is widely used in more than 40 countries and has been validated by the international scientific community[17]. SF-12 is based upon a 12 questions tool of analysis that has its roots in the instruments used since 1992, when the Medical Outcomes Study (MOS) developed a standardized survey form consisting of 115 items synthesized in 12 steps. MOS was a two-year study of patients with chronic illnesses which aimed to measure the quality of life, including physical, mental and general health. As part of the MOS, RAND, (acronym of Research and Development) developed the 36-Item Short Form Health Survey (SF-36): a set of generic, coherent and easily administered quality-of-life indicators. These measurements rely upon patient's self-reporting; thus the administration of the survey is very handy, yet a wide range of literature has backed up the quality of the results assessed by this survey. Through the analysis of case studies collected during the MOS project, RAND selected eight groups of questions, or health concepts, from the original 40 [18]. Those chosen represent the most frequently measured concepts in widely-used health surveys and those most affected by disease and treatment [19]. The questionnaire items selected also represent multiple

operational indicators of health, including: behavioural function and dysfunction, distress and well-being, objective reports and subjective ratings, and both favourable and unfavourable self-evaluations of general health status [19]. This psychometric survey was first developed in the US and then developed internationally over the last 10 years. The SF-36 idea is based on a three-level tree scheme, starting from the single 36 items, aggregating them in eight scales and defining the summary measures of physical and mental health on the third level (respectively PCS and MCS). The discovery that SF-36 physical and mental component summary scales (referred to as PCS-36 and MCS-36 respectively) capture about 85% of the reliable variance in the eight-scale SF-36 health profile provided a new strategy for meeting this challenge. While two outcome measures are satisfactory for many purposes, a survey with fewer questionnaire items could be constructed to estimate these outcomes. Predictive studies supported this strategy. 12 SF-36 items and improved scoring algorithms reproduced at least 90% of the variance in PCS-36 and MCS-36 in both general and patient populations, and reproduced the profile of eight SF-36 health concepts sufficiently for large sample studies. The reproductions of PCS-36 and MCS-36 proved to be accurate enough to warrant the use of published norms for SF-36 summary measures in interpreting SF-12 summary measures. The SF-12 Survey represents an efficient synthesis of SF-36. Several empirical studies also conducted in European populations showed that the synthetic indices of the SF-12 correlated with the corresponding indices of the SF-36 with a range of values between 0.93 and 0.97 [17]. SF-12 requires only two minutes to be filled in, and it may be self-reported. It can be added to larger surveys or printed on a single sheet and easily faxed away. Due to its handiness, yet still being of great meaningfulness as stated before, over the last decade the use of SF-12 has spread all over the world. Even the Italian National Institute of Statistics (ISTAT) decided to add an SF-12 section to its 2005 national survey on health. We will use variables collected in the ISTAT Survey by using SF-12s to construct our fuzzy inference system (FIS) on the capability of living a healthy life and compare the results obtained through FIS to the original SF-12 outputs.

## 4 A Fuzzy Inference System to Measure the Health of the Italian Population

A fuzzy inference system (FIS) (Figure 1) may be graphically represented as a tree. Starting from the right hand side we see the output of the system: the health status. Moving to the left, the tree grows and presents various nodes, representing the intermediate variables describing the macro-indicators, through to the smallest branches which show the initial inputs. The basic input variables that appear on the left side of the tree conceptually pertain to three different areas: the first, concerning perceived physical and mental health; the second, whicj attains to more objective indicators of physical health, and the third, which regards access to health services. Lifestyles were not taken into account because they represent risk factors in the medium and long term but they are not “manifestations” of the immediate state of health of individuals. What we aim to do here is instead to understand the health status over a relatively short period, such as the last four weeks. Instead of directly using SF-12 outcomes, available as a ready to use variable, we decided to build a “fuzzy SF-12”, the results of which (Physical

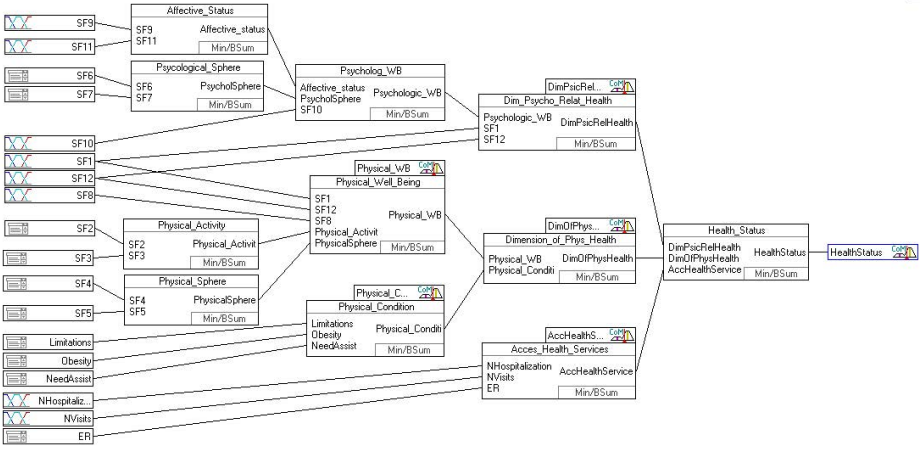



**Fig. 1.** The chosen fuzzy inference system tree

Well-being and Dimension of Psychological and Relational Health) could be used as intermediate variables for the final "status of health index" and at the same time, be compared to the original Physical Component Summary (PCS) SF-12 index of physical health and the Mental Component Summary (MCS) SF-12 index of mental health. The reason for this choice is that, as already stated, even though the SF-36 idea relies on a tree scheme basis, SF-12 outcomes are obtained as a reduction of the variables based on a statistical basis that makes impossible to reconnect the final SF-12 analysis scheme to the original logic that guided the researchers in first place. According to the SF-12 operative manual, MCS and PCS are built through the use of weighted means, using regressive coefficients coming from analysis based on the American population. However, the coefficients are given and derived from SF-36 coefficients that, in turn, come from the original 115 questions of the MOS survey; therefore, it is hard to trace back the path that led to the construction of the coefficients. Moreover, we believe that the assumption that the coefficients estimated using a sample representative of the American population in the mid '90s remains valid even when applied to the analysis of other countries and almost ten years later is rather strained. There is no guarantee of the validity of results. As a last consideration, we must notice that MCS and PCS are two indicators that have been designed to be two well separated indexes, not to be bundled toghether in a single synthetic health indicator. Instead in our opinion, supported by our health experts' opinions and by the literature, it is possible to proceed with the construction of a third synthetic index that takes into account elements of both dimensions. For this reason, even if we assume that SF-12 results are proven to be reliable, we wanted to produce indexes whose results could arise from immediately understandable choices and that could also produce a unified health index. Our "fuzzy SF-12" is hence an expert system, driven by experts' judgments, so that the survey outcomes are the direct reflection of a precise will, connected to the analysis of the specific Italian framework. Moreover in our evaluation system, following Wagstaff et alt. (1991) we have decided to propose not just the PCS and MCS scheme, but three macro-indicators (physical and mental health, physical condition and access to the health services). The "health" of the

**Table 1.** The system variables: abbreviations and relative questions

| | |
|---|---|
| ER | Did you need assistance from the ER during the last 12 months? |
| Limitations | Did you experience limitations during at least the last six months? |
| NeedAssist | Do you think that you need house care assistance? |
| NHospitalizations | Number of Hospitalizations in the last 3 months |
| NVisits | Number of visits during the last 4 weeks |
| Obesity | Are you obese? |
| SF1 | In general, would you say your health is: excellent, very good, good, fair or poor? |
| SF10 | How much of the time during the past 4 weeks did you have a lot of energy? |
| SF11 | How much of the time during the past 4 weeks have you felt downhearted and blue? |
| SF12 | During the past 4 weeks, how much of the time has your physical health or emotional problems interfered with your social activities (like visiting with friends, relatives, etc.)? |
| SF2 | Does your health now limit you in moderate activities, such as moving a table, pushing a vacuum cleaner, bowling, or playing golf |
| SF3 | Does your health now limit you in climbing several flights of stairs? |
| SF4 | During the past 4 weeks (relatively to your work or other regular daily activities as a result of any physical problems) did you accomplished less than you would like? |
| SF5 | During the past 4 weeks, as a result of any physical problems, were you limited in the kind of work or other regular daily activities? |
| SF6 | During the past 4 weeks (relatively to your work or other regular daily activities as a result of any emotional problems such as feeling depressed or anxious) did you accomplished less than you would like? |
| SF7 | During the past 4 weeks, as a result of any emotional problems such as feeling depressed or anxious, didn't you do work or other regular daily activities as carefully as usual? |
| SF8 | During the past 4 weeks, how much did pain interfere with your normal work (including both paid work and housework)? |
| SF9 | How much of the time during the past 4 weeks have you felt calm and peaceful? |
| AccHealthService | Access to the Health Services |
| DimOfPhysHealth | Dimension of Physical Health |
| DimPsicRelHealth | Dimension of Psycho-Relational Health |
| HealthStatus | Health Status |
| Physical_Conditi | Physical Condition |
| Physical_WB | Physical Well-being |
| Affective_status | Affective Status |
| Physical_Activit | Physical Activity |
| PhysicalSphere | Physical Sphere |
| PsycholSphere | Psychological Sphere |

fuzzy system's final output (Health Status) investigated from a physical point of view (physical health dimension) and a mental or psychological point of view (mental health dimension) use the items in the SF-12 survey; however, it is not just the result of the use

of these items: firstly we have a third dimension, bound up in the actual use of services and structures connected to the healthcare service. Thus in our vision there are not just two dimensions but three. We have noticed that SF-12 items are far too connected to a subjective evaluation of health. This third leg of the tree helps to connect subjective to objective information. In addition to this the physical health dimension is not just the result of the elaboration of the SF-12 items, but, for the same reason, we have added physical objective data. Looking at the PCS items, it becomes clear that the items attain to "Physical Well-being". For a comprehensive evaluation of health, its perception represents an important reference as it helps to capture the multidimensionality of the concept itself, defined according to the World Health Organization as a state of "complete physical, mental and social wellbeing" [21]. Adding information about people's physical conditions greatly helps to better evaluate the dimension of physical health. In this way functional indicators define health in relation to the loss of skills in performing 'normal' daily activities. Medical indicators identify the presence of specific diseases or disabling conditions diagnosed by physicians. Subjective ones,on the other hand, define health according to the perception of the individual. In a fuzzy system, the same variable can be used several times. The complexity of relationships between different determinants of individual health is indicated by the presence of some input variables, in keeping with the literature in more than one dimension of the state of individual health. The "access to health care services" dimension comprises information (or basic variables) as the number of hospitalizations (excluding childbirth hospitalizations), over the past three months, the number of accesses to the Emergency Room (ER) over the past 12 months, not counting the so called white codes, meaning wrong or unnecessary accesses to the ER, and the number of doctors's visits, excluding dental visits. As may be easily understood, the effect of these variables (and of the intermediate index) on the final variable, "individual health status" is negative because a high number of accesses to health services is likely to be connected to a poor health status. In order to fuzzify the inputs, the experts have decided to identify three linguistic membership functions per each variable, respectively named "none", "some" and "many". These are applied to "number of visits", for which 0 is connected to the spike of none, 2 to the spike of some and 4 to the spike of many.

The same membership functions (MBFs) were applied to the number of hospitalizations, so that 0 is associated to none, some to 3 and many to 5. The access to the ER instead is just a dummy and it tells us whether an individual had to ask for assistance over the previous 12 months.

The aggregation method amongst fuzzified variables is not an explicit function, but it is expressed in the form of the explicit rule block, where every possible interaction between the fuzzy sets (for instance none, some and many( is represented by a block line in the "IF" part, while the effect on the variable on which they insist is represented by a synthetic lexical effect in the "THEN" part. Since more than one rule may be activated at the same time, every rule is activated with the MIN aggregating rule, which stands for the minimum level of activation between the sets (always between 0 and 1), acting in the "IF" part. If a term is activated with a level of 0, it means that it is absolutely not activated (the data do not belong to that fuzzy set). On the other hand if the level is 1 it means that the term is fully activated, meaning that the data belongs

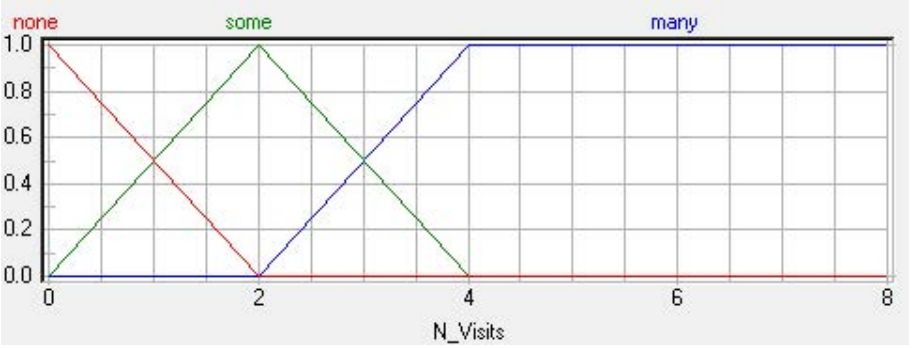



**Fig. 2.** Fuzzyfication scheme and membership functions of N_Visits

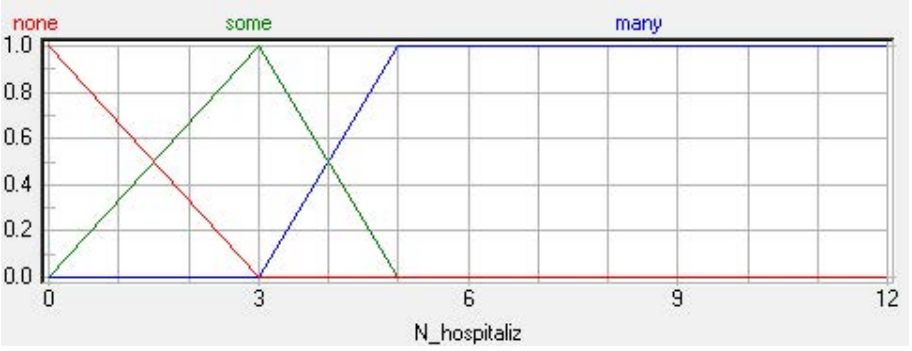



**Fig. 3.** Fuzzyfication scheme and membership functions of N_Hospitaliz

entirely to that specific term and just to that one. Every number in between stands for a partial belonging between different fuzzy sets. The way the membership degree to a particular fuzzy set is decided depends on the specific membership function of every fuzzy set. On the THEN side, there may be many lines that lead to the same lexical effect. If there are more activated lines in the same rule block with the same effect, the chosen aggregation rule is the bounded sum (BSUM): all the effect activation levels get summed up tol the level of 1. Any effect added to that level produces no result. The described aggregating process, through the use of rule blocks, is iterated from the left to the right of the system tree. At the end of the process, to make the results intelligible to human beings it is necessary to de-fuzzify them. This is done with a system called "Center of Maximum" or, in short, CoM: if more effect are active at the same time in the final rule block, only the highest will be considered and the result will be equivalent to the peak of its membership function.

The variable named "Dimension of Physical Health" was designed to be an aggregation between the "Physical condition" and the "Physical Well-being". The "Physical condition" identifies health conditions caused by chronic or incapacitating diseases through objective indicators such as the presence of limitations for at least six months,

**Table 2.** A rule-block example:the access to health services

| If | | | Then |
|---|---|---|---|
| **Hospitalizations** | **Visits** | **E.R.** | **Health Service** |
| none | none | No | very low |
| none | none | Yes | low |
| none | some | No | low |
| none | some | Yes | low |
| none | many | No | low |
| some | none | No | low |
| none | many | Yes | medium |
| some | none | Yes | medium |
| some | some | No | medium |
| some | some | Yes | medium |
| some | many | No | medium |
| many | none | No | medium |
| some | many | Yes | high |
| many | none | Yes | high |
| many | some | No | high |
| many | some | Yes | high |
| many | many | No | high |
| many | many | Yes | very high |

need for home care, and finally the presence of obesity, discriminated by body mass index values over 30 among over 18s, while in the population below 18 years of age, the corrections suggested in the literature were adopted [22]. While the first of the three basic indicators may be seen as a categorical variable on three levels, the other two are dummy variables. "Physical well-being", contributing to the definition of the "Dimension of physical health", uses some of the 12 items that make up the SF-12 questionnaire. The reason was to identify the most significant scales underlying the conceptual model, which lead to the creation of the PCS of the SF-12 survey [23,17]. Therefore, in detail the input variables are conceptually related to the following scales: general health, bodily pain, physical functioning play their role into the definition of the "Physical well-being" intermediate factor. This intermediate variable therefore contains the subjective evaluation of individual general health conditions, given by the interviewee, his perception of physical limitation due to pain at work and during usual social activities with family. The other intermediate variable taken into account is the "Dimension of Psychological and Relational Health", whose purpose is to evaluate individual health from a psychological-well being point of view. The dimension of Psychological and Relational Health is deliberately made up of many variables relying on the scales that are the main components of PCS in the SF-12 analysis: vitality, social functioning, emotional role and mental health. The aggregating process described, through the use of rule blocks, is iterated from the left to the right of the system tree. At the end of the process, to make the results intelligible to human beings it is necessary to de-fuzzify the results. This is done by using the Center of Maximum method described above.

**Table 3.** A comparison between the two Psychological Health indicators: DPRH and MCS

| Age Classes | DPRH | | MCS | |
|---|---|---|---|---|
| | Men | Women | Men | Women |
| 15-24 | 88.64 | 84.34 | 53.70 | 51.36 |
| 25-34 | 85.56 | 81.05 | 52.58 | 50.52 |
| 35-44 | 82.14 | 78.34 | 51.59 | 49.88 |
| 45-54 | 78.50 | 72.89 | 50.81 | 48.60 |
| 55-64 | 74.21 | 67.73 | 50.43 | 48.03 |
| 65-74 | 68.20 | 59.99 | 49.67 | 46.43 |
| 75+ | 56.21 | 47.84 | 47.16 | 44.09 |

## 5 The Individual Health Status in Italy and the Role of Observable Conversion Factors

In this paragraph we will analyze the results of the FIS applied to health, trying to place some personal or social factors in relationship with the development of the capability of living a healthy life. The degree of development will be approximated using the final output of the FIS presented in the previous paragraph. The sample for this analysis, as already stated, comes from the Italian National Institute of Statistics survey on health for 2004-2005. In particular, the object of our investigation is the subset of people over 14 who did not present any missing values on the variables chosen to run the FIS. In fact, one of the main prerequisites of an FIS is that the data matrix has to be dense. Since our dataset contains a relatively high number of observations, this prerequisite can be easily satisfied: the final sample is made up of 111,151 individuals, weighted to be significant both at a national and at a regional level. In Table 3 we compare the results on the measurement of the two Phsycological Health indicators: the fuzzy DPRH and the SF-12 MCS, while in Table 4 we compare the results obtained for the two Physical Health indicators the fuzzy PWB and the SF-12 PCS. Table 5 contains the results of the Fuzzy final output value on Health by gender and age.

Standardizing both the outputs of the evaluation system on a 0-100 range, we discovered that the fuzzy indexes are generally higher than MCS and PCS with respect to all the age classes, for both genders, but it is also pretty clear that the variability of the fuzzy indexes is much higher; hence the fuzzy outcomes are more sensitive to the changes caused by age. Furthermore, even though the results are generally higher, the trends are the same: women's health is worse than men's at every age, with a strong and constant decrease over time.

This result is also confirmed by the trend in the main index (Health Status), which is higher, on average, among the youngest individuals, a little better for men than for women, decreasing with age. All the indexes obtained and analyzed present a similar trend.

If we consider people's health status and we compare it now with their employment status, we see that the results are fairly consistent with what we might expect: students and people seeking their first job are expected to be younger and they actually receive

**Table 4.** A comparison between the two Physical Health indicators: PWB and PCS

| Age Classes | PWB | | PCS | |
|---|---|---|---|---|
| | Men | Women | Men | Women |
| 15-24 | 92.35 | 90.60 | 55.28 | 55.31 |
| 25-34 | 90.45 | 87.71 | 54.52 | 53.95 |
| 35-44 | 88.19 | 85.98 | 53.59 | 53.12 |
| 45-54 | 85.49 | 80.93 | 52.43 | 51.13 |
| 55-64 | 80.99 | 74.22 | 50.38 | 48.27 |
| 65-74 | 72.97 | 63.55 | 46.96 | 44.10 |
| 75+ | 55.46 | 44.23 | 40.39 | 36.87 |

**Table 5.** The average Health Status index by gender and age class

| Age Classes | Men | Women |
|---|---|---|
| 15-24 | 87.19 | 85.13 |
| 25-34 | 88.46 | 85.87 |
| 35-44 | 85.91 | 84.02 |
| 45-54 | 83.20 | 80.02 |
| 55-64 | 79.83 | 75.79 |
| 65-74 | 74.87 | 69.48 |
| 75+ | 65.50 | 59.61 |

the highest marks. On the other hand, we find people who are retired from work whose health status is worse given their average higher age.

But if we consider employed and unemployed people (Figure 4) we see that these two groups, which apparently should not differ so much as regards their average age, present quite different marks: 85.95 for the employed males against 81.62 for the unemployed and 83.48 for the employed women, compared to 81.48 of the unemployed women. This is in line with the health costs linked to unemployment status as outlined in Sen (1997). Turning to education (Figure 5), the data confirm what the literature claims as common ground: a higher educational level is positively related to individual health. We then completed our analysis by estimating a multivariate OLS regression model that allows us to take into account the weight of the different conversion factors on the index of living a healthy life resulting from the implementation of our FIS model (Table 6) to the data.The results obtained confirm a negative effect of ageing on the fuzzy measure of health and, having controlled for age, one can see that women are still characterized by worse health than men especially with regards to the psychic dimension. Health improves when the education level is higher. Turning to employment conditions, we can see that controlling for age and education levels, if one holds a temporary work position, his/her health status deteriorates (the control variable being employed on a permanent basis). Joblessness is also, consistent with Sen's analysis (1997), leading to lower health. Joblessness or temporary work contract have a higher negative effect on mental health. Whereas retired or disabled show a lower achievement in physical health. We control also for the type of disease with the higher negative

**Table 6.** Health Status: a multivariate analyses (standard errors in parenthesis)

| Variables | Health | Physical | Mental |
|---|---|---|---|
| Female | -0.0247*** (0.00140) | -0.0277*** (0.00222) | -0.0487*** (0.00264) |
| Age | -0.0429*** (0.00186) | -0.0301*** (0.00294) | -0.110*** (0.00350) |
| High School | 0.0365*** (0.00146) | 0.0625*** (0.00231) | 0.0394*** (0.00276) |
| Degree and over | 0.0478*** (0.00230) | 0.0726*** (0.00362) | 0.0631*** (0.00432) |
| Temporary | -0.00683** (0.00319) | -0.000697 (0.00503) | -0.0239*** (0.00600) |
| Retired | -0.0289*** (0.00209) | -0.0487*** (0.00332) | -0.0129*** (0.00394) |
| Disable | -0.521*** (0.00547) | -0.934*** (0.0101) | -0.655*** (0.0111) |
| Other condition | -0.0265*** (0.00174) | -0.0455*** (0.00275) | -0.0335*** (0.00328) |
| Unemployed | -0.0157*** (0.00291) | -0.00992** (0.00459) | -0.00539*** (0.00548) |
| South | -0.0165*** (0.00135) | -0.0317*** (0.00214) | -0.0202*** (0.00255) |
| Respiratory diseases | -0.0550*** (0.00181) | -0.0637*** (0.00289) | -0.0829*** (0.00342) |
| Diabetes | -0.0952*** (0.00296) | -0.137*** (0.00480) | -0.147*** (0.00565) |
| Cataract | -0.121*** (0.00369) | -0.206*** (0.00611) | -0.159*** (0.00708) |
| Hypertension | -0.0836*** (0.00182) | -0.112*** (0.00290) | -0.112*** (0.00344) |
| Bones diseases | -0.139*** (0.00187) | -0.212*** (0.00297) | -0.197*** (0.00354) |
| Cancer | -0.116*** (0.00404) | -0.102*** (0.00660) | -0.182*** (0.00778) |
| Ulcer | -0.0488*** (0.00394) | -0.0258*** (0.00637) | -0.0962*** (0.00752) |
| Gall/Kidney stones | -0.0504*** (0.00406) | -0.0607*** (0.00655) | -0.0782*** (0.00773) |
| Cirrhosis | -0.134*** (0.0117) | -0.131*** (0.0193) | -0.206*** (0.0230) |
| Migraine | -0.0319*** (0.00207) | -0.0164*** (0.00331) | -0.0719*** (0.00393) |
| Depression | -0.161*** (0.00240) | -0.117*** (0.00387) | -0.416*** (0.00457) |
| Alzheimer/ Parkinson's disease | -0.445*** (0.00711) | -0.643*** (0.0139) | -0.639*** (0.0147) |
| Neural system's diseases | -0.149*** (0.00541) | -0.131*** (0.00901) | -0.305*** (0.0105) |
| Thyroid diseases | -0.00967*** (0.00297) | 0.00378 (0.00474) | -0.0235*** (0.00563) |
| Skin's diseases | -0.0346*** (0.00570) | -0.0177* (0.00910) | -0.0737*** (0.0108) |
| Other pathologies | -0.0241*** (0.00620) | -0.0302*** (0.00986) | -0.0300** (0.0117) |
| Constant | 4.632*** (0.00684) | 4.633*** (0.0108) | 4.839*** (0.0129) |
| Observations | 111,151 | 109,43 | 110,306 |
| R-squared | 0.31875 | 0.23611 | 0,258333333 |

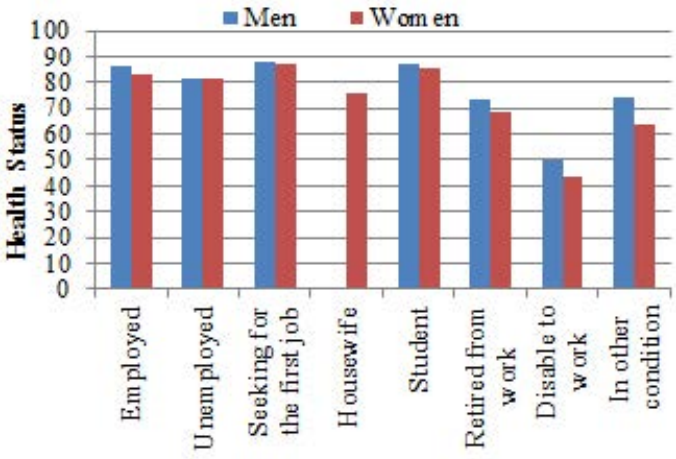



**Fig. 4.** The average Health Status by gender and employment status

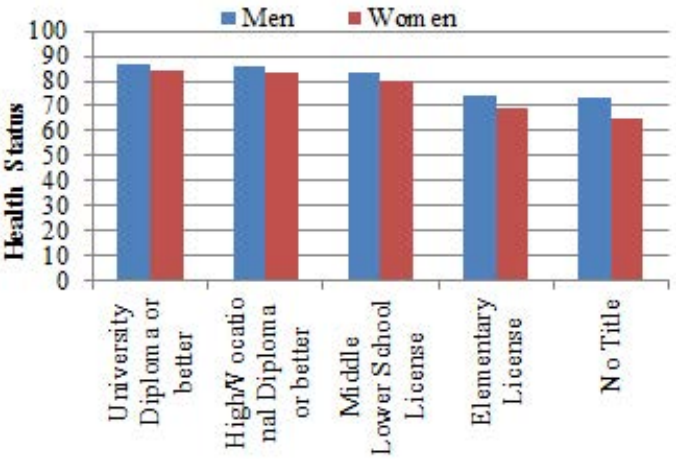



**Fig. 5.** The average Health Status by gender and educational level

effect on health status being connected to Alzheimer or Parkinsons disease followed by nervous disease and depression. Those living in the South of Italy show a lower level of health achievement, and this is probably connected to worse health infrastructures in the South of Italy. Deeper analyses on regional variability will be performed in further research by matching our population data with health infrastructures administrative data. A first attempt in this direction shows that the coverage of pap test by region on women aged over 25 has a positive effect on women's health and that people living in regions with higher percentage of places for elderly in public services or in household assisted public services show a higher health status.

## 6 Conclusions

In the analysis of individual well-being, health status is a central dimension. In this paper we have analyzed the individual health status by considering its multidimensional nature. In order not to lose its complexity we have proposed a modular approach (the fuzzy tree diagram) which allows us to obtain an index on health without losing single macro-index information. The choice, interaction and the effects of the various available indicators were chosen by the authors on the basis of health experts' opinions, expressed through linguistic rules. This methodology reduces the debated problem of the numerical attribution of weights. The health status (the final output of our fuzzy inference system) is determined by the interaction in the FIS of access to health services, the dimension of mental health and that of physical health. The first innovative product is thus precisely the use of a fuzzy inference system on the health status since it shows the individual settlement through the combination of the observable variables in the survey on the health status of the Italian population. We then analyzed the crisp value produced in relation to individual and family variables which may interact with the very fundation of a healthy condition. During the construction of the intermediate variables and of the whole system, the method that we applied maintains the complexity of the definition of health status, while at the same time, is able to produce a synthetic and numeric index. On average, the health status index of the Italian population is found to be lower for women than for men and for people holding unstable working positions, without work or living in the South. Further developments of the multivariate analysis of the health status include the effect of health services. Preliminary results show that the coverage of pap test by region on women aged over 25 has a positive effect on women's health and that people living in regions with higher percentage of places for elderly in public services or in household assisted public services show a higher health status.

## References

1. Fleurbaey, M.: Beyond gdp: the quest for a measure of social welfare. Journal of Economic Literature 47(4), 1029–1075 (2009)
2. Stiglitz, J.E., Sen, A.K., Fitoussi, J.P., et al.: Report by the commission on the measurement of economic performance and social progress. In: Commission on the Measurement of Economic Performance and Social Progress, Paris (2009)
3. United Nations Development Programmes: Human development report 1990: Concept and measurement of human development. In: UNDP. Oxford University Press, New York (1990)
4. Sen, A.K.: Capability and well-being. In: Nussbaum, M., Sen, A.K. (eds.) The Quality of Life. Clarendon Press, Oxford (1993)
5. Pirotti, T.: Le dimensioni del benessere: la salute. misure sfocate nell approccio delle capacità. Master's thesis, Università di Modena e Reggio Emilia (2006)
6. Kuklys, W.: Measurement and determinants of welfare achievement-evidence. In: 3rd Conference on the Capability Approach, Pavia (2003)
7. Mackenbach, J.P., Bakker, M.J., et al.: Tackling socioeconomic inequalities in health: analysis of European experiences. The Lancet 362(9393), 1409–1414 (2003)
8. Addabbo, T., Favaro, D.: Part-time and temporary employment in a gender perspective. In: Addabbo, T., Solinas, G. (eds.) Non Standard Employment and Quality of Work. The Case of Italy, ch. 3. Springer (2011) (forthcoming)

9. Ware Jr., J.E., Kosinski, M., Keller, S.D.: Sf-12: How to score the sf-12 physical and mental health summary scales (1998)
10. Cheli, B., Lemmi, A.: A "totally" fuzzy and relative approach to the multidimensional analysis of poverty. Economic Notes, 115–134 (1995)
11. Chiappero Martinetti, E.: A multidimensional assessment of well-being based on Sens functioning approach. Rivista Internazionale di Scienze Sociali 2, 207–239 (2000)
12. Addabbo, T., Di Tommaso, M.L., Facchinetti, G.: To what extent fuzzy set theory and structural equation modelling can measure functionings? An application to child well-being. Materiali di Discussione del Dipartimento di Economia Politica (468) (2004)
13. Addabbo, T., Facchinetti, G., Mastroleo, G.: Capability and functionings. a fuzzy way to measure interaction between father and child. In: Khalid, S., Pejas, J., Romuald, M. (eds.) Biometrics, Computer Security Systems and Artificial Intelligence Applications, pp. 185–195. Springer Science (2006)
14. Addabbo, T., Di Tommaso, M.L.: Children's capabilities and family characteristics in Italy. Materiali di Discussione del Dipartimento di Economia Politica (590) (2008)
15. Addabbo, T., Facchinetti, G., Maccagnan, A., Mastroleo, G., Pirotti, T.: A fuzzy system to evaluate how parents interact with their children. Metody Informatyki Stosowanej 3 (2010)
16. Addabbo, T., Chiarolanza, A., Fuscaldo, M., Pirotti, T.: Una valutazione estesa del benessere secondo lapproccio delle capacità: vivere una vita sana. In: Baldini, M., Bosi, P., Silvestri, P. (eds.) Le città Incartate, Mutamenti Nel Modello Emiliano Alle Soglie Della Crisi, Il Mulino, Bologna, ch. 7 (2010)
17. Gandek, B., Ware Jr., J.E., Aaronson, N.K., Apolone, G., Bjorner, J.B., Brazier, J.E., Bullinger, M., Kaasa, S., Leplege, A., Prieto, L., et al.: Cross-validation of item selection and scoring for the sf-12 health survey in nine countries: Results from the iqola project. Journal of Clinical Epidemiology 51(11), 1171–1178 (1998)
18. Ware Jr., J.E., Kosinski, M., Keller, S.D.: A 12-item short-form health survey: construction of scales and preliminary tests of reliability and validity. Medical Care 34(3), 220 (1996)
19. Ware Jr., J.E., Snow, K.K., Kosinski, M., Gandek, B.: SF-36 health survey: manual and interpretation guide. The Health Institute, New England Medical Center (1993)
20. Wagstaff, A., Paci, P., Van Doorslaer, E.: On the measurement of inequalities in health. Social Science & Medicine 33(5), 545–557 (1991)
21. Di Martino, M.: Le condizioni di salute della popolazione: Un'analisi multilivello delle disuguaglianze sociali. In: XXIX Conferenza Italiana di Scienze Regionali, Bari (2008)
22. Cole, T.J., Bellizzi, M.C., Flegal, K.M., Dietz, W.H.: Establishing a standard definition for child overweight and obesity worldwide: international survey. British Medicine Journal 320(7244), 1240–1243 (2000)
23. Apolone, G., Mosconi, P., Ware, J.E.: Questionario sullo stato di salute SF12. versione italiana, Milano, Istituto di Ricerche Farmacologiche Mario Negri (2005)
24. Sen, A.K.: Inequality, unemployment and contemporary europe. Int'l Lab. Rev. 136, 155 (1997)

# Fuzzy Median and Min-Max Centers: An Spatiotemporal Solution of Optimal Location Problems with Bidimensional Trapezoidal Fuzzy Numbers*

Julio Rojas-Mora[1,**], Didier Josselin[1], and Marc Ciligot-Travain[2]

[1] UMR Espace 7300 CNRS, University of Avignon (UAPV), Avignon, France
[2] LANGL, University of Avignon (UAPV), Avignon, France
{julio.rojas,didier.josselin,marc.ciligot}@univ-avignon.fr

**Abstract.** The calculation of the center of a set of points in an open space, subject to a given metric, has been a widely explored topic in operations research. In this paper, we present the extension of two of these centers, the median and the min-max centers, when there is uncertainty in the location of the points. These points, modeled by two-dimensional trapezoidal fuzzy numbers (TrFN), induce uncertainties in the distance between them and the center, causing that the resulting center may also be a two-dimensional TrFN. The solution gives flexibility to planners, as the value of the membership function at any given coordinate can be seen as a degree of "appropriateness" of the final location of the center. We further consider how to model the existing space constraints and what is their effect on the calculated centers. Finally, in the case of temporal analysis, we can determine the durability of the location of the center at a given point of the study area.

## 1 Introduction

Finding an optimal center in space became a common process in planning, because it allows to affect a set of demands to one or several locations that offer dedicated facilities. For instance, a center collecting wastes, a vehicle depot for logistic purpose or a hospital complex, all require a relevant metric to minimize cost or maximize access to them. Mathematicians, economists and geographers developed methods which locate these centers according to either equity (minimax) or efficiency (minisum) objectives, following the work in $k$-facilities location problems on networks [12], that respectively correspond to the $k$-median and the $k$-center. Indeed, there exist many mathematical problems and formalisms for optimal location problems [13]. More recently, we can see a larger scope of the domain and sets where these issues appear [3]. Other books complete the state-of-the-art [9,11,17] or focus on applications in transportation [15,18] or health care [1].

Methodologies for optimal location can be applied on continuous space, finite space or networks (graphs or roads for instance). If $k = 1$, then the aim is to find a single center. The choice of the metric $p$ is also significant because it involves, on the one hand,

* This work was funded by the French ANR ROLSES Project.
** Corresponding author.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 213–240.
DOI: 10.1007/978-3-642-35638-4_15 

the method to set the distance separating the demands to the center, and on the other, how to combine these distances according to a given objective function. Thus, there exist many ways to calculate a center for many points of demand, even when reducing complexity by considering a continuous space, a unique center and the Minkowski distance of $L_p$ norms. The first parameter, $p$, defines the norm of the distance separating the demand points to the center: rectilinear ($p = 1$), Euclidean ($p = 2$) or Chebyshev's ($p \to \infty$). The second parameter, $p'$, relates to the calculation of the center itself. The sum of the distances is minimized when $p' = 1$, the sum of the squared distances when $p' = 2$ and the maximum of the distances when ($p' \to \infty$).

Among all the possibilities crossing $p$ and $p'$ of the $L_p$ norms, only three cases can be computed in closed form: the median center, which minimizes the sum of the rectilinear distances ($p = p' = 1$), the centroid or barycenter, which minimizes the sum of the squared Euclidean distances ($p = p' = 2$) and the min-max center, which minimizes the maximum of the maximum marginal distances ($p \to \infty$ and $p' \to \infty$).

Scientists and planners use to consider the final location to be accurate and crisp, or, at least, as a finite set of possible predefined locations. However, there might be uncertainty on the estimated distances, due to uncertainty carried by the demand location itself. This is particularly true when considering urban sprawl, as it can generate non negligible variations on the location of the town's center, which in place might affect the location of the optimal center. There is also the case when subjective or vague information is used to define the demand location. The result, then, cannot possibly be a crisp point, and solutions that assume crisp data when non is available, might be at risk being far from optimal.

By modeling the demand points as bi-dimensional fuzzy sets we prove in this paper that the results obtained for crisp environments can be easily extended to the fuzzy ones, attaining homologous closed form expressions. As the solutions depend only on arithmetic operations of fuzzy numbers, thus obtaining fuzzy numbers as its coordinates, the approach followed in this work deviates from the path trailed by many fuzzy location papers, in which constraints are fuzzy, but the solution is not [8,2,4,16].

Fuzzy solutions also give some leeway to planners which might be forced to select the final location of the center away from the place with the highest membership value, but that can the measure the impact of their decision and, thus, asses its "appropriateness".

The space in which the points are contained may present some constraints that can be modeled as fuzzy sets. We propose a method, based on the intersection of fuzzy sets, to incorporate these constraints into the initial solution, obtaining a constrained fuzzy center. The intersection of a fuzzy center with these constraints may alter the membership function of the former, and thus, the center would cease to be a bidimensional TrFN to become a fuzzy subset.

This same methodology can be used to study the durability of a proposed location, by the intersection of the centers calculated at different instants. The membership function of the resulting fuzzy set can be seen as the degree of durability of the location. We also study the situation when the intersection is an empty set, proposing the gradual shedding of the farthest solutions until the intersection not only ceases to be empty,

but also has an objective function value greater than a given threshold provided by the decision maker.

This paper is structured in the following way. In Section 2, we introduce the closed form expressions for centers usually used in the literature. Then, on Section 3, the basic concepts of fuzzy sets and fuzzy numbers used through our paper are defined. Section 4 covers the demonstrations used to prove that the closed form expressions found for some centers in crisp environments can be extended to fuzzy points. In Section 5 we present our methodology based on the intersection of fuzzy sets to include space constraints and evaluate the durability of a given location. A small numerical example, joined by some figures in which the results can be easily seen, is developed in Section 6. Finally, Section 7 presents the conclusions as well as the future work based on our results.

## 2 The Median Center and the Min-Max Center

A recurrent problem in geography and planning is the need to find the center of a set of demand points that minimizes a given objective function. Without taking into consideration the road network that links these points, i.e., in an open space, there are two simple, but also widely used methods to solve this problem, the median center and the min-max center.

**Definition 1.** *For a set $P = \{p^{(i)}\}$ of $n$ points in $\mathbb{R}^2$, i.e., $p^{(i)} = \{p^{(i,x)}, p^{(i,y)}\}$, the median center $m = \{m^{(x)}, m^{(y)}\}$ is found by the median of their coordinates in $x$ and $y$:*

$$m^{(x)} = \textit{median}\left(p^{(i,x)}\right) \tag{1}$$

$$m^{(y)} = \textit{median}\left(p^{(i,y)}\right). \tag{2}$$

**Definition 2.** *For a set $P = \{p^{(i)}\}$ of $n$ points in $\mathbb{R}^2$, i.e., $p^{(i)} = \{p^{(i,x)}, p^{(i,y)}\}$, the min-max center $z = \{z^{(x)}, z^{(y)}\}$ is found by the average of the extremes in $x$ and $y$:*

$$z^{(x)} = \frac{\max\limits_{i=1,\ldots,n}\left(p^{(i,x)}\right) + \min\limits_{i=1,\ldots,n}\left(p^{(i,x)}\right)}{2} \tag{3}$$

$$z^{(y)} = \frac{\max\limits_{i=1,\ldots,n}\left(p^{(i,x)}\right) + \min\limits_{i=1,\ldots,n}\left(p^{(i,x)}\right)}{2}. \tag{4}$$

The median center is the result of the minimization of the sum of the squared Manhattan distances from each point to the center. On the other hand, the min-max center is obtained by minimizing the maximum of the maximum marginal distances (the distance from a point to the center on $x$ or $y$). The first method is used in cases in which the stability of the solution is of prime concern, while the second is used when the maximum cost needs to be minimized. The median center is only affected by changes in the middle points, but changes in extreme points affect only the min-max center. The selection of the appropriate method to find the center depends on which points are most likely to change and on which objective is pursued [7].

## 3 Fuzzy Sets and Fuzzy Numbers

When it is difficult to say that an object clearly belongs to a class, classical set theory loses its usefulness. The fuzzy sets theory [19] overcomes this problem by assigning degrees of membership of elements to a set. In this section we will recall the concepts of the fuzzy set theory that will be used in this paper.

### 3.1 Basic Definitions

**Definition 3.** *A fuzzy subset $\underset{\sim}{A}$ is a set whose elements do not follow the law of the excluded middle that rules over Boolean logic, i.e., their membership function is mapped as:*

$$\mu_{\underset{\sim}{A}} : X \to [0, 1]. \tag{5}$$

**Definition 4.** *In general, a fuzzy subset $\underset{\sim}{A}$ can be represented by a set of pairs composed of the elements $x$ of the universal set $X$, and a grade of membership $\mu_{\underset{\sim}{A}}(x)$:*

$$\underset{\sim}{A} = \left\{ \left( x, \mu_{\underset{\sim}{A}}(x) \right) \mid x \in X \, , \, \mu_{\underset{\sim}{A}}(x) \in [0, 1] \right\}. \tag{6}$$

**Definition 5.** *An $\alpha$-cut of a fuzzy subset $\underset{\sim}{A}$ is defined by:*

$$A_\alpha = \left\{ x \in X : \, \mu_{\underset{\sim}{A}}(x) \geq \alpha \right\}, \tag{7}$$

*i.e., the subset of all elements that belong to $\underset{\sim}{A}$ at least in a degree $\alpha$.*

**Definition 6.** *A fuzzy subset $\underset{\sim}{A}$ is convex, if and only if:*

$$\lambda x_1 + (1 - \lambda x_2) \in A_\alpha \, \forall x_1, x_2 \in A_\alpha, \, \alpha, \lambda \in [0, 1], \tag{8}$$

*i.e., all the points in $[x_1, x_2]$ must belong to $A_\alpha$, for any $\alpha$.*

**Definition 7.** *A fuzzy subset $\underset{\sim}{A}$ is normal, if and only if:*

$$\max_{x \in X} \left( \mu_{\underset{\sim}{A}}(x) \right) = 1. \tag{9}$$

**Definition 8.** *The core of a fuzzy subset $\underset{\sim}{A}$ is defined as:*

$$N_{\underset{\sim}{A}} = \left\{ x : \, \mu_{\underset{\sim}{A}}(x) = 1 \right\}. \tag{10}$$

**Definition 9.** *A fuzzy number $\underset{\sim}{A}$ is a normal, convex fuzzy subset with domain in $\mathbb{R}$ for which:*

1. $\bar{x} := N_{\underset{\sim}{A}}$, $\text{card}\,(\bar{x}) = 1$, and
2. $\mu_{\underset{\sim}{A}}$ is at least piecewise continuous.

The mean value $\bar{x}$ [20], also called maximum of presumption [14], identifies a fuzzy number in such a way that the proposition "about 9" can be modeled with a fuzzy number whose maximum of presumption is $x = 9$. As Zimmermann explains, for computational simplicity there is a tendency to call "fuzzy number" any normal, convex fuzzy subset whose membership function is, at least, piecewise continuous, without taking into consideration the uniqueness of the maximum of presumption. Thus, this definition will include "fuzzy intervals", fuzzy numbers in which $\bar{x}$ covers an interval[1], and particularly trapezoidal fuzzy numbers (TrFN).

**Definition 10.** *A TrFN is defined by the membership function:*

$$\mu_{\underset{\sim}{A}}(x) = \begin{cases} 1 - \frac{x_2 - x}{x_2 - x_1}, & \text{if } x_1 \leq x < x_2 \\ 1, & \text{if } x_2 \leq x \leq x_3 \\ 1 - \frac{x - x_3}{x_4 - x_3}, & \text{if } x_3 < x \leq x_4 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

This kind of fuzzy interval represents the case when the maximum of presumption, the modal value, can not be identified in a single point, but only in an interval between $x_2$ and $x_3$, decreasing linearly to zero at the worst case deviations $x_1$ and $x_4$. The TrFN is represented by a 4-tuple whose first and fourth elements correspond to the extremes from where the membership function begins to grow, and whose second and third components are the limits of the interval where the maximum certainty lies, i.e., $\underset{\sim}{A} = (x_1, x_2, x_3, x_4)$.

**Definition 11.** *The image of a TrFN is defined as:*

$$Im\left(\underset{\sim}{A}\right) = (-a_4, -a_3, -a_2, -a_1)\,.$$

**Definition 12.** *The addition and subtraction of two TrFN $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are defined as:*

$$\underset{\sim}{A} \oplus \underset{\sim}{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4) \tag{12}$$

$$\underset{\sim}{A} \ominus \underset{\sim}{B} = \underset{\sim}{A} \oplus Im(\underset{\sim}{B}). \tag{13}$$

### 3.2 Miscelaneous Definitions

Comparing fuzzy numbers is a task that can only be achieved via defuzzification, i.e., by calculating its expected value. For its simplicity, we have selected the graded mean integrated representation (GMIR) of a TrFN [5] as the method used in this paper to defuzzify and compare TrFN.

**Definition 13.** *The GMIR of non-normal TrFN is:*

$$E\left(\underset{\sim}{M}\right) = \frac{\int_0^{\max\left(\mu_{\underset{\sim}{M}}\right)} \frac{\mu}{2}\left(L_{\underset{\sim}{M}}^{-1}(\mu) + R_{\underset{\sim}{M}}^{-1}(\mu)\right) d\mu}{\int_0^{\max\left(\mu_{\underset{\sim}{M}}\right)} \mu \, d\mu}. \tag{14}$$

[1] As a matter of fact, they are also called "flat fuzzy numbers" [10].

*Remark 1.* For a normal TrFN as defined in (11), the GMIR is:

$$E\left(\underset{\sim}{A}\right) = \frac{a_1 + 2a_2 + 2a_3 + a_4}{6}. \tag{15}$$

*Remark 2.* The GMIR is linear, i.e., $E(\underset{\sim}{A} \oplus \underset{\sim}{B}) = E(\underset{\sim}{A}) + E(\underset{\sim}{B})$ and $E(\alpha \cdot \underset{\sim}{A}) = \alpha \cdot E(\underset{\sim}{A})$.

To calculate the distance between two TrFN, we must first define the absolute value of a TrFN. We will rely on the work of [6] for this.

**Definition 14.** *The absolute value of a TrFN is defined as:*

$$\left|\underset{\sim}{A}\right| = \begin{cases} \underset{\sim}{A}, & \text{if } E\left(\underset{\sim}{A}\right) > 0 \\ 0, & \text{if } E\left(\underset{\sim}{A}\right) = 0 \\ Im\left(\underset{\sim}{A}\right), & \text{if } E\left(\underset{\sim}{A}\right) < 0. \end{cases} \tag{16}$$

**Proposition 1.** *For a TrFN $\underset{\sim}{A}$, $E(\left|\underset{\sim}{A}\right|) = \left|E(\underset{\sim}{A})\right|$.*

*Proof.* For $E(\underset{\sim}{A}) \geq 0$ the proof is trivial. For $E(\underset{\sim}{A}) < 0$ we have:

$$\begin{aligned} E\left(\left|\underset{\sim}{A}\right|\right) &= E\left(Im\left(\underset{\sim}{A}\right)\right) \\ &= \frac{-a_4 - 2a_3 - 2a_2 - a_1}{6} \\ &= -E\left(\underset{\sim}{A}\right) \\ &= \left|E\left(\underset{\sim}{A}\right)\right|. \end{aligned}$$

**Definition 15.** *The fuzzy Minkoswki family of distances between two fuzzy $n$-dimensional vectors $\underset{\sim}{A}$ and $\underset{\sim}{B}$ composed of TrFN:*

$$\underset{\sim}{d}_p\left(\underset{\sim}{A}, \underset{\sim}{B}\right) = \left(\sum_{i=1}^{n} \left(\left|\underset{\sim}{A}_i \ominus \underset{\sim}{B}_i\right|\right)^p\right)^{\frac{1}{p}}. \tag{17}$$

*Remark 3.* As with the crisp Minkowski family of distances, the fuzzy Manhattan distance is defined for $p = 1$, the fuzzy Euclidean distance is defined for $p = 2$, and the fuzzy Chebyshev distance is defined for $p = \infty$.

*Remark 4.* In our proofs, we will use the form:

$$\underset{\sim}{d}^p\left(\underset{\sim}{A}, \underset{\sim}{B}\right) = \sum_{i=1}^{n} \left(\left|\underset{\sim}{A}_i \ominus \underset{\sim}{B}_i\right|\right)^p, \tag{18}$$

except for $p = \infty$ in which:

$$\underset{\sim}{d}^{\infty}\left(\underset{\sim}{A}, \underset{\sim}{B}\right) = \underset{\left|\underset{\sim}{A}_i \ominus \underset{\sim}{B}_i\right|}{\arg} \max_{i=1}^{n} E\left(\left|\underset{\sim}{A}_i \ominus \underset{\sim}{B}_i\right|\right). \tag{19}$$

## 4 Fuzzy Median Center and Fuzzy Min-Max Center

We will prove that for a set of fuzzy points, the fuzzy median center and the fuzzy min-max center are extensions of their respective counterparts in crisp settings, i.e., that they can be obtained by the median or the average of the maximum $X$ and $Y$ coordinates of the fuzzy points, respectively.

**Proposition 2.** *For two TrFN* $\underset{\sim}{p^{(1)}}$ *and* $\underset{\sim}{p^{(2)}}$*, such that* $E(\underset{\sim}{p^{(1)}}) < E(\underset{\sim}{p^{(2)}})$, $\arg\min_{\underset{\sim}{c}} E(\sum_{i\in\{1,2\}} \underset{\sim}{d}^1(\underset{\sim}{p^{(i)}}, \underset{\sim}{c})) = \{\underset{\sim}{c} : E(\underset{\sim}{c}) \in [E(\underset{\sim}{p^{(1)}}), E(\underset{\sim}{p^{(2)}})]\}$.

*Proof.* Let $\underset{\sim}{p^{(i)}} = (p_1^{(i)}, p_2^{(i)}, p_3^{(i)}, p_4^{(i)})$ and $\underset{\sim}{c} = (c_1, c_2, c_3, c_4)$, hence:

$$\underset{\sim}{d}^1\left(\underset{\sim}{p^{(i)}}, \underset{\sim}{c}\right) = \left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{c}\right|.$$

By properties of the GMIR:

$$E\left(\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{c}\right) = E\left(\underset{\sim}{p^{(i)}}\right) - E\left(\underset{\sim}{c}\right).$$

If $E(\underset{\sim}{c}) \leq E(\underset{\sim}{p^{(1)}})$ and by (16), then:

$$\underset{\sim}{d}^1\left(\underset{\sim}{p^{(1)}}, \underset{\sim}{c}\right) = \underset{\sim}{p^{(1)}} \ominus \underset{\sim}{c}, \tag{20}$$

$$\underset{\sim}{d}^1\left(\underset{\sim}{p^{(2)}}, \underset{\sim}{c}\right) = \underset{\sim}{p^{(2)}} \ominus \underset{\sim}{c}. \tag{21}$$

By (20) and (21): as $\underset{\sim}{p^{(1)}} \ominus \underset{\sim}{c} = (0,0,0,0)$ and $\underset{\sim}{p^{(2)}} \ominus \underset{\sim}{c} = \underset{\sim}{p^{(2)}} \ominus \underset{\sim}{p^{(1)}}$. For any $\{\underset{\sim}{c} : E(\underset{\sim}{c}) < E(\underset{\sim}{p^{(1)}})\}$, $E(\underset{\sim}{p^{(2)}} \ominus \underset{\sim}{c}) > 0$ and $E(\underset{\sim}{c} \ominus \underset{\sim}{p^{(1)}}) > E(\underset{\sim}{p^{(2)}} \ominus \underset{\sim}{p^{(1)}})$.

Equivalently, if $E(\underset{\sim}{p^{(2)}}) \leq E(\underset{\sim}{c})$ by (16), then:

$$\underset{\sim}{d}^1\left(\underset{\sim}{p^{(1)}}, \underset{\sim}{c}\right) = \underset{\sim}{c} \ominus \underset{\sim}{p^{(1)}}, \tag{22}$$

$$\underset{\sim}{d}^1\left(\underset{\sim}{p^{(2)}}, \underset{\sim}{c}\right) = \underset{\sim}{c} \ominus \underset{\sim}{p^{(2)}}. \tag{23}$$

By (22) and (23),

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i\in\{1,2\}} \left(\underset{\sim}{d}^1\left(\underset{\sim}{p^{(i)}}, \underset{\sim}{c}\right)\right)\right) = \underset{\sim}{p^{(2)}},$$

as $\underset{\sim}{c} \ominus \underset{\sim}{p^{(2)}} = (0,0,0,0)$ and $\underset{\sim}{c} \ominus \underset{\sim}{p^{(1)}} = \underset{\sim}{p^{(2)}} \ominus \underset{\sim}{p^{(1)}}$. For any $\{\underset{\sim}{c} : E(\underset{\sim}{p^{(2)}}) < E(\underset{\sim}{c})\}$, $E(\underset{\sim}{c} \ominus \underset{\sim}{p^{(2)}}) > 0$ and $E(\underset{\sim}{c} \ominus \underset{\sim}{p^{(1)}}) > E(\underset{\sim}{p^{(2)}} \ominus \underset{\sim}{p^{(1)}})$.

Given that $E(\underset{\sim}{p}^{(1)}) < E(\underset{\sim}{c})$ and by (16), then:

$$\begin{aligned}\underset{\sim}{d}^1\left(\underset{\sim}{p}^{(1)}, \underset{\sim}{c}\right) &= \underset{\sim}{c} \ominus \underset{\sim}{p}^{(1)} \\ &= \left(c_1 - p_4^{(1)}, c_2 - p_3^{(1)}, c_3 - p_2^{(1)}, c_1 - p_4^{(1)}\right). \end{aligned} \tag{24}$$

Given that $E(\underset{\sim}{c}) < E(\underset{\sim}{p}^{(2)})$ and by (16), then:

$$\begin{aligned}\underset{\sim}{d}^1\left(\underset{\sim}{p}^{(2)}, \underset{\sim}{c}\right) &= \underset{\sim}{p}^{(2)} \ominus \underset{\sim}{c} \\ &= \left(p_1^{(2)} - c_4, p_2^{(2)} - c_3, p_3^{(2)} - c_2, p_4^{(2)} - c_1\right). \end{aligned} \tag{25}$$

From (24) and (25):

$$\begin{aligned}\sum_{i\in\{1,2\}} \left(\underset{\sim}{d}^1\left(\underset{\sim}{p}^{(i)}, \underset{\sim}{c}\right)\right) &= \left(c_1 - p_4^{(1)}, c_2 - p_3^{(1)}, c_3 - p_2^{(1)}, c_4 - p_1^{(1)}\right) \oplus \\ &\quad \left(p_1^{(2)} - c_4, p_2^{(2)} - c_3, p_3^{(2)} - c_2, p_4^{(2)} - c_1\right) \\ &= \left(p_1^{(2)} - p_1^{(1)} + c_1 - c_4, p_2^{(2)} - p_2^{(1)} + c_2 - c_3,\right. \\ &\quad \left. p_3^{(2)} - p_3^{(1)} + c_3 - c_2, p_4^{(2)} - p_4^{(1)} + c_4 - c_1\right). \end{aligned} \tag{26}$$

Applying GMIR to (26) :

$$E\left(\textstyle\sum_{i\in\{1,2\}}\left(\underset{\sim}{d}^1\left(\underset{\sim}{p}^{(i)}, \underset{\sim}{c}\right)\right)\right) = E\left(\underset{\sim}{p}^{(2)} - \underset{\sim}{p}^{(1)}\right). \tag{27}$$

Being that (27) is independent from $\underset{\sim}{c}$:

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i\in\{1,2\}}\left(\underset{\sim}{d}^1\left(\underset{\sim}{p}^{(i)}, \underset{\sim}{c}\right)\right)\right) = \left\{\underset{\sim}{c} :\ E\left(\underset{\sim}{c}\right) \in \left[E\left(\underset{\sim}{p}^{(1)}\right), E\left(\underset{\sim}{p}^{(2)}\right)\right]\right\}.$$

The result obtained in Proposition 2 shows than any fuzzy point $\underset{\sim}{c}$ between two fuzzy points $\underset{\sim}{p}^{(1)}$ and $\underset{\sim}{p}^{(2)}$ gives an equally good solution to the problem of the minimization of distances. An arbitrary, but frequently found solution to the crisp version of this problem, is using the average of both points:

$$\underset{\sim}{c} = \frac{\underset{\sim}{p}^{(1)} \oplus \underset{\sim}{p}^{(2)}}{2}. \tag{28}$$

In the following proposition we will see what happens for a set of $n$ fuzzy points, but first, let us define the notion of order statistic for fuzzy numbers.

**Definition 16.** *For a set $P = \{\underset{\sim}{p^{(i)}}\}$, $\forall i = 1, \ldots, n$, of TrFN, the $k-$th order statistic $\underset{\sim}{p^{([k])}}$ is defined as the $k-$th point for which $E(\underset{\sim}{p^{([k])}}) \leq E(\underset{\sim}{p^{([k+1])}})$.*

**Proposition 3.** *For a set $P = \{\underset{\sim}{p^{(i)}}\}$, $i = 1, \ldots, n$, of TrFN, $\underset{\sim}{c}^*$ is the point for which $\arg\min_{\underset{\sim}{c}} E\left(\sum_{i=1}^{n} \underset{\sim}{d}^1\left(\underset{\sim}{p^{([i])}}, \underset{\sim}{c}\right)\right) = \left\{\underset{\sim}{c} :\ E(\underset{\sim}{c}) \in \left[E(\underset{\sim}{p^{([\frac{n}{2}])}}), E(\underset{\sim}{p^{([\frac{n}{2}+1])}})\right]\right\}$, if $n$ is even, but if it is odd $\arg\min_{\underset{\sim}{c}} E\left(\sum_{i=1}^{n} \underset{\sim}{d}^1\left(\underset{\sim}{p^{([i])}}, \underset{\sim}{c}\right)\right) = \underset{\sim}{p^{([\frac{n+1}{2}])}}$.*

*Proof.* Given that the $k-$th order statistic of the set $P$ is $\underset{\sim}{p^{([k])}}$, we can apply iteratively the result in Proposition 2. In first place:

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i\in\{1,n\}} \underset{\sim}{d}^1\left(\underset{\sim}{p^{([i])}}, \underset{\sim}{c}\right)\right) = \left\{\underset{\sim}{c} :\ E\left(\underset{\sim}{c}\right) \in \left[E\left(\underset{\sim}{p^{([1])}}\right), E\left(\underset{\sim}{p^{([n])}}\right)\right]\right\}.$$

From Definition 4:

$$\left[E\left(\underset{\sim}{p^{([2])}}\right), E\left(\underset{\sim}{p^{([n-1])}}\right)\right] \in \left[E\left(\underset{\sim}{p^{([1])}}\right), E\left(\underset{\sim}{p^{([n])}}\right)\right],$$

so the solution is now:

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i\in\{1,2,n-1,n\}} \underset{\sim}{d}^1\left(\underset{\sim}{p^{([i])}}, \underset{\sim}{c}\right)\right) = \left\{\underset{\sim}{c} :\ E\left(\underset{\sim}{c}\right) \in \left[E\left(\underset{\sim}{p^{([2])}}\right), E\left(\underset{\sim}{p^{([n-1])}}\right)\right]\right\}.$$

If we keep applying iteratively this logic, and $n$ is even, we get that

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i=1}^{n} \underset{\sim}{d}^1\left(\underset{\sim}{p^{([i])}}, \underset{\sim}{c}\right)\right) = \left\{\underset{\sim}{c} :\ E\left(\underset{\sim}{c}\right) \in \left[E\left(\underset{\sim}{p^{([\frac{n}{2}])}}\right), E\left(\underset{\sim}{p^{([\frac{n}{2}+1])}}\right)\right]\right\}.$$

If $n$ is odd, we will have three points in the next-to-last iteration, $\left\{\underset{\sim}{p^{([\frac{n-1}{2}])}}, \underset{\sim}{p^{([\frac{n+1}{2}])}}, \underset{\sim}{p^{([\frac{n+3}{2}])}}\right\}$. We can present the problem as:

$$\arg\min_{\underset{\sim}{c}} E\sum_{i=1}^{n}\left(\underset{\sim}{d}^1\left(\underbrace{p^{([i])}}_{\sim},\underset{\sim}{c}\right)\right) =$$

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i=\frac{n-1}{2}}^{\frac{n-3}{2}}\underset{\sim}{d}^1\left(\underbrace{p^{([i])}}_{\sim},\underset{\sim}{c}\right)\right)$$

$$= \arg\min_{\underset{\sim}{c}} E\left(\sum_{i=\left\{\frac{n-1}{2},\frac{n-3}{2}\right\}}\underset{\sim}{d}^1\left(\underbrace{p^{([i])}}_{\sim},\underset{\sim}{c}\right) + \underset{\sim}{d}^1\left(\underbrace{p^{([\frac{n+1}{2}])}}_{\sim},\underset{\sim}{c}\right)\right).$$

We know that:

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i=\left\{\frac{n-1}{2},\frac{n-3}{2}\right\}}\left(\underset{\sim}{d}^1\left(\underbrace{p^{(i)}}_{\sim},\underset{\sim}{c}\right)\right)\right) =$$

$$\left\{\underset{\sim}{c}:\ E\left(\underset{\sim}{c}\right)\in\left[E\left(\underbrace{p^{([\frac{n-1}{2}])}}_{\sim}\right),E\left(\underbrace{p^{([\frac{n-3}{2}])}}_{\sim}\right)\right]\right\}.$$

Therefore, it is clear that:

$$\arg\min_{\underset{\sim}{c}} E\left(\underset{\sim}{d}^1\left(\underbrace{p^{([\frac{n+1}{2}])}}_{\sim},\underset{\sim}{c}\right)\right) = \underbrace{p^{([\frac{n+1}{2}])}}_{\sim}.$$

So, given that $\underset{\sim}{c} = \underbrace{p^{([\frac{n+1}{2}])}}_{\sim}$ and that $E(p^{([\frac{n+1}{2}])}) \in [E(\underbrace{p^{([\frac{n-1}{2}])}}_{\sim}), E(\underbrace{p^{([\frac{n-3}{2}])}}_{\sim})]$, for $n$ even:

$$\arg\min_{\underset{\sim}{c}} E\left(\sum_{i=1}^{n}\left(\underset{\sim}{d}^1\left(\underbrace{p^{(i)}}_{\sim},\underset{\sim}{c}\right)\right)\right) = \underbrace{p^{([\frac{n+1}{2}])}}_{\sim}.$$

Applying (28) to the result of Proposition 3 we get the definition of the median for a set of TrFN.

**Proposition 4.** *The median of a set* $P = \{\underbrace{p^{(i)}}_{\sim}\}$, $\forall i = 1,\ldots,n$, *of TrFN is defined as:*

$$\textit{median}\,(P) = \begin{cases} \dfrac{\underbrace{p^{([\frac{n}{2}])}}_{\sim}\oplus\underbrace{p^{([\frac{n}{2}+1])}}_{\sim}}{2}, & \text{if } n \text{ is odd,} \\ \underbrace{p^{([\frac{n+1}{2}])}}_{\sim}, & \text{if } n \text{ is even.} \end{cases}$$

In an $\mathbb{R}^2$ space, the solution is equivalent, as we will see in the following proposition.

**Proposition 5.** *For a set* $P = \left\{\underbrace{P^{(i)}}_{\sim} : \underbrace{P^{(i)}}_{\sim} = \left\{\underbrace{p^{(i,j)}}_{\sim}\right\}, \forall i = 1,\ldots,n,\ j \in \{x,y\}\right\}$, *where* $\underbrace{p^{(i,j)}}_{\sim}$ *is a TrFN,* $\arg\min_{\underset{\sim}{C}} E\left(\sum_{i=1}^{n}\underset{\sim}{d}^1\left(\underbrace{P^{(i)}}_{\sim},\underset{\sim}{C}\right)\right) = \left\{\textit{median}\left(\underbrace{p^{(i,x)}}_{\sim}\right),\textit{median}\left(\underbrace{p^{(i,y)}}_{\sim}\right)\right\}$.

*Proof.* Due to the linearity of the GMIR:

$$E\left(\sum_{i=1}^{n}\left(\underset{\sim}{d}^1\left(\underset{\sim}{P}^{(i)},\underset{\sim}{C}\right)\right)\right)=\sum_{i=1}^{n}\sum_{j\in\{x,y\}}\left|E\left(\underset{\sim}{p}^{(i,j)}\right)-E\left(\underset{\sim}{c}^{j}\right)\right| \tag{29}$$

$$=\sum_{i=1}^{n}\left|E\left(\underset{\sim}{p}^{(i,x)}\right)-E\left(\underset{\sim}{c}^{(x)}\right)\right|+ \tag{30}$$

$$\sum_{i=1}^{n}\left|E\left(\underset{\sim}{p}^{(i,y)}\right)-E\left(\underset{\sim}{c}^{(y)}\right)\right|. \tag{31}$$

As both terms in (31) are independent from each other:

$$\min_{c^{(j)}}\left(\sum_{j\in\{x,y\}}\sum_{i=1}^{n}\left|E\left(\underset{\sim}{p}^{(i,x)}\right)-E\left(\underset{\sim}{c}^{(x)}\right)\right|\right)=$$

$$\sum_{j\in\{x,y\}}\min_{c^{(j)}}\sum_{i=1}^{n}\left|E\left(\underset{\sim}{p}^{(i,x)}\right)-E\left(\underset{\sim}{c}^{(x)}\right)\right|.$$

The optimization problem is then reduced to applying independently for each $j\in\{x,y\}$ the result of Proposition 3 with Definition 4. Thus:

$$\arg\min_{\underset{\sim}{C}}E\left(\sum_{i=1}^{n}\underset{\sim}{d}^1\left(\underset{\sim}{P}^{(i)},\underset{\sim}{C}\right)\right)=$$

$$\left\{\text{median}\left(\underset{\sim}{p}^{(i,x)}\right),\text{median}\left(\underset{\sim}{p}^{(i,y)}\right)\right\}. \tag{32}$$

Finally, we will address the subject of the fuzzy min-max center, found using (19).

**Proposition 6.** *For a set* $P=\left\{\underset{\sim}{p}^{(i)}\right\},\forall i=1,\ldots,n,$ *of TrFN,* $\max_{i=1}^{n}(E(\left|\underset{\sim}{p}^{(i)}\ominus\underset{\sim}{c}\right|)=\frac{1}{2}\cdot E(p^{([n])}-p^{([1])})$.

*Proof.* Let $E(\underset{\sim}{c})=\frac{1}{2}\left(E\left(p^{([1])}\right)+E\left(\underset{\sim}{p}^{([n])}\right)\right)$. Due to the linearity of the GMIR and Proposition 1, $\max_{i=1}^{n}\left(E\left(\left|\underset{\sim}{p}^{(i)}\ominus\underset{\sim}{c}\right|\right)\right)=\max_{i=1}^{n}\left|E(\underset{\sim}{p}^{(i)})-E(\underset{\sim}{c})\right|$.So:

$$-\frac{E\left(\underset{\sim}{p}^{([n])}\right)-E\left(p^{([1])}\right)}{2}\leq E\left(\underset{\sim}{p}^{([i])}\right)-\frac{E\left(\underset{\sim}{p}^{([n])}\right)+E\left(\underset{\sim}{p}^{([1])}\right)}{2}$$

$$\leq\frac{E\left(\underset{\sim}{p}^{([n])}\right)-E\left(p^{([1])}\right)}{2}$$

then:

$$\left| E\left(\underset{\sim}{p^{([i])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| \leq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(p^{([1])}\right)}{2}$$

$$\max_{i=1}^{n} \left| E\left(\underset{\sim}{p^{([i])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| \leq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(p^{([1])}\right)}{2}.$$

In fact:

$$\begin{aligned}
\left| E\left(\underset{\sim}{p^{([n])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| &= \left| \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| \\
&= \left| -\frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| \\
&= \left| E\left(\underset{\sim}{p^{([1])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| \\
&\geq \left| E\left(\underset{\sim}{p^{([i])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right|.
\end{aligned}$$

So:

$$\max_{i=1}^{n} \left| E\left(\underset{\sim}{p^{([i])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \right| = \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2},$$

i.e.:

$$\max_{i=1}^{n} \left| E\left(\underset{\sim}{p^{([i])}}\right) - E\left(\underset{\sim}{c}\right) \right| = \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2}.$$

**Proposition 7.** *For a TrFN* $\underset{\sim}{c'}$*, such that for every TrFN* $\underset{\sim}{p}$ $\max_{i=1}^{n}\left(E\left(\left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{p}\right|\right)\right) \geq \max_{i=1}^{n}\left(E\left(\left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{c'}\right|\right)\right)$*, then* $E(\underset{\sim}{c}) = E(\underset{\sim}{c'})$.

*Proof.* Let $E(\underset{\sim}{c}) = \frac{1}{2}(E(p^{([1])}) + E(\underset{\sim}{p^{([n])}}))$. Taking $\underset{\sim}{p} = \underset{\sim}{c}$:

$$\begin{aligned}\max_{i=1}^{n}\left(E\left(\left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{c'}\right|\right)\right) &\leq \max_{i=1}^{n}\left(E\left(\left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{c}\right|\right)\right) \\ &= \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2},\end{aligned}$$

so:

$$E\left(\left|\underset{\sim}{p^{([1])}} \ominus \underset{\sim}{c'}\right|\right) \leq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2}$$

$$E\left(\left|\underset{\sim}{p^{([n])}} \ominus \underset{\sim}{c'}\right|\right) \leq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2}$$

and:

$$\begin{aligned}E\left(\underset{\sim}{c'}\right) &\leq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2} + E\left(\underset{\sim}{p^{([1])}}\right) \\ &= \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2} \\ E\left(\underset{\sim}{c'}\right) &\geq E\left(\underset{\sim}{p^{([n])}}\right) - \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2} \\ &= \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2}.\end{aligned}$$

So:

$$E\left(\underset{\sim}{c'}\right) = \frac{E\left(\underset{\sim}{p^{([n])}}\right) + E\left(\underset{\sim}{p^{([1])}}\right)}{2}$$

**Proposition 8.** *For a set* $P = \left\{\underset{\sim}{p^{(i)}}\right\}, \forall i = 1, \ldots, n$, *of TrFN and a TrFN* $\underset{\sim}{p}$, $\max_{i=1}^{n}(E(\left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{p}\right|) = \max_{i=1}^{n}(E(\left|\underset{\sim}{p^{(i)}} \ominus \underset{\sim}{c}\right|)$.

*Proof.* Let $E(\underset{\sim}{c}) = \frac{1}{2}(E(p^{([1])}) + E(\underset{\sim}{p^{([n])}}))$. If $E(\underset{\sim}{p}) \leq E\left(\underset{\sim}{c}\right)$, $E(\underset{\sim}{p^{([n])}}) - E(\underset{\sim}{p}) \geq E(\underset{\sim}{p^{([n])}}) - E(\underset{\sim}{c})$. So:

$$\begin{aligned}\max_{i=1}^{n}\left|E(\underset{\sim}{p^{([i])}}) - E(\underset{\sim}{p})\right| &\geq \left|E(\underset{\sim}{p^{([n])}}) - E(\underset{\sim}{p})\right| \\ &\geq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2} \\ &= \max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{([i])}}\right) - E\left(\underset{\sim}{c}\right)\right|.\end{aligned}$$

If $E(\underset{\sim}{p}) \geq E\left(\underset{\sim}{c}\right)$, $E(\underset{\sim}{p}) - E(\underset{\sim}{p^{([1])}}) \geq E(\underset{\sim}{c}) - E(\underset{\sim}{p^{([1])}})$. So:

$$\begin{aligned}\max_{i=1}^{n}\left|E(\underset{\sim}{p^{([i])}}) - E(\underset{\sim}{p})\right| &\geq \left|E(\underset{\sim}{p}) - E(\underset{\sim}{p^{([1])}})\right| \\ &\geq \frac{E\left(\underset{\sim}{p^{([n])}}\right) - E\left(\underset{\sim}{p^{([1])}}\right)}{2} \\ &= \max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{([i])}}\right) - E\left(\underset{\sim}{c}\right)\right|.\end{aligned}$$

Again, due to the linearity of the GMIR, $\max_{i=1}^{n}\left|E(\underset{\sim}{p^{([i])}}) - E(\underset{\sim}{c})\right| = \max_{i=1}^{n} E\left|\underset{\sim}{p^{([i])}} \ominus \underset{\sim}{c}\right|$

**Proposition 9.** *For a set* $P = \{\underset{\sim}{P^{(i)}} : \underset{\sim}{P^{(i)}} = \{\underset{\sim}{p^{(i,j)}}\}\}$, $\forall i \in 1, \ldots, n$, $j \in \{x, y\}$, *where* $\underset{\sim}{p^{(i,j)}}$ *is a TrFN, and the fuzzy center* $\underset{\sim}{C} = \{\underset{\sim}{c^{(j)}}\}$, $\max_{i=1}^{n}(\underset{\sim}{d^{\infty}}(\underset{\sim}{P^{(i)}}, \underset{\sim}{P})) \geq \max_{i=1}^{n}(\underset{\sim}{d^{\infty}}(\underset{\sim}{P^{(i)}}, \underset{\sim}{C}))$.

*Proof.* Let $E(\underset{\sim}{c^{(j)}}) = \frac{1}{2}E(\underset{\sim}{p^{([1],j)}} \oplus \underset{\sim}{p^{([n],j)}})$ and a fuzzy point $\underset{\sim}{P} = \{\underset{\sim}{p^{(j)}}\}$. Then:

$$\begin{aligned}\max_{i=1}^{n}\underset{\sim}{d^{\infty}}\left(d^{\infty}\left(\underset{\sim}{P^{(i)}}, \underset{\sim}{P}\right)\right) &= \max_{i=1}^{n}\max_{j\in\{x,y\}} E\left(\left|\underset{\sim}{p^{(i,j)}} \ominus \underset{\sim}{p^{(j)}}\right|\right) \\ &= \max_{j\in\{x,y\}}\max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{(i,j)}}\right) - E\left(\underset{\sim}{p^{(j)}}\right)\right|.\end{aligned}$$

From Proposition 8, we will recall that:

$$\max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{(i,j)}}\right) - E\left(\underset{\sim}{p^{(j)}}\right)\right| \geq \max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{(i,j)}}\right) - E\left(\underset{\sim}{c^{(j)}}\right)\right|,$$

so:

$$\begin{aligned}\max_{j\in\{x,y\}}\max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{(i,j)}}\right) - E\left(\underset{\sim}{p^{(j)}}\right)\right| &\geq \max_{j\in\{x,y\}}\max_{i=1}^{n}\left|E\left(\underset{\sim}{p^{(i,j)}}\right) - E\left(\underset{\sim}{c^{(j)}}\right)\right| \\ &= \max_{i=1}^{n}\underset{\sim}{d^{\infty}}\left(\underset{\sim}{P^{(i)}}, \underset{\sim}{C}\right).\end{aligned}$$

**Proposition 10.** *For a set* $P = \{\underset{\sim}{P^{(i)}} : \underset{\sim}{P^{(i)}} = \{\underset{\sim}{p^{(i,j)}}\}, \forall i \in 1, \ldots, n, j \in \{x, y\}\}$, *where* $\underset{\sim}{p^{(i,j)}}$ *is a TrFN, the fuzzy min-max center* $\underset{\sim}{C}^* = \left\{\underset{\sim}{c^{(j)}}\right\}$ *is* $\arg\min_{\underset{\sim}{C}} \max_{i=1}^{n} \underset{\sim}{d}^{\infty}\left(\underset{\sim}{P^{(i)}}, \underset{\sim}{C}\right) = \{\underset{\sim}{c} : E(c^{(j)}) = \frac{1}{2}E(\underset{\sim}{p^{([1],j)}} \oplus \underset{\sim}{p^{([n],j)}})\}$.

*Proof.* Let the fuzzy point $\underset{\sim}{P} = \{\underset{\sim}{p^{(j)}}\}$, then:

$$\begin{aligned}\max_{i=1}^{n} E\left(\underset{\sim}{d}^{\infty}\left(\underset{\sim}{P^{(i)}}, \underset{\sim}{P}\right)\right) &= \max_{i=1}^{n} \max_{j\in\{x,y\}} E\left(\left|\underset{\sim}{p^{(i,j)}} \ominus \underset{\sim}{p^{(j)}}\right|\right) \\ &= \max_{i=1}^{n} \max_{j\in\{x,y\}} \left|E\left(\underset{\sim}{p^{(i,j)}}\right) - E\left(\underset{\sim}{p^{(j)}}\right)\right|.\end{aligned}$$

By the result of Proposition 9:

$$E\left(\underset{\sim}{c^{(j)}}\right) = \frac{E\left(\underset{\sim}{p^{(i,j)}}\right) + E\left(\underset{\sim}{p^{(j)}}\right)}{2}.$$

Then:

$$\max_{i=1}^{n} \underset{\sim}{d}^{\infty}\left(\underset{\sim}{P^{(i)}}, \underset{\sim}{P}\right) \geq \max_{i=1}^{n} \underset{\sim}{d}^{\infty}\left(\underset{\sim}{P^{(i)}}, \underset{\sim}{C}\right).$$

Given that the solution of the fuzzy min-max center is a set of fuzzy points, we will extend the result for crisp values with the following definition.

**Definition 17.** *For a set* $P = \left\{\underset{\sim}{P^{(i)}} : \underset{\sim}{P^{(i)}} = \left\{\underset{\sim}{p^{(i,j)}}\right\}, \forall i \in 1, \ldots, n, j \in \{x, y\}\right\}$, *where* $\underset{\sim}{p^{(i,j)}}$ *is a TrFN, the coordinates of the fuzzy min-max center* $\underset{\sim}{C} = \left\{\underset{\sim}{c^{(j)}}\right\}$ *are defined as:*

$$\underset{\sim}{c^{(j)}} := \frac{\underset{\sim}{p^{([1],j)}} \oplus \underset{\sim}{p^{([n],j)}}}{2}. \tag{33}$$

## 5 Spatiotemporal Analysis

Throughout this work, the solutions to the optimal location problem were obtained without including any constraints into the model. However, from a practical perspective, it is necessary to consider the scenario in which the problem is defined and the restrictions it imposes to the solution. One way of doing it relies on expert knowledge for the creation of a bi-dimensional matrix which reflects the feasibility of the solution including a given coordinate.

Let's start by defining this matrix $\mathsf{M} = (m_{\iota,\kappa})_{M_y \times M_x}$, where $M_x$ (conversely $M_y$) is the number of columns (conversely rows) in which the area of study (which must fully include all the fuzzy points) is divided, with each element $m_{\iota,\kappa} \in [0, 1]$. By constraining

the range of $m_{\iota,\kappa}$ to the $[0,1]$ interval and making $M_x, M_y \to \infty$, $\mathsf{M}$ becomes a fuzzy subset $\underset{\sim}{M}$ that models the proposition "the solution to the problem of optimal location can occupy the coordinate $x, y$."

One notable property of fuzzy sets is that as sets they can be intersected, allowing complex interactions between them. We can use the fuzzy set that models the area of study, to obtain a constrained solution from its intersection with the unconstrained one. Thus, this new solution is defined by:

$$\underset{\sim}{\Phi} = \underset{\sim}{M} \cap \underset{\sim}{C}, \tag{34}$$

for which:

$$\mu_{\underset{\sim}{\Phi}}(x,y) = \min\left(\mu_{\underset{\sim}{M}}(x,y), \mu_{\underset{\sim}{C}}(x,y)\right). \tag{35}$$

It is clear that if $\underset{\sim}{\Phi} = \emptyset$ then the solution is unfeasible, as $\mu_{\underset{\sim}{\Phi}}(x,y) = 0$, $\forall x, y$.

Another layer of analysis can be applied when we consider the temporal evolution of the data. If data is gathered from different instants, fuzzy points can vary both in shape and location. Let's suppose that at any given instant $t = t_0, t_1, \ldots, t_{\max}$ we have the set $P^t$ of fuzzy points, which gives the constrained solution $\underset{\sim}{\Phi^t}$, i.e., an optimal constrained solution for the particular instant $t$. Thus, the solution for $t = t_0$ can be radically different from that for $t = t_{\max}$, and if we are going to build the center of attention of demand using the former solution, it might not be appropriate for latter instant.

In terms of the period $[t_0, t_{\max}]$, an idea of how "durable" a solution is can be attained from the intersection of the solutions obtained at each time frame, thus:

$$\underset{\sim}{\Phi} = \bigcap_{t=t_0}^{t_{\max}} \underset{\sim}{\Phi^t}, \tag{36}$$

for which:

$$\mu_{\underset{\sim}{\Phi}}(x,y) = \min\left(\mu_{\underset{\sim}{\Phi^t}}(x,y)\right). \tag{37}$$

The value of the membership function of the intersected solution at a given point in space can be seen as a degree of appropriateness of the final location for the whole period. But it can also be interpreted as the degree of durability of the solution, specially if the data points follow a progressive evolutionary pattern which resembles many phenomena like urban sprawl or forest shrinkage. When $\underset{\sim}{\Phi} = \emptyset$, no solution will be durable for the evaluated period, i.e., no point can be useful for the whole period; by removing solutions from $t_{\max}$ to $t$ until the intersected solution shows values of its membership function that satisfy the decision maker, it is possible to determine the durability of a selected location. Let's define $\underset{\sim}{\Theta^{t_i}}$, the intersected solution for $t' = t_0, \ldots, t_i$, $i = 0, \ldots, n$:

$$\underset{\sim}{\Theta^{t_i}} = \bigcap_{t'=t_0}^{t_i} \underset{\sim}{\Phi^{t'}}, \tag{38}$$

for which:

$$\mu_{\underset{\sim}{\Theta^{t_i}}}(x,y) = \min_{x,y}\left(\mu_{\underset{\sim}{\Phi^{t'}}}(x,y)\right). \tag{39}$$

One idea is to find, for a given $x_0, y_0$, the farthest $t_i$ for which $\mu_{\underset{\sim}{\Theta^{t_i}}}(x_0, y_0) > \mu_{min}$, i.e.:

$$t'_{\max} = \underset{t_i}{\arg\min} \left( \mu_{\underset{\sim}{\Theta^{t_i}}}(x_0, y_0) \right), \forall t_i : \mu_{\underset{\sim}{\Theta^{t_i}}}(x_0, y_0) > \mu_{\min}. \tag{40}$$

In case no coordinate $x, y$ is given, it is only needed to find the farthest $t_i$ that for any $x, y$ makes $\mu_{\underset{\sim}{\Theta^{t_i}}}(x, y) > \mu_{\min}$, i.e.:

$$t'_{\max} = \underset{t_i}{\arg\min} \left( \max_{x,y} \left( \mu_{\underset{\sim}{\Theta^{t_i}}}(x, y) \right) \right), \forall t_i : \max_{x,y} \left( \mu_{\underset{\sim}{\Theta^{t_i}}}(x, y) \right) > \mu_{\min}. \tag{41}$$

Thus, the solution is durable, at a threshold $\mu_{\min}$, for the period $[t_0, t'_{\max}]$.

## 6 Numerical Example

In the following numerical example we will see how the two centers are found and how they differ from each other. Let's suppose there are three fuzzy demand points:

$$\begin{aligned} \underset{\sim}{P^{(1)}} &= \left\{ \underset{\sim}{p^{(1,x)}}, \underset{\sim}{p^{(1,y)}} \right\} \\ \underset{\sim}{p^{(1,x)}} &= (18, 35, 37, 40) \\ \underset{\sim}{p^{(1,y)}} &= (31, 49, 49, 68) \end{aligned}$$

$$\begin{aligned} \underset{\sim}{P^{(2)}} &= \left\{ \underset{\sim}{p^{(2,x)}}, \underset{\sim}{p^{(2,y)}} \right\} \\ \underset{\sim}{p^{(2,x)}} &= (58, 75, 75, 94) \\ \underset{\sim}{p^{(2,y)}} &= (87, 103, 105, 121) \end{aligned}$$

$$\begin{aligned} \underset{\sim}{P^{(3)}} &= \left\{ \underset{\sim}{p^{(3,x)}}, \underset{\sim}{p^{(3,y)}} \right\} \\ \underset{\sim}{p^{(3,x)}} &= (73, 83, 86, 107) \\ \underset{\sim}{p^{(3,y)}} &= (10, 20, 21, 29) \end{aligned}$$

The expected values for these three points would be:

$$E\left( \underset{\sim}{p^{(1,x)}} \right) = 33.667$$
$$E\left( \underset{\sim}{p^{(1,y)}} \right) = 49.167$$

$$E\left( \underset{\sim}{p^{(2,x)}} \right) = 75.333$$
$$E\left( \underset{\sim}{p^{(2,y)}} \right) = 104$$

$$E\left( \underset{\sim}{p^{(3,x)}} \right) = 86.333$$
$$E\left( \underset{\sim}{p^{(3,y)}} \right) = 20.167$$

For these points, the fuzzy median center (see Figure 1a) would be:

$$\begin{aligned}
\underset{\sim}{M} &= \left\{\underset{\sim}{m}^{(x)}, \underset{\sim}{m}^{(y)}\right\} \\
\underset{\sim}{m}^{(x)} &= \underset{\sim}{\text{median}}_{i=1,\ldots,3}\left(\underset{\sim}{p}^{(i,x)}\right) \\
&= (58, 75, 75, 94) \\
\underset{\sim}{m}^{(y)} &= \underset{\sim}{\text{median}}_{i=1,\ldots,3}\left(\underset{\sim}{p}^{(i,y)}\right) \\
&= (31, 49, 49, 68).
\end{aligned}$$

And the min-max center (see Figure 1b) would be:

$$\begin{aligned}
\underset{\sim}{Z} &= \left\{\underset{\sim}{z}^{(x)}, \underset{\sim}{z}^{(y)}\right\} \\
\underset{\sim}{z}^{(x)} &= \tfrac{1}{2}\textstyle\sum_{i\in\{1,3\}} \underset{\sim}{p}^{([i],x)} \\
&= (49.667, 64.333, 66, 80.333) \\
\underset{\sim}{z}^{(y)} &= \tfrac{1}{2}\textstyle\sum_{i\in\{1,3\}} \underset{\sim}{p}^{([i],y)} \\
&= (42.667, 57.333, 58.333, 72.667).
\end{aligned}$$

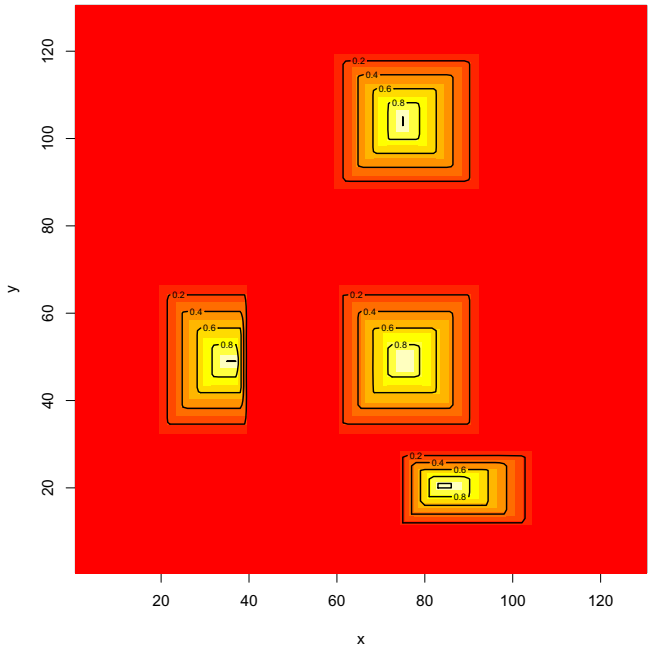
As we can see from Figure 1, using fuzzy numbers to model the demand points is much more closer to the scenarios that geographers and planners face. For example, monocentric urban areas can be represented with bidimensional fuzzy numbers, and the results obtained give planners some flexibility in the final location of the center. Different centers, obeying to different objectives, not only are placed in different locations, but also have different membership function values for the same coordinates, covering different areas.

Now, let's add a constraint space (see Figure 2), that as we have seen is theoretically a fuzzy set. In our case, let's assume a price constraint, that can be modeled as a bidimensional gaussian function, which needs to be intersected with the solutions previously found.

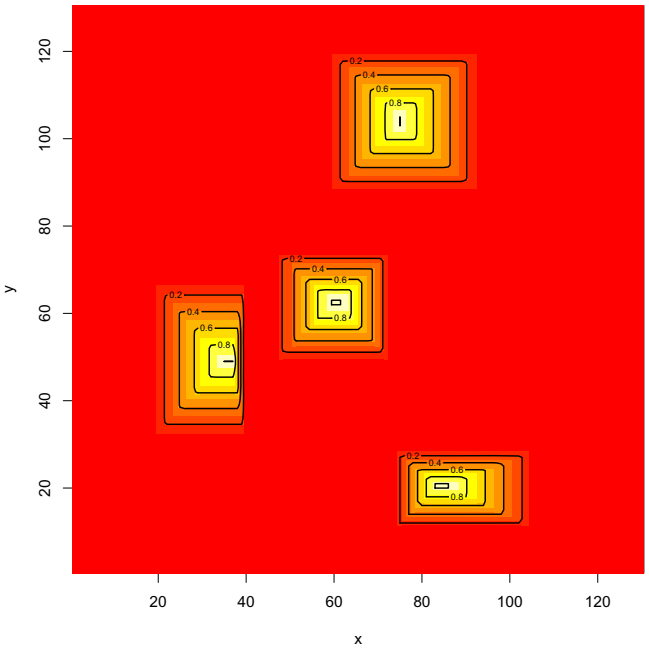
In practice, we will build a matrix that represents the constraint space at a given granularity. The value for each element of this matrix can be given by experts or calculated from the membership function of a given fuzzy set. For each center, it will be needed to build a matrix of the same size as that of the constraint space, using its membership function. This way, the process of finding the intersection is straightforward, using the minimum value of the two matrices (the one for the constraint space and the one for the chosen center) as the value for the constrained solution.

We can see in Figures 3a and 3b how the median and min-max centers are clipped by our constraint space, and as a result the centers are no longer bidimensional fuzzy numbers, only fuzzy sets.

Next, let's assume that the first set of points and the respective fuzzy median and min-max centers belong to the instant $t_0$. We will add two more sets of points for two instants, $t_1$ and $t_2$. For $t_1$ the points are:




(a) Fuzzy median center.




(b) Fuzzy min-max center.

**Fig. 1.** Fuzzy centers

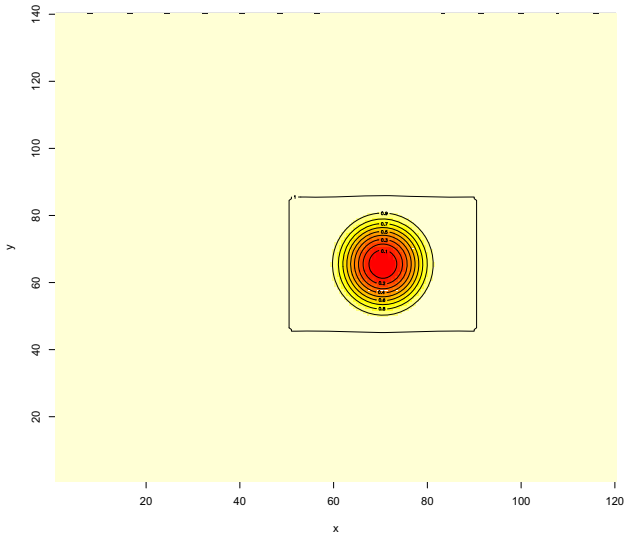



**Fig. 2.** Constraint space

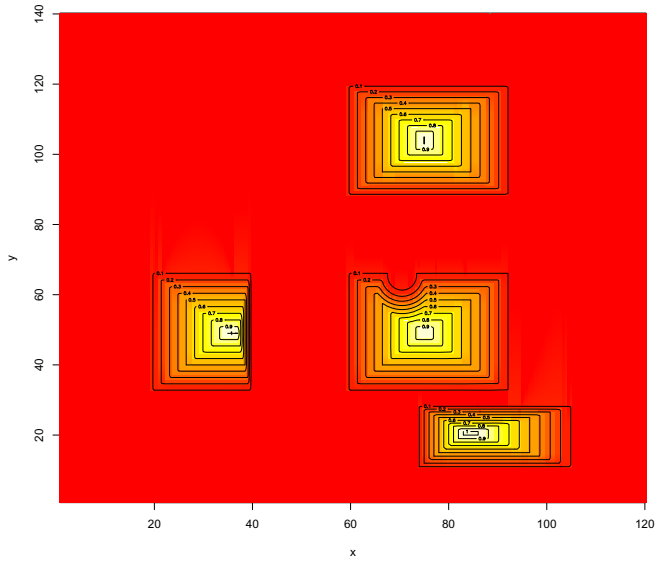
$$
\begin{aligned}
\underbrace{P^{(t_1,1)}} &= \left\{\underbrace{p^{(t_1,1,x)}}, \underbrace{p^{(t_1,1,y)}}\right\} \\
\underbrace{p^{(t_1,1,x)}} &= (20, 30, 33, 45) \\
\underbrace{p^{(t_1,1,y)}} &= (45, 50, 53, 71)
\end{aligned}
$$

$$
\begin{aligned}
\underbrace{P^{(t_1,2)}} &= \left\{\underbrace{p^{(t_1,2,x)}}, \underbrace{p^{(t_1,2,y)}}\right\} \\
\underbrace{p^{(t_1,2,x)}} &= (50, 65, 70, 85) \\
\underbrace{p^{(t_1,2,y)}} &= (75, 95, 97, 110) \\
\underbrace{P^{(t_1,3)}} &= \left\{\underbrace{p^{(t_1,3,x)}}, \underbrace{p^{(t_1,3,y)}}\right\} \\
\underbrace{p^{(t_1,3,x)}} &= (80, 85, 95, 110) \\
\underbrace{p^{(t_1,3,y)}} &= (20, 25, 31, 39).
\end{aligned}
$$

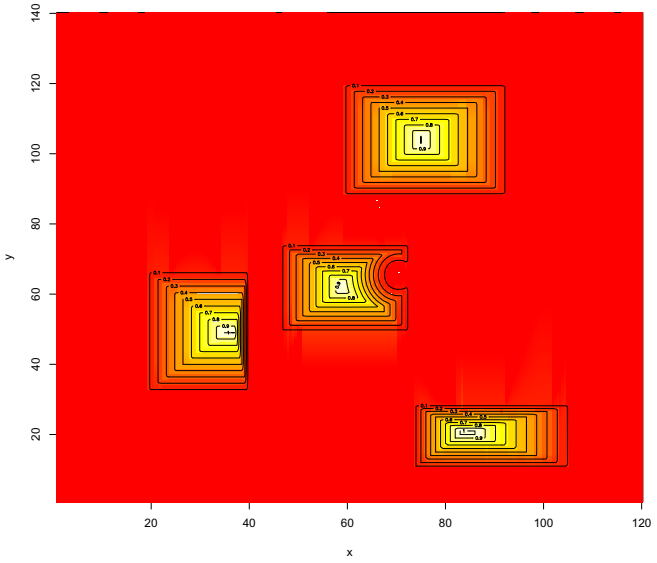
For $t_2$ the points are:

$$
\begin{aligned}
\underbrace{P^{(t_2,1)}} &= \left\{\underbrace{p^{(t_2,1,x)}}, \underbrace{p^{(t_2,1,y)}}\right\} \\
\underbrace{p^{(t_2,1,x)}} &= (24, 28, 32, 47) \\
\underbrace{p^{(t_2,1,y)}} &= (38, 56, 59, 84)
\end{aligned}
$$

$$
\begin{aligned}
\underbrace{P^{(t_2,2)}} &= \left\{\underbrace{p^{(t_2,2,x)}}, \underbrace{p^{(t_2,2,y)}}\right\} \\
\underbrace{p^{(t_2,2,x)}} &= (50, 71, 78, 86) \\
\underbrace{p^{(t_2,2,y)}} &= (90, 112, 115, 119)
\end{aligned}
$$




(a) Fuzzy constrained median center.




(b) Fuzzy constrained min-max center.

**Fig. 3.** Fuzzy constrained centers

$$\begin{aligned} \underset{\sim}{P^{(t_2,3)}} &= \left\{\underset{\sim}{p^{(t_2,3,x)}}, \underset{\sim}{p^{(t_2,3,y)}}\right\} \\ \underset{\sim}{p^{(t_2,3,x)}} &= (69, 91, 96, 112) \\ \underset{\sim}{p^{(t_2,3,y)}} &= (9, 17, 21, 40)\,. \end{aligned}$$

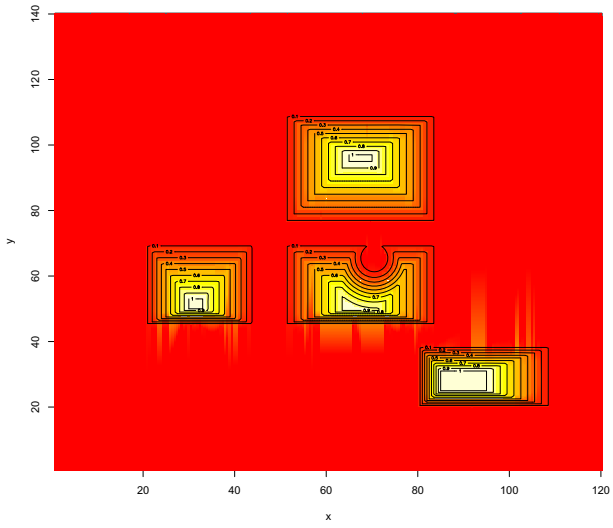
Their respective median centers (see Figure 4) are:

$$\begin{aligned} \underset{\sim}{M^{t_1}} &= \left\{\underset{\sim}{m^{(t_1,x)}}, \underset{\sim}{m^{(t_1,y)}}\right\} \\ \underset{\sim}{m^{(t_1,x)}} &= \underset{\sim}{\text{median}}_{i=1,\ldots,3}\left(\underset{\sim}{p^{(t_1,i,x)}}\right) \\ &= (50, 65, 70, 85) \\ \underset{\sim}{m^{(t_1,y)}} &= \underset{\sim}{\text{median}}_{i=1,\ldots,3}\left(\underset{\sim}{p^{(t_1,i,y)}}\right) \\ &= (45, 50, 53, 71) \\ \underset{\sim}{M^{t_2}} &= \left\{\underset{\sim}{m^{(t_2,x)}}, \underset{\sim}{m^{(t_2,y)}}\right\} \\ \underset{\sim}{m^{(t_2,x)}} &= \underset{\sim}{\text{median}}_{i=1,\ldots,3}\left(\underset{\sim}{p^{(t_2,i,x)}}\right) \\ &= (50, 71, 78, 86) \\ \underset{\sim}{m^{(t_2,y)}} &= \underset{\sim}{\text{median}}_{i=1,\ldots,3}\left(\underset{\sim}{p^{(t_2,i,y)}}\right) \\ &= (38, 56, 59, 84)\,. \end{aligned}$$

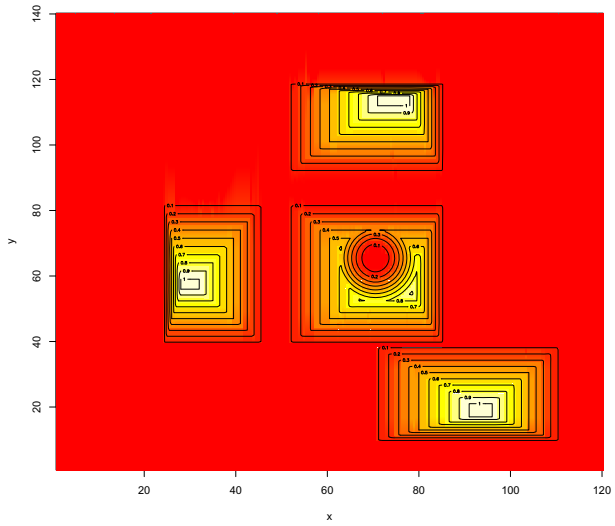
and their respective min-max centers (see Figure 5) are:

$$\begin{aligned} \underset{\sim}{Z^{t_1}} &= \left\{\underset{\sim}{z^{(t_1,x)}}, \underset{\sim}{z^{(t_1,y)}}\right\} \\ \underset{\sim}{z^{(t_1,x)}} &= \tfrac{1}{2}\textstyle\sum_{i\in\{1,3\}} \underset{\sim}{p^{(t_1,[i],x)}} \\ &= (50, 57.5, 64, 77.5) \\ \underset{\sim}{z^{(t_1,y)}} &= \tfrac{1}{2}\textstyle\sum_{i\in\{1,3\}} \underset{\sim}{p^{(t_1,[i],y)}} \\ &= (47.5, 60, 64, 74.5) \\ \underset{\sim}{Z^{t_2}} &= \left\{\underset{\sim}{z^{(t_2,x)}}, \underset{\sim}{z^{(t_2,y)}}\right\} \\ \underset{\sim}{z^{(t_2,x)}} &= \tfrac{1}{2}\textstyle\sum_{i\in\{1,3\}} \underset{\sim}{p^{(t_2,[i],x)}} \\ &= (46.5, 59.5, 64, 79.5) \\ \underset{\sim}{z^{(t_2,y)}} &= \tfrac{1}{2}\textstyle\sum_{i\in\{1,3\}} \underset{\sim}{p^{(t_2,[i],y)}} \\ &= (49.5, 64.5, 68, 79.5)\,. \end{aligned}$$

The intersection $\underset{\sim}{\Theta^{t_2}}$ of the solutions for the three instants $(t_0, t_1, t_2)$ is calculated by the intersection of the matrices that we have found for each constrained solution. As we



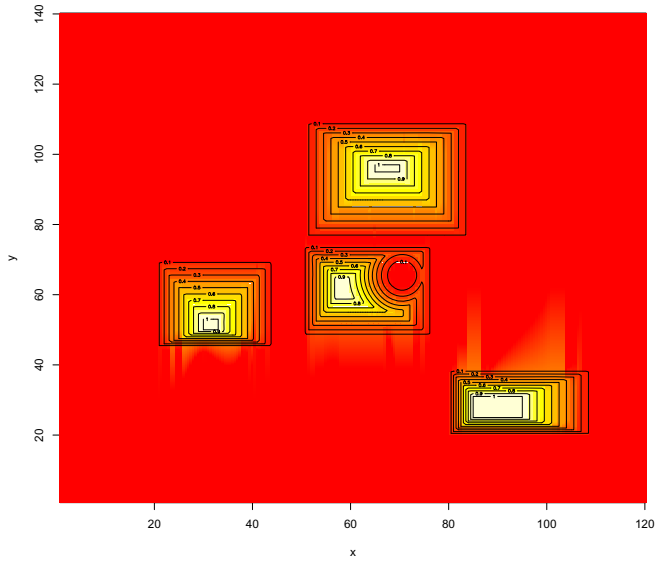

(a) Fuzzy constrained median center for $t_1$.



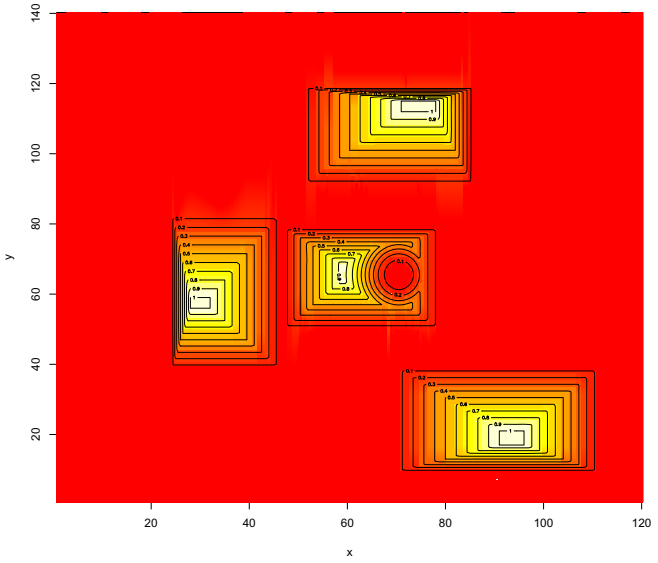

(b) Fuzzy constrained median center for $t_2$.

**Fig. 4.** Fuzzy constrained median centers

see in Figure 6, the evolution of the data points affects in different ways each solution; while the min-max center has a maximum value for its membership function that lies between 0.9 and 1, the median center has only a point above 0.8 and a clear area above 0.7. In this case, we can say that the most possible location using the median center is less durable than using the min-max center.



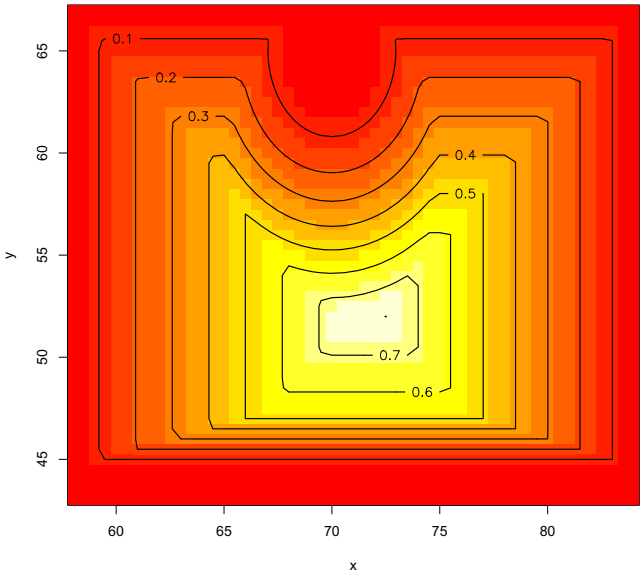

(a) Fuzzy constrained min-max center for $t_1$.



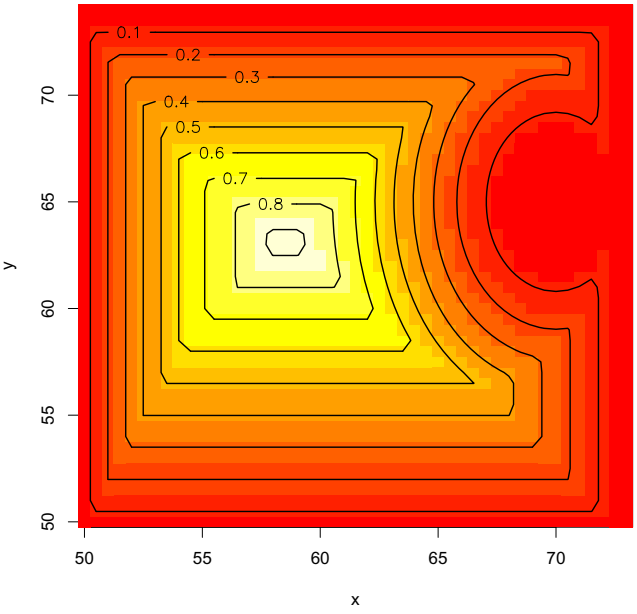

(b) Fuzzy constrained min-max center for $t_2$.

**Fig. 5.** Fuzzy constrained min-max centers

In case the decision maker wanted a minimum value of the membership function, $\mu_{\min} = 0.9$, then the solution for the min-max center would be stable for the period $[t_0, t_2]$, but not the one for the median center. It would be needed to start shedding layers, from the farthest ($t_2$) to the nearest ($t_0$) until a suitable solution is found.

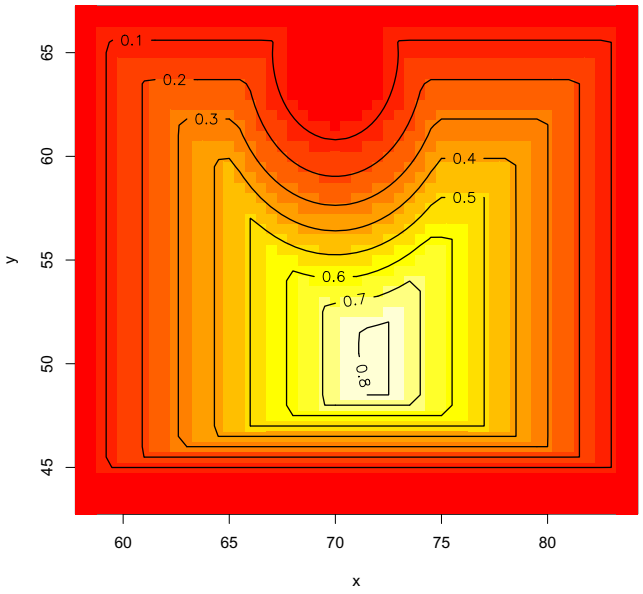



(a) Durable fuzzy constrained median center.

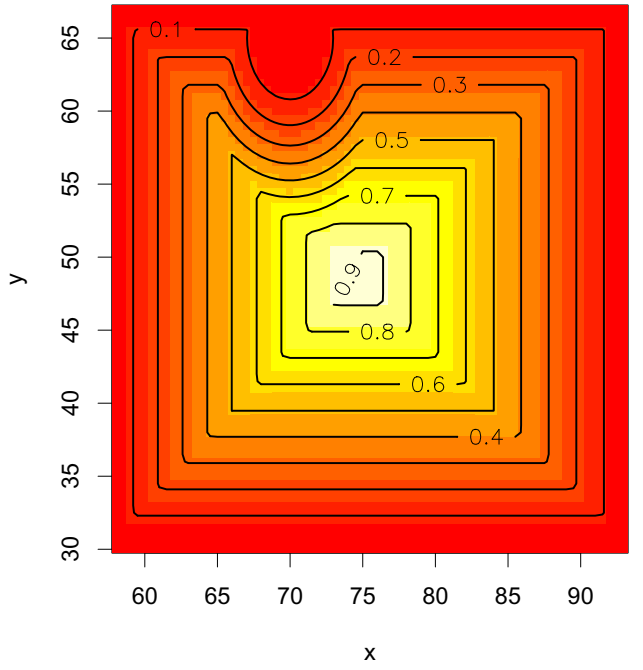



(b) Durable fuzzy constrained min-max center.

**Fig. 6.** Durable fuzzy constrained solutions




a) For $\underset{\sim}{\Theta^{t_1}}$.




b) For $\underset{\sim}{\Theta^{t_0}}$.

**Fig. 7.** Durable fuzzy constrained min-max center

In Figure 7, we can see that $\underset{\sim}{\Theta^{t_1}}$ does not reach a level that complies with the requirements imposed by the decision maker, and only $\underset{\sim}{\Theta^{t_0}}$ does. This means that for our case, a solution based on the median center is not durable at all at a threshold $\mu_{\min} = 0.9$, requiring that either the decision maker fixes a lower value or a new solution is found after $t_0$.

## 7 Conclusions

In this paper we have shown that the results found for the solution of the median center and the min-max center can be extended to fuzzy environments, where both the demand points and the center are modeled with fuzzy numbers. The use of fuzzy numbers is due to the need to incorporate the uncertainty about available information on demand. Not only the data might be vague or subjective, but it could also involve disagreements or lack of confidence in the methodology used in its collection. Therefore, it is necessary to have a solution that, while simply obtained, incorporates this uncertainty.

Fuzzy solutions can also give flexibility to planners on the final location of the center, according to constraints that are not easily modeled. The selected center will have a membership value that reflects its "appropriateness" according to the data.

It has been also shown that it is simple to consider the spatiotemporal analysis of a problem, having a solution that answers to both, spatial constraints and the progressive change of the data points. This added a new level of analysis to the decision maker, who by giving a threshold value to the membership function of the solution can obtain the durability or temporal stability of the exact location selected for the final placement of the demand attention center.

## References

1. Brandeau, M.L., Sainfort, F., Pierskalla, W.P. (eds.): Operations research and health care: A handbook of methods and applications. Kluwer Academic Press, Dordrecht (2004)
2. Canós, M.J., Ivorra, C., Liern, V.: An exact algorithm for the fuzzy *p*-median problem. European Journal of Operational Research 116, 80–86 (1999)
3. Chan, Y.: Location Transport and Land-Use: Modelling Spatial-Temporal Information. Springer, Berlin (2005)
4. Chen, C.-T.: A fuzzy approach to select the location of the distribution center. Fuzzy Sets and Systems 118, 65–73 (2001)
5. Chen, S.-H., Hsieh, C.-H.: Graded mean integration representation of generalized fuzzy number. Journal of Chinese Fuzzy System 5(2), 1–7 (1999)
6. Chen, S.-H., Wang, C.-C.: Fuzzy distance using fuzzy absolute value. In: Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding (2009)
7. Ciligot-Travain, M., Josselin, D.: Impact of the Norm on Optimal Locations. In: Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2009, Part I. LNCS, vol. 5592, pp. 426–441. Springer, Heidelberg (2009)
8. Darzentas, J.: On fuzzy location model. In: Kacprzyk, J., Orlovski, S.A. (eds.) Optimization Models Using Fuzzy Sets and Possibility Theory, pp. 328–341. D. Reidel, Dordrecht (1987)
9. Drezner, Z., Hamacher, H.W. (eds.): Facility location. Applications and theory. Springer, Berlin (2004)

10. Dubois, D., Prade, H.: Fuzzy real algebra: some results. Fuzzy Sets and Systems 2, 327–348 (1979)
11. Griffith, D.A., Amrhein, C.G., Huriot, J.M. (eds.): Econometric advances in spatial modelling and methodology. Essays in honour of Jean Paelinck. Advanced studies in theoretical and applied econometrics, vol. 35 (1998)
12. Hakimi, S.L.: Optimum locations of switching center and the absolute center and medians of a graph. Operations Research 12, 450–459 (1964)
13. Hansen, P., Labbé, M., Peeters, D., Thisse, J.F., Vernon Henderson, J.: Systems of cities and facility locations. In: Fundamentals of Pure and Applied Economics. Harwood Academic Publisher, London (1987)
14. Kaufmann, A., Gupta, M.M.: Introduction to Fuzzy Arithmetic. Van Nostrand Reinhold, New York (1985)
15. Labbé, M., Peeters, D., Thisse, J.F.: Location on networks. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Handbook of Operations Research and Management Science: Network Routing, vol. 8, pp. 551–624. North Holland, Amsterdam (1995)
16. Moreno Pérez, J.A., Marcos Moreno Vega, J., Verdegay, J.L.: Fuzzy location problems on networks. Fuzzy Sets and Systems 142, 393–405 (2004)
17. Nickel, S., Puerto, J.: Location theory. A unified approach. Springer, Berlin (2005)
18. Thomas, I.: Transportation Networks and the Optimal Location of Human Activities, a numerical geography approach. Transport Economics, Management and Policy. Edward Elgar, Northampton (2002)
19. Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
20. Zimmermann, H.-J.: Fuzzy Sets. In: Theory and its Applications, 4th edn. Springer (2005)

# Goodness of Fit Measures and Model Selection in a Fuzzy Least Squares Regression Analysis

Francesco Campobasso and Annarita Fanizzi*

Department of Economics and Mathematics, University of Bari, Bari, Italy
{fracampo,a.fanizzi}@dss.uniba.it

**Abstract.** Market researches and opinion polls usually include customers' responses as verbal labels of sets with vague and uncertain borders. Recently we generalized the estimation procedure of a simple regression model with triangular fuzzy numbers, into the space of which Diamond introduced a metrics, to the case of a multivariate model with an asymmetric intercept also fuzzy.

In this paper we show under what conditions the sum of squares of the dependent variable can be decomposed in exactly the same way as the classical OLS estimation and we propose a fuzzy version of the coefficient of determination, which takes into account the corresponding freedom degrees. Furthermore we introduce a stepwise procedure designed not only to include only one independent variable at a time, but also to eliminate in each iteration that variable whose explanatory contribution is subrogated by the combination of the other ones included after it was.

**Keywords:** Fuzzy least square regression, multivariate generalization, asymmetric fuzzy intercept, total sum of squares, goodness of fit, stepwise selection of independent variables.

## 1 Introduction

Modalities of quantitative variables are commonly given as exact single values, although sometimes they cannot be precise. The imprecision of measuring instruments and the continuous nature of some observations, for example, prevent researcher from obtaining the corresponding true values.

On the other hand qualitative variables are commonly expressed using common linguistic terms, which also represent verbal labels of sets with uncertain borders.

An appropriate way to manage such an uncertainty of observations in dependent model is provided by using fuzzy numbers [1-2].

In 1988 P. M. Diamond [3] introduced a metric into the space of triangular fuzzy numbers and derived the expression of the estimated coefficients in a simple fuzzy regression model. Its adequacy in multiple contexts has been repeatedly demonstrated (for example see [4]).

Starting from a multivariate generalization of such a model, we provided in previous works some results on the decomposition of the total sum of squares of the dependent variable according to Diamond's metric.

* The contribution is the result of joint reflections by the authors, with the following contributions attributed to F. Campobasso (chapters 3, 5 and 6), and to A. Fanizzi (chapters 1, 2 and 4).

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 241–257.
DOI: 10.1007/978-3-642-35638-4_16 

## 2 The Fuzzy Least Square Regression

A triangular fuzzy number $\tilde{X} = (x, x_L, x_R)_T$ for the variable X is characterized by a function $\mu_{\tilde{X}} : X \to [0,1]$, like the one represented in Fig. 1, that expresses the membership degree of any possible value of X to $\tilde{X}$ [5].

The accumulation value x is considered the core of the fuzzy number, while $\underline{\xi} = x_R - x$ and $\overline{\xi} = x - x_L$ are considered the left spread and the right spread respectively.

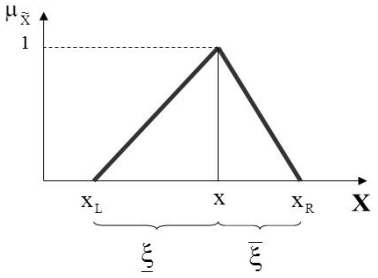



**Fig. 1.** Representation of a triangular fuzzy number

Note that x belongs to $\tilde{X}$ with the highest degree (equal to 1), while the other values included between the left extreme $x_L$ and the right extreme $x_R$ belong to $\tilde{X}$ with a gradually lower degree.

The set of triangular fuzzy numbers is closed under addition: given two triangular fuzzy numbers $\tilde{X} = (x, x_L, x_R)_T$ and $\tilde{Y} = (y, y_L, y_R)_T$, their sum $\tilde{Z}$ is still a triangular fuzzy number $\tilde{Z} = \tilde{X} + \tilde{Y} = (x + y, x_L + y_L, x_R + y_R)_T$. Moreover the opposite of a triangular fuzzy number $\tilde{X} = (x, x_L, x_R)_T$ is $-\tilde{X} = (-x, -x_R, -x_L)_T$.

It follows that, given n fuzzy numbers $\tilde{X}_i = (x_i, x_{Li}, x_{Ri})_T$, i = 1, 2, .., n, their average is $\overline{X} = \frac{\sum \tilde{X}_i}{n} = \left( \frac{\sum x_i}{n}, \frac{\sum x_{Li}}{n}, \frac{\sum x_{Ri}}{n} \right)_T$.

Diamond introduced a metrics into the space of triangular fuzzy numbers; according to this metrics, the squared distance between $\tilde{X}$ and $\tilde{Y}$ is

$$d(\tilde{X}, \tilde{Y})^2 = d\big((x, x_L, x_R)_T, (y, y_L, y_R)_T\big)^2 = (x-y)^2 + (x_L - y_L)^2 + (x_R - y_R)^2.$$

The same Author treated the fuzzy regression model of a dependent variable $\tilde{Y}$ on a single independent variable $\tilde{X}$, which can be written as

$$\tilde{Y} = a + b\tilde{X}, \qquad a, b \in IR,$$

when the intercept a is non-fuzzy, as well as

$$\tilde{Y} = \tilde{A} + b\tilde{X} \quad a, b \in IR,$$

when the intercept $\tilde{A} = (a, a_L, a_R)_T$ is fuzzy, where it is $a_L = a - \underline{\gamma}$, $a_R = a - \overline{\gamma}$ and $\underline{\gamma}, \overline{\gamma} > 0$.

The expression of the corresponding parameters a, $\underline{\gamma}, \overline{\gamma}$ and b is derived from minimizing, with respect to them, the sum $\sum d(\tilde{Y}_i, \tilde{Y}_i^*)^2$ of the squared distances between theoretical and empirical values of the fuzzy dependent variable $\tilde{Y}$ in n observed units.

Such a sum takes different forms according to the signs of the regression coefficient b, as the product of a fuzzy number $\tilde{X} = (x, x_L, x_R)_T$ and a real number depends on whether the latter is positive or negative.

Diamond demonstrated that the optimization problem has a unique solution under certain conditions.

In previous works we provided some theoretical results about the estimates of the regression coefficients and about the decomposition of the sum of squares of the dependent variable [6] in a multiple regression model. In particular we treated the case of a non-fuzzy intercept, as well as the case of a fuzzy intercept, which seems more appropriate [7] for some reasons which will be clearer later.

## 3 A Multivariate Generalization of the Regression Model

### 3.1 A Generalization of the Model Including a Non-fuzzy Intercept

Let's assume to observe a fuzzy dependent variable $\tilde{Y}_i = (y_i, y_{Li}, y_{Ri})_T$ and two fuzzy independent variables, $\tilde{X}_i = (x_i, x_{Li}, x_{Ri})_T$ and $\tilde{Z}_i = (z_i, z_{Li}, z_{Ri})_T$, on a set of n units. The linear regression model is given by

$$\tilde{Y}_i^* = a + b\tilde{X}_i + c\tilde{Z}_i, \qquad I = 1,2, ...,n; \quad a, b, c \in IR.$$

The corresponding parameters a, b, c are determined by minimizing, with respect to them, the sum of Diamond's distances between theoretical and empirical values of the dependent variable

$$\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2 \tag{1}$$

Similarly to what we stated above, such a sum assumes different expressions according to the signs of the regression coefficients b and c. This generates the following four cases:

*Case 1*: *b>0, c>0*

$$\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$

$$= \sum [(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a - bx_{Li} - cz_{Li})^2 + (y_{Ri} - a - bx_{Ri} - cz_{Ri})^2]$$

*Case 2*: *b<0, c>0*

$$\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$

$$= \sum [(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a - bx_{Ri} - cz_{Li})^2 + (y_{Ri} - a - bx_{Li} - cz_{Ri})^2]$$

*Case 3*: *b>0, c<0*

$$\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$

$$= \sum [(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a - bx_{Li} - cz_{Ri})^2 + (y_{Ri} - a - bx_{Ri} - cz_{Li})^2]$$

*Case 4*: *b<0, c<0*

$$\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$

$$= \sum [(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a - bx_{Ri} - cz_{Ri})^2 + (y_{Ri} - a - bx_{Li} - cz_{Li})^2]$$

Let's consider, as an example, case 3 and let's express it in metrical terms. The expression to be minimized is given by

$$\mathbf{G}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + (\mathbf{y_L} - \mathbf{X_L}\boldsymbol{\beta})'(\mathbf{y_L} - \mathbf{X_L}\boldsymbol{\beta}) + (\mathbf{y_R} - \mathbf{X_R}\boldsymbol{\beta})'(\mathbf{y_R} - \mathbf{X_R}\boldsymbol{\beta}) \qquad (2)$$

where

$\mathbf{y} = [y_i]$, is the n-dimensional vector of the cores of the dependent variable;

$\mathbf{y_L} = [y_{Li}]$ and $\mathbf{y_R} = [y_{Ri}]$ are the n-dimensional vectors of the lower extremes and the upper extremes of the dependent variable respectively;

$\mathbf{X}$ is the n×3 matrix of the cores of the independent variables, formed by the vectors $\mathbf{1}$, $\mathbf{x} = [x_i]$, $\mathbf{z} = [z_i]$;

$\mathbf{X_L}$ is the n×3 matrix of the lower bounds of the independent variables, formed by the vectors $\mathbf{1}$, $\mathbf{x_L} = [x_{Li}]$, $\mathbf{z_R} = [z_{Ri}]$;

$\mathbf{X_R}$ is the n×3 matrix of the upper bounds of the independent variables (analogous to $\mathbf{X_L}$), formed by the vectors $\mathbf{1}$, $\mathbf{x_R}$, $\mathbf{z_L}$;

$\boldsymbol{\beta}$ is the vector (a, b, c) '.

The estimates of the regression coefficients are derived from minimizing $G(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, i.e. from seeking the solutions of the system

$$[\mathbf{X}'\mathbf{X} + \mathbf{X_L}'\mathbf{X_L} + \mathbf{X_R}'\mathbf{X_R}]\boldsymbol{\beta} - [\mathbf{y}'\mathbf{X} + \mathbf{y_L}'\mathbf{X_L} + \mathbf{y_R}'\mathbf{X_R}] = 0.$$

In particular we obtain

$$\boldsymbol{\beta} = [\mathbf{X}'\mathbf{X} + \mathbf{X_L}'\mathbf{X_L} + \mathbf{X_R}'\mathbf{X_R}]^{-1}[\mathbf{X}'\mathbf{y} + \mathbf{X_L}'\mathbf{y_L} + \mathbf{X_R}'\mathbf{y_R}].$$

Similarly to OLS estimation procedure, the optimization problem admits a single and finite solution if $[\mathbf{X}'\mathbf{X} + \mathbf{X_L}'\mathbf{X_L} + \mathbf{X_R}'\mathbf{X_R}]$ is invertible and the hessian matrix is definite positive.

The found solution $\boldsymbol{\beta}^* = (a^*, b^*, c^*)'$ is admissible if the signs of the regression coefficients are coherent with the basic assumptions (b >0, c <0).

In the remaining three cases the expression (2) to be minimized is obtained after replacing $\mathbf{z_R}$ by $\mathbf{z_L}$ in $\mathbf{X_L}$ and $\mathbf{z_L}$ by $\mathbf{z_R}$ in $\mathbf{X_R}$ (case 1), $\mathbf{x_L}$ by $\mathbf{x_R}$ and $\mathbf{z_R}$ by $\mathbf{z_L}$ in $\mathbf{X_L}$ and also $\mathbf{x_R}$ by $\mathbf{x_L}$ and $\mathbf{z_L}$ by $\mathbf{z_R}$ in $\mathbf{X_R}$ (case 2), $\mathbf{x_L}$ by $\mathbf{x_R}$ in $\mathbf{X_L}$ and $\mathbf{x_R}$ by $\mathbf{x_L}$ in $\mathbf{X_R}$ (case 4) respectively.

The optimum solution corresponds to that (admissible) one which makes minimum (1) among all.

The generalization of such a procedure to the case of several independent variables is immediate; note that the number of solutions to analyze, in order to identify the optimum one, growths exponentially with the considered number of variables. For example, if the model includes k independent variables, $2^k$ possible cases must be taken into account, which derive from combining the signs of the regression coefficients.

### 3.2 A Generalization of the Model Including a Fuzzy Intercept

Now we analyze an extension of the model with a fuzzy intercept, which seems more appropriate than the non-fuzzy one as it expresses the average value of the dependent variable (which is also *fuzzy*) when the independent variables equal zero.

For this purpose we start from the results obtained by Diamond in the case of the univariate regression model with a fuzzy intercept.

**The Univariate Model.** Let's regress, for example, the dependent variable $\tilde{Y}_i = (y_i, y_{Li}, y_{Ri})_T$ on a single independent variable $\tilde{X}_i = (x_i, x_{Li}, x_{Ri})_T$ in a set of n units. If we consider a symmetric fuzzy intercept $\tilde{A} = (a, a_L, a_R)_T$, where $a_L = a - \gamma$, $a_R = a + \gamma$ and $\gamma > 0$ (if $\gamma = 0$, $\tilde{A}$ would be no more fuzzy), the model assumes the following expression:

$$\tilde{Y}_i^* = \tilde{A} + b\tilde{X}_i \qquad i = 1, 2, ..., n;\ a, b \in \mathbf{R}.$$

The fuzzy regression parameters a, $\gamma$ and b are determined by minimizing, with respect to them, the sum of the squared Diamond's distances between theoretical and empirical fuzzy values of the dependent variable

$$\sum d(\tilde{A} + b\tilde{X}_i, \tilde{Y}_i)^2 .$$

The function to minimize assumes different expressions according to the sign of the regression coefficients b. Supposing that $b > 0$, the estimates of a, b and $\gamma$ are obtained as the solutions a*, b* and $\gamma$* of the system of equations

$$\begin{cases} a\sum(x_i + x_{Li} + x_{Ri}) + \gamma\sum(x_{Ri} - x_{Li}) + b\sum(x_i^2 + x_{Li}^2 + x_{Ri}^2) = \sum(y_i x_i + y_{Li} x_{Li} + y_{Ri} x_{Ri}) \\ 2n\gamma = \sum[(y_{Ri} - y_{Li}) - b(x_{Ri} - x_{Li})] \\ na = \frac{1}{3}\sum\left[y_i + y_{Li} + y_{Ri} - b(x_i + x_{Li} + x_{Ri})\right]. \end{cases}$$

Otherwise, supposing $b<0$, the estimates of a, b and $\gamma$ are obtained as the solutions $a_*$, $b_*$ and $\gamma_*$ of the system of equations

$$\begin{cases} a\sum(x_i + x_{Li} + x_{Ri}) - \gamma\sum(x_{Ri} - x_{Li}) + \\ b\sum(x_i^2 + x_{Li}^2 + x_{Ri}^2) = \sum(x_i y_i + y_{Li} x_{Ri} + y_{Ri} x_{Li}) \\ 2n\gamma = \sum[(y_{Ri} - y_{Li}) + b(x_{Ri} - x_{Li})] \\ na = \frac{1}{3}\sum\left[y_i + y_{Li} + y_{Ri} - b(x_i + x_{Li} + x_{Ri})\right]. \end{cases}$$

As Diamond shows, the solution of this minimization problem exists and is unique if the following conditions occur simultaneously:

1. either $b^* < 0$ or $b_* > 0$;
2. $\sum\left[(x_{Ri} - x_{Li}) - \frac{1}{n}(x_{Ri} - x_{Li})\right]\left[(y_{Ri} - y_{Li}) - \frac{1}{n}(y_{Ri} - y_{Li})\right] \geq 0$;
3. $b^* > b_*$.

**The Multivariate Model.** Now we generalize the regression model with a fuzzy intercept to the case of more than a single independent variable.

Let us assume to regress a dependent variable $\tilde{Y}_i = (y_i, y_{Li}, y_{Ri})_T$ on two independent variables $\tilde{X}_i = (x_i, x_{Li}, x_{Ri})_T$ and $\tilde{Z}_i = (z_i, z_{Li}, z_{Ri})_T$ in a set of n units. If we consider a fuzzy asymmetric intercept $\tilde{A} = (a, a_L, a_R)_T$, where $a_L = a - \underline{\gamma}$, $a_R = a - \overline{\gamma}$ and $\underline{\gamma}$, $\overline{\gamma} > 0$ (if $\underline{\gamma} = \overline{\gamma} = 0$, $\tilde{A}$ would be no more fuzzy), the model assumes the following expression:

$$\tilde{Y}_i^* = \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i, \qquad i = 1, 2, ..., n; \quad a, b, c \in \mathbb{R}.$$

Note that the asymmetric intercept is more appropriate than the symmetric one, in terms of a better adaptation to the data.

The corresponding estimates of the parameters a, $\underline{\gamma}$, $\overline{\gamma}$, b and c are again determined by minimizing, with respect to them, the sum of the squared Diamond's distances between empirical and theoretical values of the dependent variable

$$\sum d(\tilde{Y}_i, \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i)^2 \tag{3}$$

Such a sum assumes different expressions according to the signs of the regression coefficients b and c.

*Case 1*: b>0, c>0

$$\sum d(\tilde{Y}_i, \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$
$$= \sum[(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a_L - bx_{Li} - cz_{Li})^2 + (y_{Ri} - a_R - bx_{Ri} - cz_{Ri})^2]$$

*Case 2*: b<0, c>0

$$\sum d(\tilde{Y}_i, \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$
$$= \sum[(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a_L - bx_{Ri} - cz_{Li})^2 + (y_{Ri} - a_R - bx_{Li} - cz_{Ri})^2]$$

*Case 3*: b>0, c<0

$$\sum d(\tilde{Y}_i, \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$

$$= \sum[(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a_L - bx_{Li} - cz_{Ri})^2 + (y_{Ri} - a_R - bx_{Ri} - cz_{Li})^2]$$

*Case 4*: b<0, c<0

$$\sum d(\tilde{Y}_i, \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$

$$= \sum[(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a_L - bx_{Ri} - cz_{Ri})^2 + (y_{Ri} - a_R - bx_{Li} - cz_{Li})^2]$$

Let's consider, as an example, case 3 and let's express it in matricial terms. The expression to be minimized is given by

$$\mathbf{G(\beta) = (y - X\beta)'(y - X\beta) + (y_L - X_L\beta)'(y_L - X_L\beta) + (y_R - X_R\beta)'(y_R - X_R\beta)} \quad (4)$$

where

$\mathbf{y} = [y_i]$, is the n-dimensional the vector of cores of the dependent variable;

$\mathbf{y_L} = [\, y_{Li}\,]$ and $\mathbf{y_R} = [\, y_{Ri}\,]$ are the n-dimensional vectors of the lower extremes and the upper extremes of the dependent variable respectively;

$\mathbf{X}$ is the n×5 matrix of the cores of the independent variables, formed by the vectors $\mathbf{1}$, $\mathbf{x} = [\, x_i\,]$, $\mathbf{z} = [\, z_i\,]$ and two vectors $\mathbf{0}$;

$\mathbf{X_L}$ is the n×5 matrix of lower bounds of the independent variables, formed by the vectors $\mathbf{1}$, $\mathbf{x_L} = [\, x_{Li}\,]$, $\mathbf{z_R} = [\, z_{Ri}\,]$ and $\mathbf{-1, 0}$;

$\mathbf{X_R}$ is the n×5 matrix of the upper bounds of the independent variables (analogous to $\mathbf{X_L}$), formed by the vectors $\mathbf{1}$, $\mathbf{x_R}$, $\mathbf{z_L}$ and $\mathbf{0}$, $\mathbf{1}$;

$\boldsymbol{\beta}$ is the vector $(a, b, c, \underline{\gamma}, \overline{\gamma})'$.

The estimates of the regression coefficients are derived from minimizing $G(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, i.e. from seeking the solutions of the system

$$[\mathbf{X'X} + \mathbf{X_L'X_L} + \mathbf{X_R'X_R}]\boldsymbol{\beta} - [\mathbf{y'X} + \mathbf{y_L'X_L} + \mathbf{y_R'X_R}] = 0.$$

In particular we obtain

$$\boldsymbol{\beta} = [\mathbf{X'X} + \mathbf{X_L'X_L} + \mathbf{X_R'X_R}]^{-1}[\,\mathbf{X'y} + \mathbf{X_L'y_L} + \mathbf{X_R'y_R}].$$

Similarly to OLS estimation procedure, the optimization problem admits a single and finite solution if $[\mathbf{X'X} + \mathbf{X_L'X_L} + \mathbf{X_R'X_R}]$ is invertible and the hessian matrix is definite positive.

The found solution $\boldsymbol{\beta}^* = (a^*, b^*, c^*, \underline{\gamma}^*, \overline{\gamma}^*)'$ is admissible if the signs of the regression coefficients are coherent with the basic assumptions ($b > 0$, $c < 0$ and $\underline{\gamma}, \overline{\gamma} > 0$).

In the remaining three cases the expression (4) to be minimized is obtained after appropriately replacing the vectors of the left and right extremes in the matrices as described above, according to the case considered. The optimum solution corresponds to that (admissible) one which makes minimum (3) among all.

When the intercept is symmetric, we estimate one parameter less than the previous model, because the spreads left and right coincide [8]. Note that the matrices $\mathbf{X}$, $\mathbf{X_L}$ and $\mathbf{X_R}$, relative to independent variables, and the vector of parameters $\beta$ change their expression than before. In particular we have that

$\mathbf{X}$ is the n×4 matrix of the cores of the independent variables, formed by the vectors $\mathbf{1}$, $\mathbf{x} = [\, x_i \,]$, $\mathbf{z} = [\, z_i \,]$ and $\mathbf{0}$;

$\mathbf{X_L}$ is the n×4 matrix of the lower bounds of the independent variables, formed by the vectors $\mathbf{1}$, $\mathbf{x_L} = [\, x_{Li} \,]$, $\mathbf{z_R} = [\, z_{Ri} \,]$ and $\mathbf{-1}$;

$\mathbf{X_R}$ is the n×4 matrix of the upper bounds of the independent variables (analogous to $\mathbf{X_L}$), formed by the vectors $\mathbf{1}$, $\mathbf{x_R}$, $\mathbf{z_L}$ and $\mathbf{1}$;

$\beta$ is the vector $(a, b, c, \gamma)'$.

# 4 Decomposition of the Total Sum of Squares of the Dependent Variable

In this section two important theoretical results will be demonstrated: the first one regards the inequality between theoretical and empirical averages of the fuzzy dependent variable (unlike in the classical OLS estimation procedure); the second one regards the decomposition of the total sum of squares of the dependent variable, which involves other two additive components besides the regression and the residual sum of squares.

## 4.1 The Model Including a Non-fuzzy Intercept

Let's consider, as an example, the sum of Diamond's distances between theoretical and empirical values of the dependent variable in the *case 3*:

$$\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$
$$= \sum [(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a - bx_{Li} - cz_{Ri})^2 + (y_{Ri} - a - bx_{Ri} - cz_{Li})^2] .$$

Setting equal to 0 the derivate of $\sum d(\tilde{Y}_i, a + b\tilde{X}_i + c\tilde{Z}_i)^2$ respect to a, b and c, we can obtain the following system of equations:

$$\begin{cases} -2\sum[(y_i - a - bx_i - cz_i) + (y_{Li} - a - bx_{Li} - cz_{Ri}) + (y_{Ri} - a - bx_{Ri} - cz_{Li})] = 0 \\ -2\sum[(y_i - a - bx_i - cz_i)x_i + (y_{Li} - a - bx_{Li} - cz_{Ri})x_{Li} + (y_{Ri} - a - bx_{Ri} - cz_{Li})x_{Ri}] = 0 \\ -2\sum[(y_i - a - bx_i - cz_i)z_i + (y_{Li} - a - bx_{Li} - cz_{Ri})z_{Ri} + (y_{Ri} - a - bx_{Ri} - cz_{Li})z_{Li}] = 0 \end{cases}$$

Such a system can be written as

$$\begin{cases} \sum(a + bx_i + cz_i) + \sum(a + bx_{Li} + cz_{Ri}) + \sum(a + bx_{Ri} + cz_{Li}) = \sum(y_i + y_{Li} + y_{Ri}) \\ \sum(a + bx_i + cz_i)x_i + \sum(a + bx_{Li} + cz_{Ri})x_{Li} + \sum(a + bx_{Ri} + cz_{Li})x_{Ri} = \sum y_i x_i + \sum y_{Li} x_{Li} + \sum y_{Ri} x_{Ri} \\ \sum(a + bx_i + cz_i)z_i + \sum(a + bx_{Li} + cz_{Ri})z_{Ri} + \sum(a + bx_{Ri} + cz_{Li})z_{Li} = \sum y_i z_i + \sum y_{Li} z_{Ri} + \sum y_{Ri} z_{Li} \end{cases}$$

Recalling that the theoretical values of the fuzzy dependent variable are $y_i^* = a + bx_i + cz_i$ $y_{Li}^* = a + bx_{Li} + cz_{Ri}$ and $y_{Ri}^* = a + bx_{Ri} + cz_{Li}$, we obtain

$$\begin{cases} \sum(y_i^* + y_{Li}^* + y_{Ri}^*) = \sum(y_i + y_{Li} + y_{Ri}) \\ \sum y_i^* x_i + y_{Li}^* x_{Li} + y_{Ri}^* x_{Ri} = \sum y_i x_i + y_{Li} x_{Li} + y_{Ri} x_{Ri} \\ \sum y_i^* z_i + y_{Li}^* z_{Ri} + y_{Ri}^* z_{Li} = \sum y_i z_i + y_{Li} z_{Ri} + y_{Ri} z_{Li}^L \end{cases} \quad (5)$$

The first equation of the system (5) shows that the total sum of the lower extremes, the cores and the upper extremes of the theoretical values of the dependent variable coincides with the same amount referred to the empirical values. This equation does not allow us to state that theoretical and empirical averages of the fuzzy dependent variable coincide.

Let's examine how the total sum of squares of dependent variable

$$\text{TotSS} = \sum[(y_i - \bar{y})^2 + (y_{Li} - \bar{y}_L)^2 + (y_{Ri} - \bar{y}_R)^2]$$

can be decomposed according to Diamond's metric.

Adding and subtracting the corresponding theoretical value within all the squares and developing them, the total sum of squares can be expressed as:

$$\text{TotSS} = \sum[(y_i - y_i^* + y_i^* - \bar{y})^2 + (y_{Li} - y_{Li}^* + y_{Li}^* - \bar{y}_L)^2 + (y_{Ri} - y_{Ri}^* + y_{Ri}^* - \bar{y}_R)^2] =$$
$$= \sum[(y_i - y_i^*)^2 + (y_i^* - \bar{y})^2 + 2(y_i - y_i^*)(y_i^* - \bar{y}) + (y_{Li} - y_{Li}^*)^2 + (y_{Li}^* - \bar{y}_L)^2 + 2(y_{Li} - y_{Li}^*)(y_{Li}^* - \bar{y}_L) +$$
$$+ (y_{Ri} - y_{Ri}^*)^2 + (y_{Ri}^* - \bar{y}_R)^2 + 2(y_{Ri} - y_{Ri}^*)(y_{Ri}^* - \bar{y}_R)].$$

Adding and subtracting the theoretical average values of the lower extremes, of the cores and of the upper extremes of the dependent variable within all the squares and developing them, the previous expression becomes

$$\text{TotSS} = \sum[(y_i - y_i^*)^2 + (y_i^* - \bar{y}^* + \bar{y}^* - \bar{y})^2 + 2(y_i - y_i^*)(y_i^* - \bar{y}) + (y_{Li} - y_{Li}^*)^2 + (y_{Li}^* - \bar{y}_L^* + \bar{y}_L^* - \bar{y}_L)^2 +$$
$$+ 2(y_{Li} - y_{Li}^*)(y_{Li}^* - \bar{y}_L) + (y_{Ri} - y_{Ri}^*)^2 + (y_{Ri}^* - \bar{y}_R^* + \bar{y}_R^* - \bar{y}_R)^2 + 2(y_{Ri} - y_{Ri}^*)(y_{Ri}^* - \bar{y}_R)] =$$
$$= \sum[(y_i - y_i^*)^2 + (y_i^* - \bar{y}^*)^2 + (\bar{y}^* - \bar{y})^2 + 2(y_i^* - \bar{y}^*)(\bar{y}^* - \bar{y}) + 2(y_i - y_i^*)(y_i^* - \bar{y}) + (y_{Li} - y_{Li}^*)^2 + (y_{Li}^* - \bar{y}_L^*)^2 +$$
$$+ (\bar{y}_L^* - \bar{y}_L)^2 + 2(y_{Li}^* - \bar{y}_L^*)(\bar{y}_L^* - \bar{y}_L) + 2(y_{Li} - y_{Li}^*)(y_{Li}^* - \bar{y}_L) + (y_{Ri} - y_{Ri}^*)^2 + (y_{Ri}^* - \bar{y}_R^*)^2 + (\bar{y}_R^* - \bar{y}_R)^2 +$$
$$+ 2(y_{Ri}^* - \bar{y}_R^*)(\bar{y}_R^* - \bar{y}^R) + 2(y_{Ri} - y_{Ri}^*)(y_{Ri}^* - \bar{y}_R)]$$

where:

$\text{Reg SS} = \sum d(\tilde{Y}_i^*, \bar{Y})^2 = \sum[(y_i^* - \bar{y})^2 + (y_{Li}^* - \bar{y}_L)^2 + (y_{Ri}^* - \bar{y}_R)^2]$ represents the regression sum of squares,

$\text{Res SS} = \sum d(\tilde{Y}_i, \tilde{Y}_i^*)^2 = \sum[(y_i - y_i^*)^2 + (y_{Li} - y_{Li}^*)^2 + (y_{Ri} - y_{Ri}^*)^2]$ represents the residual sum of squares and

$nd(\bar{Y}^*, \bar{Y})^2 = n[(\bar{y}^* - \bar{y})^2 + (\bar{y}_L^* - \bar{y}_L)^2 + (\bar{y}_R^* - \bar{y}_R)^2]$ represents the distance between theoretical and empirical average values of the dependent variable.

Synthetically the expression of Tot SS can be written as:

$$\text{Tot SS} = \text{Reg SS} + \text{Res SS} + \text{nd}(\overline{Y}, \overline{Y}^*)^2 + \eta$$

where:

$$\eta = 2\sum[(y_i^* - \bar{y}^*)(\bar{y}^* - \bar{y}) + (y_{Li}^* - \bar{y}_L^*)(\bar{y}_L^* - \bar{y}_L) + (y_{Ri}^* - \bar{y}_R^*)(\bar{y}_R^* - \bar{y}_R)] + 2\sum[(y_i - y_i^*)(y_i^* - \bar{y}) +$$
$$+ (y_{Li} - y_{Li}^*)(y_{Li}^* - \bar{y}_L) + (y_{Ri} - y_{Ri}^*)(y_{Ri}^* - \bar{y}_R)].$$

As the sums of the differences between each component and its average equal zero, then it is

$$\sum[(y_i^* - \bar{y}^*)(\bar{y}^* - \bar{y}) + (y_{Li}^* - \bar{y}_L^*)(\bar{y}_L^* - \bar{y}_L) + (y_{Ri}^* - \bar{y}_R^*)(\bar{y}_R^* - \bar{y}_R)] = 0$$

and the amount $\eta$ is reduced to

$$\eta = 2\sum[(y_i - y_i^*)(y_i^* - \bar{y}) + (y_{Li} - y_{Li}^*)(y_{Li}^* - \bar{y}_L) + (y_{Ri} - y_{Ri}^*)(y_{Ri}^* - \bar{y}_R)] =$$
$$= 2\sum[(y_i - y_i^*)y_i^* - (y_i - y_i^*)\bar{y} + (y_{Li} - y_{Li}^*)y_{Li}^* - (y_{Li} - y_{Li}^*)\bar{y}_L + (y_{Ri} - y_{Ri}^*)y_{Ri}^* - (y_{Ri} - y_{Ri}^*)\bar{y}_R].$$

Moreover, as it is $y_i^* = a + bx_i + cz_i$, $y_{Li}^* = a + bx_{Li} + cz_{Ri}$ and $y_{Ri}^* = a + bx_{Ri} + cz_{Li}$, it is also

$$2\sum(y_i - y_i^*)y_i^* + 2\sum(y_{Li} - y_{Li}^*)y_{Li}^* + 2\sum(y_{Ri} - y_{-Ri}^*)y_{Ri}^* = 0.$$

By replacing expressions of the theoretical values in the latter equation, we obtain

$$\eta = 2\sum[a(y_i + y_{Li} + y_{Ri}) - a(y_i^* + y_{Li}^* + y_{Ri}^*) + b(y_i x_i + y_{Li}x_{Li} + y_{Ri}x_{Ri}) +$$
$$- b(y_i^* x_i + y_{Li}^* x_{Li} + y_{Ri}^* x_{Ri}) + c(y_i z_i + y_{Li}z_{Ri} + y_{Ri}z_{Li}) - c(y_i^* x_i + y_{Li}^* x_{Li} + y_{Ri}^* x_{Ri})] +$$
$$- 2\sum[(y_i - y_i^*)\bar{y} + (y_{Li} - y_{Li}^*)\bar{y}_L + (y_{Ri} - y_{Ri}^*)\bar{y}_R].$$

According to the condition (5), the last expression can be reduced to

$$\eta = -2\sum[(y_i - y_i^*)\bar{y} + (y_{Li} - y_{Li}^*)\bar{y}_L + (y_{Ri} - y_{Ri}^*)\bar{y}_R].$$

Note that, if the residual sum of squares equals zero, also $\eta$ and $d(\overline{Y}, \overline{Y}^*)^2$ equal zero, because theoretical and empirical average values of the dependent variable coincide for each observation.

Therefore:

- if the regression sum of squares equals zero, then the model has no forecasting capability, because the sum of the components of the i-th theoretical value equals the sum of the components of the empirical average value (i = 1 ,..., n). Actually it is for each i

$$\sum(y_i^* + y_{Li}^* + y_{Ri}^*) = \sum(y_i + y_{Li} + y_{Ri}) \Rightarrow y_i^* + y_{Li}^* + y_{Ri}^* = \bar{y} + \bar{y}_L + \bar{y}_R;$$

- if the residual sum of squares equals zero, the relationship between the dependent variable and the independent ones is well represented by the estimated model. In this case, the total sum of squares is entirely explained by the regression sum of squares.

### 4.2 The Model Including a Fuzzy Intercept

Let's consider, as an example of the model with a fuzzy intercept, the sum of Diamond's distances between theoretical and empirical values of the dependent variable in the *case 3*:

$$\sum d(\tilde{Y}_i, \tilde{A} + b\tilde{X}_i + c\tilde{Z}_i)^2 =$$
$$= \sum[(y_i - a - bx_i - cz_i)^2 + (y_{Li} - a_L - bx_{Li} - cz_{Ri})^2 + (y_{Ri} - a_R - bx_{Ri} - cz_{Li})^2]$$

By minimizing such a sum with respect to a, $\underline{\gamma}$, $\overline{\gamma}$, b, and c (remember that $a_L = a - \underline{\gamma}$ and $a_R = a + \overline{\gamma}$), we can obtain the following system of equations

$$\begin{cases} -2\sum[(y_i - a - bx_i - cz_i) + (y_{Li} - a + \underline{\gamma} - bx_{Li} - cz_{Ri}) + (y_{Ri} - a - \overline{\gamma} - bx_{Ri} - cz_{Li})] = 0 \\ -2\sum[(y_i - a - bx_i - cz_i)x_i + (y_{Li} - a + \underline{\gamma} - bx_{Li} - cz_{Ri})x_{Li} + (y_{Ri} - a - \overline{\gamma} - bx_{Ri} - cz_{Li})x_{Ri}] = 0 \\ -2\sum[(y_i - a - bx_i - cz_i)z_i + (y_{Li} - a + \underline{\gamma} - bx_{Li} - cz_{Ri})z_{Ri} + (y_{Ri} - a - \overline{\gamma} - bx_{Ri} - cz_{Li})z_{Li}] = 0 \\ 2\sum(y_{Li} - a + \underline{\gamma} - bx_{Li} - cz_{Ri}) = 0 \\ -2(y_{Ri} - a - \overline{\gamma} - bx_{Ri} - cz_{Li}) = 0 \end{cases}$$

Such a system can be written as

$$\begin{cases} \sum(a + bx_i + cz_i) + \sum(a - \underline{\gamma} + bx_{Li} + cz_{Ri}) + \sum(a + \overline{\gamma} + bx_{Ri} + cz_{Li}) = \sum(y_i + y_{Li} + y_{Ri}) \\ \sum(a + bx_i + cz_i)x_i + \sum(a - \underline{\gamma} + bx_{Li} + cz_{Ri})x_{Li} + \sum(a + \overline{\gamma} + bx_{Ri} + cz_{Li})x_{Ri} = \sum y_i x_i + \sum y_{Li} x_{Li} + \sum y_{Ri} x_{Ri} \\ \sum(a + bx_i + cz_i)z_i + \sum(a - \underline{\gamma} + bx_{Li} + cz_{Ri})z_{Ri} + \sum(a + \overline{\gamma} + bx_{Ri} + cz_{Li})z_{Li} = \sum y_i z_i + \sum y_{Li} z_{Ri} + \sum y_{Ri} z_{Li} \\ \sum(a - \underline{\gamma} + bx_{Li} + cz_{Ri}) = \sum y_{Li} \\ \sum(a + \overline{\gamma} + bx_{Ri} + cz_{Li}) = \sum y_{Ri} \end{cases}$$

Recalling that the theoretical values of the fuzzy dependent variable are $y_i^* = a + bx_i + cz_i$, $y_{Li}^* = a - \underline{\gamma} + bx_{Li} + cz_{Ri}$ and $y_{Ri}^* = a + \overline{\gamma} + bx_{Ri} + cz_{Li}$ respectively, we obtain

$$\begin{cases} \sum(y_i^* + y_{Li}^* + y_{Ri}^*) = \sum(y_i + y_{Li} + y_{Ri}) \\ \sum y_i^* x_i + \sum y_{Li}^* x_{Li} + \sum y_{Ri}^* x_{Ri} = \sum y_i x_i + \sum y_{Li} x_{Li} + \sum y_{Ri} x_{Ri} \\ \sum y_i^* z_i + \sum y_{Li}^* z_{Ri} + \sum y_{Ri}^* z_{Li} = \sum y_i z_i + \sum y_{Li} z_{Ri} + \sum y_{Ri} z_{Li} \\ \sum y_{Li}^* = \sum y_{Li} \\ \sum y_{Ri}^* = \sum y_{Ri} \end{cases} \quad (6)$$

The first equation shows that the total sum of the cores and the extremes of the theoretical values of the dependent variable coincides with the same amount referred to the empirical values. The combination of the first equation with the last two allows us to state that theoretical and empirical values of the average fuzzy dependent variable coincide, like it happens in the classic OLS estimation procedure.

Let's examine how the total sum of squares of dependent variable can be decomposed according to Diamond's metric:

$$\text{TotSS}=\sum[(y_i-\bar{y})^2+(y_{Li}-\bar{y}_L)^2+(y_{Ri}-\bar{y}_R)^2].$$

Adding and subtracting the corresponding theoretical value within all the each squares and developing them, the total sum of squares can be expressed as:

$$\begin{aligned}\text{Tot SS}&=\sum[(y_i-y_i^*+y_i^*-\bar{y})^2+(y_{Li}-y_{Li}^*+y_{Li}^*-\bar{y}_L)^2+(y_{Ri}-y_{Ri}^*+y_{Ri}^*-\bar{y}_R)^2]=\\&=\sum[(y_i-y_i^*)^2+(y_i^*-\bar{y})^2+2(y_i-y_i^*)(y_i^*-\bar{y})+(y_{Li}-y_{Li}^*)^2+(y_{Li}^*-\bar{y}_L)^2+2(y_{Li}-y_{Li}^*)(y_{Li}^*-\bar{y}_L)+\\&+(y_{Ri}-y_{Ri}^*)^2+(y_{Ri}^*-\bar{y}_R)^2+2(y_{Ri}-y_{Ri}^*)(y_{Ri}^*-\bar{y}_R)].\end{aligned}$$

Adding and subtracting the theoretical average values of the lower extremes, of the cores and of the upper extremes of the dependent variable within all the squares and developing them, the previous expression becomes

$$\begin{aligned}\text{TotSS}&=\sum[(y_i-y_i^*)^2+(y_i^*-\bar{y}^*+\bar{y}^*-\bar{y})^2+2(y_i-y_i^*)(y_i^*-\bar{y})+(y_{Li}-y_{Li}^*)^2+(y_{Li}^*-\bar{y}_L^*+\bar{y}_L^*-\bar{y}_L)^2+\\&+2(y_{Li}-y_{Li}^*)(y_{Li}^*-\bar{y}_L)+(y_{Ri}-y_{Ri}^*)^2+(y_{Ri}^*-\bar{y}_R^*+\bar{y}_R^*-\bar{y}_R)^2+2(y_{Ri}-y_{Ri}^*)(y_{Ri}^*-\bar{y}_R)]=\\&=\sum[(y_i-y_i^*)^2+(y_i^*-\bar{y}^*)^2+(\bar{y}^*-\bar{y})^2+2(y_i^*-\bar{y}^*)(\bar{y}^*-\bar{y})+2(y_i-y_i^*)(y_i^*-\bar{y})+(y_{Li}-y_{Li}^*)^2+\\&+(y_{Li}^*-\bar{y}_L^*)^2+(\bar{y}_L^*-\bar{y}_L)^2+2(y_{Li}^*-\bar{y}_L^*)(\bar{y}_L^*-\bar{y}_L)+2(y_{Li}-y_{Li}^*)(y_{Li}^*-\bar{y}_L)+(y_{Ri}-y_{Ri}^*)^2+\\&+(y_{Ri}^*-\bar{y}_R^*)^2+(\bar{y}_R^*-\bar{y}_R)^2+2(y_{Ri}^*-\bar{y}_R^*)(\bar{y}_R^*-\bar{y}^R)+2(y_{Ri}-y_{Ri}^*)(y_{Ri}^*-\bar{y}_R)]\end{aligned}$$

where:

$\text{RegSS}=\sum d(\tilde{Y}_i^*,\bar{Y})^2=\sum[(y_i^*-\bar{y})^2+(y_{Li}^*-\bar{y}_L)^2+(y_{Ri}^*-\bar{y}_R)^2]$ represents the regression sum of squares, while

$\text{Res SS}=\sum d(\tilde{Y}_i,\tilde{Y}_i^*)^2=\sum[(y_i-y_i^*)^2+(y_{Li}-y_{Li}^*)^2+(y_{Ri}-y_{Ri}^*)^2]$ represents the residual sum of squares. Moreover, according to the conditions (6), it is

$$\begin{aligned}&\sum[(\bar{y}^*-\bar{y})^2+2(y_i-y_i^*)(y_i^*-\bar{y})+2(y_i^*-\bar{y}^*)(\bar{y}^*-\bar{y})+(\bar{y}_L^*-\bar{y}_L)^2+2(y_{Li}-y_{Li}^*)(y_{Li}^*-\bar{y}_L)+\\&+2(y_{Li}^*-\bar{y}_L^*)(\bar{y}_L^*-\bar{y}_L)+(\bar{y}_R^*-\bar{y}_R)^2+2(y_{Ri}-y_{Ri}^*)(y_{Ri}^*-\bar{y}_R)+2(y_{Ri}^*-\bar{y}_R^*)(\bar{y}_R^*-\bar{y}_R)]=0\end{aligned}$$

Therefore the expression of the total sum of squares of the dependent variable can be reduced to

$$\text{Tot SS}=\text{Reg SS}+\text{Res SS}.$$

Ultimately the total sum of squares consists only of two addends (the regression and the residual sum of squares) like in the classic OLS estimation procedure, when the intercept has the same form as the dependent variable.

Note that, when the intercept has not the same form as the dependent variable, theoretical and empirical average values of the latter do not coincide in correspondence of each observation; rather the total sum of the lower extremes, the cores and the upper extremes of the theoretical values coincides with the same amount referred to the empirical values:

$$\begin{cases} \sum(y_i^* + y_{Li}^* + y_{Ri}^*) = \sum(y_i + y_{Li} + y_{Ri}) \\ \sum y_i^* x_i + \sum y_{Li}^* x_{Li} + \sum y_{Ri} x_{Ri} = \sum y_i x_i + \sum y_{Li} x_{Li} + \sum y_{Ri} x_{Ri} \\ \sum y_i^* z_i + \sum y_{Li}^* z_{Ri} + \sum y_{Ri}^* z_{Li} = \sum y_i z_i + \sum y_{Li} z_{Ri} + \sum y_{Ri} z_{Li} \\ \sum(y_{Ri}^* - y_{Ri}) = \sum(y_{Li}^* - y_{Li}) \end{cases}$$

In this case the total sum of squares of the dependent variable consists of two other components in addition to the regression and the residual sum of squares: the first one is residual in nature and is characterized by an uncertain sign, the second one is equal to n times the distance between theoretical and empirical average values of the dependent variable.

### 4.3 A Fuzzy Model Fit Index

We have just demonstrated that the total sum of squares of the dependent variable consists only of two addends (the regression and the residual sum of squares), when the intercept is fuzzy asymmetric. This is because theoretical and empirical average values of the dependent variable coincide and, therefore, both the total sum of squares and the regression sum of squares can be expressed in terms of distance between empirical values and their averages.

Under these circumstances, the greater the regression sum of squares the better the model fits the data.

In order to assess the goodness of fit of the regression model, we propose the following index, for simplicity called Fuzzy Fit Index (FFI):

$$FFI = 1 - \frac{ResSS}{TotSS} = 1 - \frac{\sum d(\tilde{Y}_i, \tilde{Y}_i^*)^2}{\sum d(\tilde{Y}_i, \overline{\tilde{Y}})^2}$$

where $\overline{\tilde{Y}}^* = (\bar{y}^*, \bar{y}_L^*, \bar{y}_R^*)_T$ and $\overline{\tilde{Y}} = (\bar{y}, \bar{y}_L, \bar{y}_R)_T$ denote the fuzzy theoretical average and the fuzzy empirical average of the dependent variable respectively.

The more this index is next to 1, the smaller the residual sum of squares is and the better the model fits the observed data.

In order to compare models that explain the same dependent variable by means of a different number of independent variables, it is appropriate to refer to an index that takes into account the corresponding degrees of freedom (closely linked at this number). As in the classic model, an increase in FFI does not necessarily mean that the new independent variable contributes significantly to explain $\tilde{Y}$; any excess in measuring the fit of the model can be corrected by deflating FFI for a term which increases with the number of independent variables included in the equation.

The proposed version of the adjusted FFI is

$$\overline{FFI} = 1 - \left( \frac{ResSS}{TotSS} \cdot \frac{n-1}{n-p-1} \right)$$

which increases only if the increase in FFI (i.e. in the regression sum of squares) exceeds the penalty induced by having one more independent variable in the model, and decreases otherwise.

## 5 A Stepwise Procedure to Select Independent Variables

The selection of the most significant independent variables presents greater difficulties from a computational point of view in a *fuzzy* regression model than in the classic one.

The fuzzy approach makes the search for optimal combinations of the starting variables more complex, as the total number of the potential hyperplanes to be tested increases exponentially with the considered number p of independent variables. In fact, for each subset of $q \leq p$ variables, $2^q$ different hyperplanes result from all combinations of the signs assumed by the corresponding regression coefficients.

In order to avoid complications related to the above checks, recently we proposed [9] a *stepwise forward* identification procedure. At each iteration such a procedure inserted a variable in the regression equation, according to two fundamental criteria: the significance of its contribution, measured by the relative increase of the total sum of squares in the dependent variable, and its originality, i.e. the ability to introduce information into the equation which other variables have not already introduced (assessed in terms of correlation with the latter).

In this work, we introduce a *stepwise* procedure which enables us to find the optimal combination of the independent variables not only by including only one of them at a time, but also by eliminating in each iteration that variable, whose explanatory contribution is subrogated by the combination of the other ones included after it was (Fig. 2).

This procedure drastically reduces the number of models to be estimated in order to identify the best one among them. Let's examine how it works in detail, starting from the forward selection.

After identifying an initial simple regression model (in which $\tilde{X}_{(1)}$ presents the highest correlation with $\tilde{Y}$), in each successive iteration we select the variable less correlated with those already present, provided that it significantly explains the total sum of squares of the model. In other words, focusing on the q.th step (q=2,3,...p), $\tilde{X}_{(q)}$ is candidate to be also included in the equation if its contribution is original with respect to the previous q-1 variables.

Such a contribution is evaluated by measuring the so called *tolerance* $T_q=1-\overline{FFI}_{q;1,2,...,q-1}$, in which $\overline{FFI}_{q;1,2,...,q-1}$ represents the share of variability of $\tilde{X}_{(q)}$ explained by $\tilde{X}_{(1)}, \tilde{X}_{(2)}, ..., \tilde{X}_{(q-1)}$. The tolerance ranges between 0 and 1, depending on the degree of linear correlation between $\tilde{X}_{(q)}$ and the other variables; therefore, only if $T_q$ exceeds a threshold identified between 0 and 1, $\tilde{X}_{(q)}$ is candidate to become part of the model.

Note that a high value of the threshold allows us to select very original variables, but it can also stop the process since from the initial steps; on the contrary, a low value allows us to select a greater number of variables, although more correlated to each other. In any case, if none of the variables not yet included in the equation proves significantly its originality, the selection process would stop.

The opportunity of actually introducing the selected variable $\tilde{X}_{(q)}$ into the equation is now evaluated in terms of its explanatory contribution. In particular such a contribution is measured as the increase in the adjusted Fuzzy Fit Index of the model due to the entry submitted for consideration, i.e. as $\overline{FFI}_{y;1,2,...,q} - \overline{FFI}_{y;1,2,...,q-1}$ (where the two

terms of the subtraction represent the proportion of the sum of squares of $\tilde{Y}$ explained by the model including and not including $\tilde{X}_{(q)}$ respectively).

The selected variable ends up being introduced into the equation if the correspondent increase in the adjusted Fuzzy Fit Index is higher than an arbitrary threshold value. The higher such an arbitrary value is, the easier the procedure inhibits the entry of new independent variables, whose explanatory contribution is not that relevant.

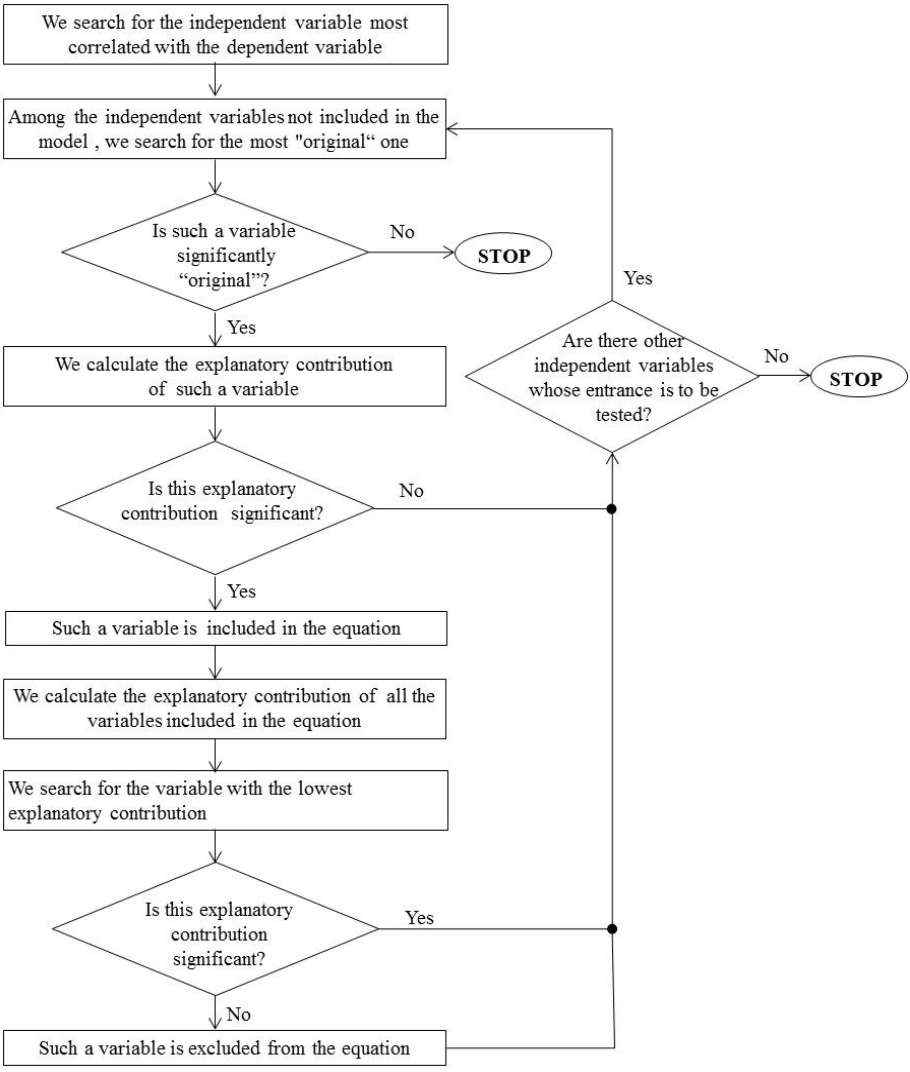



**Fig. 2.** Iterative steps of the procedure

If the explanatory contribution of $\tilde{X}_{(q)}$ is not significant, we pass to consider the inclusion of the remaining candidate variables on the basis of the same criterion. In any case, the selection procedure stops when none of them contributes significantly to explain the sum of squares of $\tilde{Y}$.

The proposed procedure provides also the possibility of eliminating in each iteration one of the variables already included in the equation. For example, once $\tilde{X}_{(q)}$ is inserted, the explanatory contribution of every $\tilde{X}_{(i)}$ (i = 1, 2, ..q-1) is still valued as the reduction of the adjusted Fuzzy Fit Index caused by the elimination of $\tilde{X}_{(i)}$ from the model, i.e. as $\overline{FFI}_{y;1,2,...,q} - \overline{FFI}_{y;1,2,...,q\,(-i)}$ (where the two terms of the subtraction represent the proportion of the sum of squares of $\tilde{Y}$ explained by the model excluding and not excluding $\tilde{X}_{(i)}$ respectively).

The variable which shows the smallest reduction is excluded from the equation if such a reduction does not exceed an arbitrary threshold value. This would happen if the explanatory contribution of the variable to be discarded was subrogated by the combination of the independent variables introduced after it was.

In conclusion it is worth noting that the threshold value of the variation in the adjusted Fuzzy Fit Index could be lower in the case of the forward selection, where priority is given to the role of tolerance, than in the case of the backward selection. This allows us to include into the equation a greater number of variables which otherwise would play no role, maintaining the opportunity to evaluate their exclusion at a later time.

A possible limitation of the proposed procedure is the fact that variables, once eliminated, cannot be nominated to enter the equation in next steps; in other words the selection of variables is limited to those never become part of the model. Instead it is plausible that a discarded variable can go to make a significant contribution in explaining the model later in the procedure (in correspondence with a different combination of variables from the one generating its output).

Future works will aim to verify if it is possible to remove this problem without incurring a less optimal procedure from other points of view.

## 6 Conclusions

In this work we first explicit the expressions of the estimated parameters of a multivariate fuzzy regression model with a fuzzy asymmetric intercept. Such an intercept is more appropriate than a non-fuzzy on, as it is to be estimated by the average value of the dependent variable (which is also fuzzy) when the independent variables equal zero.

Moreover we verify that the sum of squares of the dependent variable consists simply in the regression and the residual sum of squares, like it happens in the classic OLS estimation procedure, only when the intercept is fuzzy asymmetric triangular. Conversely, when the intercept is symmetric (both fuzzy and not), the analysis of the forecasting capability of the model is more difficult. This happens because of the presence of two additional components of the sum of squares: the first one which is related to the difference between the theoretical and the empirical average values of

the dependent variable, the second one which is residual in nature and is characterized by an uncertain sign.

The selection of the most significant independent variables in a fuzzy regression model presents computational difficulties due to the large number of potential hyperplanes to be tested. We propose to overcome such difficulties through a *stepwise* procedure, based on a fuzzy version of the $R^2$ index, which enables us to find the optimal combination of the starting variables.

In each step a single variable is included between the starting ones, according to two basic criteria: its explanatory contribution to the model and its originality with respect to the other variables already included in the model.

The procedure provides the possibility of eliminating at each iteration variables already included in the model, whose explanatory contribution is subrogated by the combination of the independent variables introduced later.

Some improvements to the model mainly concern the shape of the membership function of fuzzy membership different from the triangular one.

## References

1. Kao, C., Chyu, C.L.: Least-squares estimates in fuzzy regression analysis. European Journal of Operational Research (2003)
2. Takemura, K.: Fuzzy least squares regression analysis for social judgment study. Journal of Advanced Intelligent Computing and Intelligent Informatics (2005)
3. Diamond, P.M.: Fuzzy Least Square. Information Sciences (1988)
4. Bilancia, M., Campobasso, F., Fanizzi, A.: The pricing of risky securities in a Fuzzy Least Square Regression model. In: Locarek-Junge, H., Weihs, C. (eds.) Studies in Classification, Data Analysis and Knowledge Organization - Classification as a Tool for Research, pp. 639–646. Springer, Heidelberg (2010)
5. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
6. Campobasso, F., Fanizzi, A., Tarantini, M.: Some results on a multivariate generalization of the Fuzzy Least Square Regression. In: Proceedings of the International Conference on Fuzzy Computation, Madeira - Portugal (2009)
7. Montrone, S., Campobasso, F., Perchinunno, P., Fanizzi, A.: An Analysis of Poverty in Italy through a Fuzzy Regression Model. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011, Part I. LNCS, vol. 6782, pp. 342–355. Springer, Heidelberg (2011)
8. Campobasso, F., Fanizzi, A.: A Fuzzy Approach to the Least Squares Regression Model With a Symmetric Fuzzy Intercept. In: Proceedings of the 14th Applied Stochastic Model and Data Analysis Conference, Roma (2011)
9. Campobasso, F., Fanizzi, A.: A stepwise procedure to select variables in a fuzzy least square regression model. In: Proceedings of the International Conference on International Conference on Fuzzy Computation Theory and Applications, pp. 417–426 (2011)

# Part III
# Neural Computation Theory and Applications

# Multilayer Perceptron Learning Utilizing Reducibility Mapping

Seiya Satoh and Ryohei Nakano

Chubu University, 1200 Matsumoto-cho, Kasugai, 487-8501 Japan
tp11015-0513@sti.chubu.ac.jp, nakano@cs.chubu.ac.jp

**Abstract.** In the search space of MLP($J$), multilayer perceptron having J hidden units, there exist flat areas called singular regions created by applying reducibility mapping to the optimal solution of MLP($J-1$). Since such singular regions cause serious slowdown for learning, a learning method for avoiding singular regions has been aspired. However, such avoiding does not guarantee the quality of the final solutions. This paper proposes a new learning method which does not avoid but makes good use of singular regions to stably and successively find solutions excellent enough for MLP($J$). The potential of the method is shown by our experiments using artificial and real data sets.

**Keywords:** Multilayer perceptron, Learning method, Reducibility mapping, Singular region, Polynomial network.

## 1 Introduction

It is known in MLP learning that an MLP($J$) parameter subspace having the same input-output map as an optimal solution of MLP($J-1$) forms a flat area called a singular region, and the singular region causes the stagnation of learning [4]. Natural gradient [1,2] was once proposed to avoid such stagnation of MLP learning, but even the method may get stuck in singular regions and is not guaranteed to find an excellent solution. Recently an alternative constructive method has been proposed [6].

It is also known that many useful statistical models, such as MLP, Gaussian mixtures, and HMM, are singular models having singular regions where parameters are nonidentifiable. While theoretical research has been vigorously done to clarify mathematical structure and characteristics of singular models [10], experimental research is rather insufficient to fully support the theories.

In MLP parameter space there are a number of local minima forming equivalence class [12]. Even if we exclude equivalence class, it is widely believed that there still remain local minima [3]. When we adopt an exponential function as an activation function [8], there surely exist local minima due to the expressive power of polynomials. In XOR problem, however, it was proved there is no strict local minima [5]. Thus, since we have no clear knowledge of MLP parameter space, we run a learning method repeatedly changing initial weights to find an excellent solution.

This paper proposes a new learning method which does not avoid but makes good use of singular regions to stably and successively find excellent solutions. The method starts with an MLP having one hidden unit and then gradually increases the number of hidden units until the intended number. When it increases the number of hidden units

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 261–275.
DOI: 10.1007/978-3-642-35638-4_17 

from $J-1$ to $J$, it utilizes an optimum of MLP($J-1$) to form two kinds of singular regions in MLP($J$) parameter space. Each singular region forms a line, and the learning method can descend in the MLP($J$) parameter space since most points along the line are saddles. Thus, we can always find a solution of MLP($J$) better than the optimum of MLP($J-1$). Our method is evaluated by the experiments for sigmoidal and polynomial-type MLPs using artificial and real data sets.

## 2 Singular Regions of Multilayer Perceptron

This section explains that an optimum of MLP($J-1$) is used to form singular regions in MLP($J$) parameter space [4]. This result is universal in the sense that it does not depend on the choice of an error function or an activation function.

Consider the following MLP($J$), MLP having $J$ hidden units and one output unit. MLP($J$) having $\boldsymbol{\theta}_J$ outputs $f_J(\boldsymbol{x};\boldsymbol{\theta}_J)$ for input $\boldsymbol{x}$. Here $g(h)$ denotes an activation function, and $\boldsymbol{\theta}_J = \{w_0, w_j, \boldsymbol{w}_j, j = 1, \cdots, J\}$, where $\boldsymbol{w}_j = (w_{jk})$.

$$f_J(\boldsymbol{x};\boldsymbol{\theta}_J) = w_0 + \sum_{j=1}^{J} w_j z_j, \qquad z_j \equiv g(\boldsymbol{w}_j^T \boldsymbol{x}) \tag{1}$$

Let input vector $\boldsymbol{x} = (x_k)$ be $K$-dimensional. Given training data $\{(\boldsymbol{x}^\mu, y^\mu), \mu = 1, \cdots, N\}$, we want to find the parameter vector $\boldsymbol{\theta}_J$ which minimizes the following error function.

$$E_J = \frac{1}{2} \sum_{\mu=1}^{N} (f_J^\mu - y^\mu)^2, \qquad \text{where } f_J^\mu \equiv f_J(\boldsymbol{x}^\mu;\boldsymbol{\theta}_J) \tag{2}$$

At the same time we consider the following MLP($J-1$) having $J-1$ hidden units, where $\boldsymbol{\theta}_{J-1} = \{u_0, u_j, \boldsymbol{u}_j, j = 2, \cdots, J\}$.

$$f_{J-1}(\boldsymbol{x};\boldsymbol{\theta}_{J-1}) = u_0 + \sum_{j=2}^{J} u_j v_j, \qquad v_j \equiv g(\boldsymbol{u}_j^T \boldsymbol{x}) \tag{3}$$

The error function of MLP($J-1$) is defined as follows.

$$E_{J-1}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{\mu=1}^{N} (f_{J-1}^\mu - y^\mu)^2, \qquad \text{where } f_{J-1}^\mu \equiv f_{J-1}(\boldsymbol{x}^\mu;\boldsymbol{\theta}_{J-1}) \tag{4}$$

Let $\widehat{\boldsymbol{\theta}}_{J-1}$ denote a critical point of MLP($J-1$), which satisfies the following

$$\frac{\partial E_{J-1}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0}.$$

The necessary conditions for the critical point of MLP ($J-1$) are shown below. Here $j = 2, \cdots, J$ and $v_j^\mu \equiv g(\boldsymbol{u}_j^T \boldsymbol{x}^\mu)$.

$$\frac{\partial E_{J-1}}{\partial u_0} = \sum_{\mu} (f_{J-1}^\mu - y^\mu) = 0$$

$$\frac{\partial E_{J-1}}{\partial u_j} = \sum_{\mu} (f_{J-1}^{\mu} - y^{\mu})\, v_j^{\mu} = 0$$

$$\frac{\partial E_{J-1}}{\partial \boldsymbol{u}_j} = u_j \sum_{\mu} (f_{J-1}^{\mu} - y^{\mu})\, g'(\boldsymbol{u}_j^T \boldsymbol{x}^{\mu})\, \boldsymbol{x}^{\mu} = \boldsymbol{0}$$

Now we consider the following three reducibility mappings $\alpha$, $\beta$, $\gamma$, and let $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$, $\widehat{\boldsymbol{\Theta}}_J^{\beta}$, and $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ denote the regions obtained by applying these three mappings to an optimum $\widehat{\boldsymbol{\theta}}_{J-1} = \{\widehat{u}_0, \widehat{u}_j, \widehat{\boldsymbol{u}}_j, j = 2, \cdots, J\}$ of MLP($J-1$).

$$\widehat{\boldsymbol{\theta}}_{J-1} \xrightarrow{\alpha} \widehat{\boldsymbol{\Theta}}_J^{\alpha}, \quad \widehat{\boldsymbol{\theta}}_{J-1} \xrightarrow{\beta} \widehat{\boldsymbol{\Theta}}_J^{\beta}, \quad \widehat{\boldsymbol{\theta}}_{J-1} \xrightarrow{\gamma} \widehat{\boldsymbol{\Theta}}_J^{\gamma}$$

$$\begin{aligned} \widehat{\boldsymbol{\Theta}}_J^{\alpha} \equiv \{\boldsymbol{\theta}_J |\ & w_0 = \widehat{u}_0,\ w_1 = 0, \\ & w_j = \widehat{u}_j, \boldsymbol{w}_j = \widehat{\boldsymbol{u}}_j, j = 2, \cdots, J\} \end{aligned} \tag{5}$$

$$\begin{aligned} \widehat{\boldsymbol{\Theta}}_J^{\beta} \equiv \{\boldsymbol{\theta}_J |\ & w_0 + w_1 g(w_{10}) = \widehat{u}_0,\ \boldsymbol{w}_1 = [w_{10}, 0, \cdots, 0]^T, \\ & w_j = \widehat{u}_j, \boldsymbol{w}_j = \widehat{\boldsymbol{u}}_j, j = 2, \cdots, J\} \end{aligned} \tag{6}$$

$$\begin{aligned} \widehat{\boldsymbol{\Theta}}_J^{\gamma} \equiv \{\boldsymbol{\theta}_J |\ & w_0 = \widehat{u}_0,\ w_1 + w_2 = \widehat{u}_2,\ \boldsymbol{w}_1 = \boldsymbol{w}_2 = \widehat{\boldsymbol{u}}_2, \\ & w_j = \widehat{u}_j, \boldsymbol{w}_j = \widehat{\boldsymbol{u}}_j, j = 3, \cdots, J\} \end{aligned} \tag{7}$$

(a) region $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$ is $(K+1)$-dimensional since free vector $\boldsymbol{w}_1$ is $(K+1)$-dimensional.
(b) region $\widehat{\boldsymbol{\Theta}}_J^{\beta}$ is two-dimensional since all we have to do is to satisfy the following

$$w_0 + w_1\, g(w_{10}) = \widehat{u}_0.$$

(c) region $\widehat{\boldsymbol{\Theta}}_{\gamma}$ is a line since we have only to satisfy the following

$$w_1 + w_2 = \widehat{u}_2.$$

Here we review a critical point where the gradient $\partial E / \partial \boldsymbol{\theta}$ of an error function $E(\boldsymbol{\theta})$ gets zero. In the context of minimization, a critical point is classified into a local minimum and a saddle. A critical point $\boldsymbol{\theta}_0$ is classified as a local minimum when any point $\boldsymbol{\theta}$ in its neighborhood satisfies $E(\boldsymbol{\theta}_0) \leq E(\boldsymbol{\theta})$, otherwise is classified as a saddle.

In this paper we classify a local minimum into a wok-bottom and a gutter. A wok-bottom $\boldsymbol{\theta}_0$ is a strict local minimum where any point $\boldsymbol{\theta}$ in its neighborhood satisfies $E(\boldsymbol{\theta}_0) < E(\boldsymbol{\theta})$, and a gutter is a continuous subspace where any points $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ in the subspace satisfy $E(\boldsymbol{\theta}_1) = E(\boldsymbol{\theta}_2)$ or $E(\boldsymbol{\theta}_1) \approx E(\boldsymbol{\theta}_2)$, and any point $\boldsymbol{\theta}$ in its neighborhood satisfies $E(\boldsymbol{\theta}_1) < E(\boldsymbol{\theta})$.

The necessary conditions for the critical point of MLP ($J$) are shown below. Here $j = , \cdots, J$ and $z_j^{\mu} \equiv g(\boldsymbol{w}_j^T \boldsymbol{x}^{\mu})$.

$$\frac{\partial E_J}{\partial w_0} = \sum_{\mu} (f_J^{\mu} - y^{\mu}) = 0$$

$$\frac{\partial E_J}{\partial w_1} = \sum_{\mu} (f_J^{\mu} - y^{\mu})\, z_1^{\mu} = 0$$

$$\frac{\partial E_J}{\partial w_j} = \sum_{\mu} (f_J^{\mu} - y^{\mu})\, z_j^{\mu} = 0,$$
$$\frac{\partial E_J}{\partial \boldsymbol{w}_1} = w_1 \sum_{\mu} (f_J^{\mu} - y^{\mu})\, g'(\boldsymbol{w}_1^T \boldsymbol{x}^{\mu})\, \boldsymbol{x}^{\mu} = \boldsymbol{0}$$
$$\frac{\partial E_J}{\partial \boldsymbol{w}_j} = w_j \sum_{\mu} (f_J^{\mu} - y^{\mu})\, g'(\boldsymbol{w}_j^T \boldsymbol{x}^{\mu})\, \boldsymbol{x}^{\mu} = \boldsymbol{0}$$

Then we check if regions $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$, $\widehat{\boldsymbol{\Theta}}_J^{\beta}$, and $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ satisfy these necessary conditions. Note that in these regions we have $f_J^{\mu} = f_{J-1}^{\mu}$ and $v_j^{\mu} = z_j^{\mu},\ j = 2, \cdots, J$. Thus, we see that the first, third, and fifth equations hold, and the second and fourth equations are needed to check.

(a) In region $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$, since weight vector $\boldsymbol{w}_1$ is free, the output of the first hidden unit $z_1^{\mu}$ is free, which means it is not guaranteed that the second and fourth equations hold. Thus, $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$ is not a singular region in general.
(b) In region $\widehat{\boldsymbol{\Theta}}_J^{\beta}$, since $z_1^{\mu} (= g(w_{10}))$ is independent on $\mu$, the second equation can be reduced to the first one, and holds. However, the fourth equation does not hold in general unless $w_1 = 0$. Thus, the following area included in both $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$ and $\widehat{\boldsymbol{\Theta}}_J^{\beta}$ forms a singular region where $w_{10}$ is free. This region is called $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ and reducibility mapping from $\widehat{\boldsymbol{\theta}}_{J-1}$ to $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ is called $\alpha\beta$.

$$\begin{aligned} &w_0 = \widehat{u}_0, \quad w_1 = 0, \quad \boldsymbol{w}_1 = [w_{10}, 0, \cdots, 0]^T \\ &w_j = \widehat{u}_j,\ \boldsymbol{w}_j = \widehat{\boldsymbol{u}}_j,\ \ j = 2, \cdots, J \end{aligned} \tag{8}$$

(c) In region $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$, since $z_1^{\mu} = v_2^{\mu}$, the second and fourth equations hold. Namely, $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ is a singular region. Here we have one degree of freedom since we only have the following restriction

$$w_1 + w_2 = \widehat{u}_2 \tag{9}$$

## 3 SSF1.1 (Singularity Stairs Following, ver. 1.1) Method

This section proposes an extended version of SSF [9], which makes good use of singular regions $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ and $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ of MLP. Although the original SSF [9], called here SSF1.0, used only $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$, SSF1.1 uses not only $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ but also $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$. By searching $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ as well, we can examine the whole singular regions, may find better solutions and will get more insight into MLP search space.

Here we explain how to search these two singular regions. It is rather easy to search these regions since either region has at most one degree of freedom and most points in the region are saddles [4], which means we surely find a solution of MLP($J$) better than an optimum of MLP($J-1$).

We search $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ changing initial points repeatedly in the form of interpolation or extrapolation of eq.(9). On the other hand we search $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ using the Hessian $\boldsymbol{H}(=\partial^2 E/\partial \boldsymbol{w}\partial \boldsymbol{w}^T)$. Otherwise we cannot move the search point since the region is completely flat. We pick up a negative eigen value of $\boldsymbol{H}$ and select its eigen vector $\boldsymbol{v}$ and its negative vector $-\boldsymbol{v}$ as two search directions. The appropriate step length is decided using line search called golden section [7].

The procedure of SSF1.1 is described below. It searches the space by ascending singularity stairs one by one, beginning with MLP($J$=1) and gradually increasing $J$ until the intended largest number $J_{max}$. Here $w_0^{(J)}$, $w_j^{(J)}$, and $\boldsymbol{w}_j^{(J)}$ are weights of MLP($J$). Compared with the original SSF1.0 [9], step 2-1-2 is added to incorporate reducibility mapping $\alpha\beta$.

**SSF1.1 (Singularity Stairs Following, ver. 1.1)**

**(step 1)** Find the excellent solution of MLP($J$=1) by repeating the learning changing initial weights. let the best result be $\widehat{w}_0^{(1)}$, $\widehat{w}_1^{(1)}$, and $\widehat{\boldsymbol{w}}_1^{(1)}$. Then $J \leftarrow 1$.
**(step 2)** While $J < J_{max}$, repeat the following to get MLP($J$+1) from MLP($J$).
**(step 2-1)** If there are more than one hidden units in MLP($J$), repeat the following for each hidden unit $m(=1,\cdots,J)$ to split.
**(step 2-1-1)** Initialize weights of MLP($J$+1) using reducibility mapping $\gamma$:
$w_j^{(J+1)} \leftarrow \widehat{w}_j^{(J)},\ j \in \{0,1,\cdots,J\} \setminus \{m\}, \quad \boldsymbol{w}_j^{(J+1)} \leftarrow \widehat{\boldsymbol{w}}_j^{(J)},\ j=1,\cdots,J$
$\boldsymbol{w}_{J+1}^{(J+1)} \leftarrow \widehat{\boldsymbol{w}}_m^{(J)}$.
Initialize $w_m^{(J+1)}$ and $w_{J+1}^{(J+1)}$ many times while satisfying the restriction $w_m^{(J+1)} + w_{J+1}^{(J+1)} = \widehat{w}_m^{(J)}$ in the form of interpolation or extrapolation. Then perform MLP($J$+1) learning for each initialization and keep the best as the best of $\gamma$ for the hidden unit $m$ to split.
**(step 2-1-2)** Initialize weights of MLP($J$+1) using reducibility mapping $\alpha\beta$:
$w_j^{(J+1)} \leftarrow \widehat{w}_j^{(J)},\ j=0,1,\cdots,J, \qquad \boldsymbol{w}_j^{(J+1)} \leftarrow \widehat{\boldsymbol{w}}_j^{(J)},\ j=1,\cdots,J$
$w_{J+1}^{(J+1)} = 0, \qquad \boldsymbol{w}_{J+1}^{(J+1)} \leftarrow [0,0,\cdots,0]^T$.
Pick up a negative eigen value of $\boldsymbol{H}$ and select its eigen vector $\boldsymbol{v}$ and $-\boldsymbol{v}$ as two search directions. Find the appropriate step length using golden section. Then perform MLP($J$+1) learning and keep the best as the best of $\alpha\beta$ for $m$.
**(step 2-1-3)** As the best MLP($J+1$) for $m$, select the better from the best of $\gamma$ for $m$ and the best of $\alpha\beta$ for $m$.
**(step 2-2)** Among the best MLPs($J$+1) for different $m$, select the true best and let the weights be $\widehat{w}_0^{(J+1)}, \widehat{w}_j^{(J+1)}, \widehat{\boldsymbol{w}}_j^{(J+1)}$, $j$=1, $\cdots$, $J$+1. Then $J \leftarrow J$+1.

Now we claim the following, which will be evaluated in the next experiments.

(1) Compared with the existing methods such as BP, Newton's method, quasi-Newton methods, SSF1.1 finds excellent solutions of MLP($J$) with much higher probabilities.
(2) The excellent solution of MLP($J$) is obtained one after another for $J=1,\cdots,J_{max}$. These excellent solutions can be used for model selection. SSF1.1 guarantees that the solution of MLP($J$+1) is better than that of MLP($J$) since SSF1.1 descends in

MLP($J$+1) search space from the singular region corresponding to the excellent solution of MLP($J$). Such monotonic decrease of training error is not guaranteed for the existing methods.

## 4 Experiments

We evaluate the proposed SSF1.1 for sigmoidal and polynomial-type MLPs using artificial and real data sets. Activation functions $g(h)$ in eq. (1) for sigmoidal and polynomial-type MLPs are $g(h) = 1/(1 + e^{-h})$ and $g(h) = exp(h)$ respectively. Then the output of polynomial-type MLP is written as follows.

$$f_J = \sum_{j=0}^{J} w_j z_j, \qquad z_j = \exp\left(\sum_{k=1}^{K} w_{jk} \ln x_k\right) \tag{10}$$

The above can be rewritten as below, representing a multivariate polynomial [8].

$$f_J = \sum_{j=0}^{J} w_j z_j, \qquad z_j = \prod_{k=1}^{K} (x_k)^{w_{jk}} \tag{11}$$

In performing SSF1.1, since we have to move in singular flat regions, we employ weak weight decay where penalty coefficient $\rho = 0.001$. As a learning engine of SSF1.1 we use a kind of quasi-Newton method called BPQ [11] since any first-order method is too slow to converge. As for step 2-1-1 of SSF1.1, the weight initialization of reducibility mapping $\gamma$ is repeated 50 times each for interpolation and extrapolation.

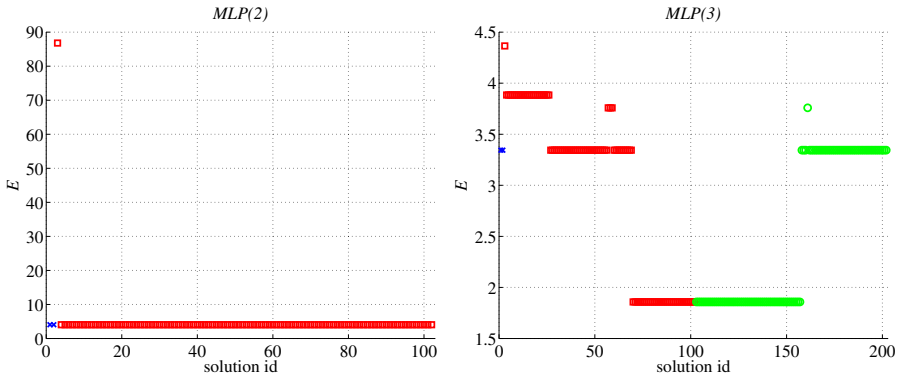
As the existing learning methods we employed BP and BPQ for comparison. Here the learning rate of BP is adaptively determined using the second-order Taylor expansion, since a constant learning rate does not work at all. BP or BPQ is performed 100 times for each MLP($J$) of each data set.

Any learning stops when a step length is less than $10^{-30}$ or the iteration exceeds 20,000 sweeps. As for the initialization of MLP weights, $w_{jk}$ and $w_j$ are randomly selected from the range $[0,\ 1]$, without $w_0 = \overline{y}$.

### 4.1 Experiment of Sigmoidal MLP Using Artificial Data

Our artificial data set for sigmoidal MLP was generated using the following MLP. Values of each explanatory variable $x_1, x_2, \cdots$, or $x_7$ were randomly selected from the range $[0,\ 1]$, while values of output $y$ were generated by adding a small Gaussian noise $\mathcal{N}(0, 0.05^2)$ to MLP outputs. Note that four explanatory variables $x_4, \cdots, x_7$ are irrelevant. The sample size was set to be 500. The number of hidden units was changed within 3: $J_{max}$ = 3.

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 2 \\ -5 \end{pmatrix}, \qquad (\boldsymbol{w}_1\ \boldsymbol{w}_2\ \boldsymbol{w}_3) = \begin{pmatrix} 5 & 6 & 7 \\ -15 & -17 & -13 \\ -17 & -10 & -16 \\ 16 & 15 & 12 \\ -14 & -15 & -19 \end{pmatrix}$$




(a) from MLP(1) to MLP(2) (b) from MLP(2) to MLP(3)

**Fig. 1.** Learning process of SSF1.1 for artificial sigmoidal data

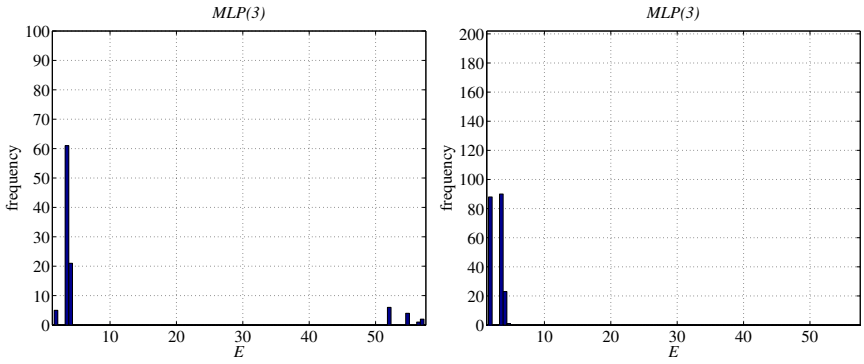
Figure 1 shows the result of SSF1.1. We repeated MLP($J$=1) learning 100 times and obtained two solutions. The better solution was used for the next step. The result for MLP($J$=2) is shown in Fig. 1 (a). We have two search points for reducibility mapping $\alpha\beta$ search, 50 points for reducibility mapping $\gamma$ interpolation, and 50 points for $\gamma$ extrapolation. The leftmost two points indicate the result of reducibility mapping $\alpha\beta$ search. Almost all search points converged to the same good solution, which was used for the next step. The result for MLP($J$=3) is shown in Fig. 1 (b). We have two points for $\alpha\beta$ search, 100 points for $\gamma$ search for splitting $w_1^{(2)}$, and other 100 points for $\gamma$ search for splitting $w_2^{(2)}$. Eighty eight search points converged to the same excellent solution.

**Table 1.** Best training error comparison for artificial sigmoidal data

| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 89.3263 | 86.0121 | 86.0121 |
| 2 | 4.0882 | 4.0322 | 4.0322 |
| 3 | 4.0950 | 1.8576 | 1.8576 |

As existing methods we ran adaptive BP and BPQ 100 times each. Table 1 compares the best training error $E$ for each $J$. BPQ and SSF1.1 achieved the same best training error for each $J$, and both showed monotonic decrease of $E$ as $J$ increased. On the other hand, adaptive BP could not achieve the best for each $J$ and did not show monotonic decrease of $E$ as $J$ increased. Figure 2 compares histograms of BPQ and SSF1.1 solutions. SSF1.1 reached the best solution 88 times out of 202, while BPQ reached the best solution 5 times out of 100. We see SSF1.1 found the excellent solution 8.7 times more stably.

Table 2 compares CPU time for artificial sigmoidal data. CPU time spent by SSF1.1 was much the same as that of BPQ. Adaptive BP was slow and all runs stopped by reaching the iteration upper bound.




(a) histogram of BPQ solutions (b) histogram of SSF1.1 solutions

**Fig. 2.** Histograms of solutions for artificial sigmoidal data

**Table 2.** CPU time comparison for artificial sigmoidal data (sec)

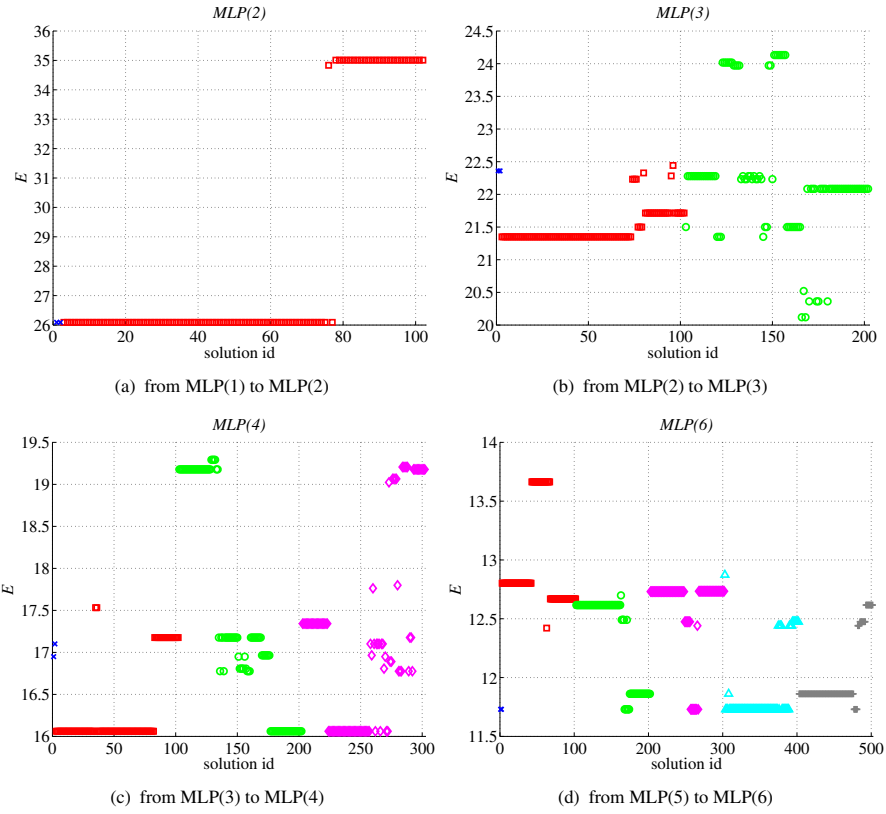
| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 412.96 | 3.31 | 3.36 |
| 2 | 498.12 | 6.39 | 5.42 |
| 3 | 610.45 | 15.69 | 19.70 |
| total | 1521.54 | 25.40 | 28.49 |

### 4.2 Experiment of Sigmoidal MLP Using Real Data

As a real data set for sigmoidal MLP we used Housing data set from UCI Machine Learning Repository. The number of explanatory variables is 12, and the sample size is 506. The number of hidden units was changed within 6: $J_{max} = 6$.

Figure 3 shows the result of SSF1.1. We repeated MLP($J$=1) learning 100 times and obtained a single solution, which was used for the next step. The result for MLP($J$=2) is shown in Fig. 3 (a). We have two search points for reducibility mapping $\alpha\beta$ search, and 100 points for reducibility mapping $\gamma$ interpolation and extrapolation. Here we have two kinds of solutions and the better one was used for the next step. The results for MLP($J$=3), MLP($J$=4), and MLP($J$=6) are shown in Fig. 3 (b), (c), and (d). The numbers of search points were 202, 302, and 502 respectively. From the figure we see the best solution was frequently obtained from different splitting. Finally, 104 points converged to the same excellent solution.

For comparison we ran adaptive BP and BPQ 100 times each. Table 3 compares the best training error $E$ for each $J$. BPQ and SSF1.1 achieved the same best training error for $J$=1 and 2, but SSF1.1 found better solutions than BPQ for larger $J$. Adaptive BP found rather poor solutions for $J > 2$ and did not show monotonic decrease as $J$ increased. Figure 4 compares histograms of BPQ and SSF1.1 solutions. SSF1.1 reached the best solution 104 times out of 502, while BPQ could not find the best solution for any 100 runs. Moreover, many solutions of SSF1.1 are located close to the best solution, while BPQ solutions are widely distributed. We see SSF1.1 found the excellent solution very stably.




(a) from MLP(1) to MLP(2)

(b) from MLP(2) to MLP(3)

(c) from MLP(3) to MLP(4)

(d) from MLP(5) to MLP(6)

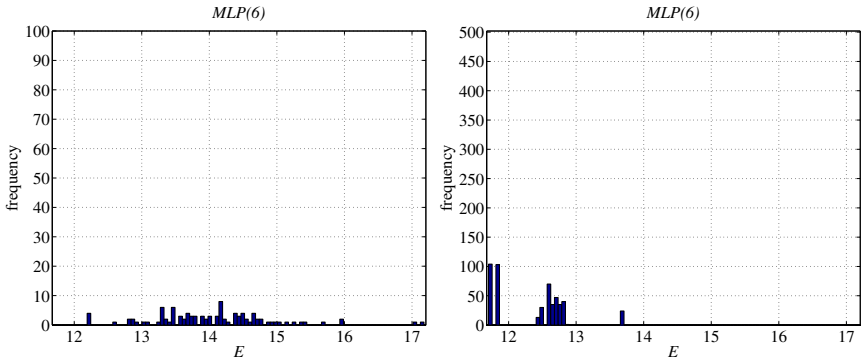
**Fig. 3.** Learning process of SSF1.1 for Housing data

**Table 3.** Best training error comparison for Housing data

| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 47.7244 | 47.5565 | 47.5565 |
| 2 | 26.7537 | 26.0902 | 26.0902 |
| 3 | 23.7415 | 20.3702 | 20.1187 |
| 4 | 21.1871 | 16.0666 | 16.0608 |
| 5 | 19.4489 | 14.2997 | 13.6640 |
| 6 | 19.4513 | 12.2430 | 11.7303 |

**Table 4.** CPU time comparison for Housing data (sec)

| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 418.83 | 4.47 | 4.41 |
| 2 | 508.05 | 12.24 | 12.69 |
| 3 | 624.40 | 15.32 | 32.50 |
| 4 | 829.45 | 25.35 | 61.62 |
| 5 | 985.75 | 37.57 | 158.54 |
| 6 | 1071.63 | 49.51 | 173.66 |
| total | 4438.12 | 144.46 | 443.43 |




(a) histogram of BPQ solutions (b) histogram of SSF1.1 solutions

**Fig. 4.** Histograms of solutions for Housing data

Table 4 compares CPU time for Housing data. SSF1.1 was three times slower than BPQ mainly because the search points got larger as $J$ increased. Adaptive BP was very slow and all runs stopped by reaching the iteration upper bound.

### 4.3 Experiment of Polynomial-Type MLP Using Artificial Data

Here we consider the following multivariate polynomial.

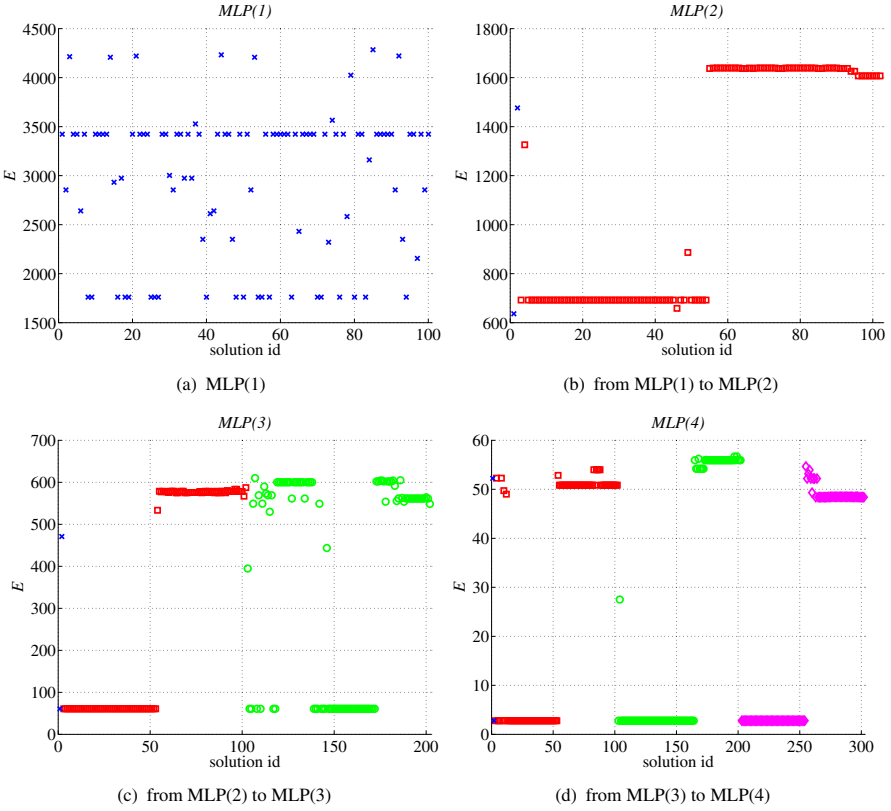
$$y = 2 + 60\, x_1^3\, x_2^6 x_3 + 40\, x_4^8\, x_5 + 20\, x_6\, x_7^7 + 10\, x_2\, x_5^8 \tag{12}$$

Values of each explanatory variable $x_1, x_2, \cdots$, or $x_{14}$ were randomly selected from the range $(0, 1)$, while values of output $y$ were generated following eq. (12). Seven explanatory variables $x_8, \cdots, x_{14}$ are irrelevant. The sample size was set to be 200. Considering eq. (12), we set as $J_{max} = 4$.

Figure 5 shows the result of SSF1.1. We repeated MLP($J$=1) learning 100 times and obtained several solutions, as shown in Fig. 3(a). The best solution was used for the next step. The result for MLP($J$=2) is shown in Fig. 5 (b). We have two search points for reducibility mapping $\alpha\beta$ search, and 100 points for $\gamma$ interpolation and extrapolation. The best solution obtained by $\alpha\beta$ search was used for the next step. The results for MLP($J$=3) and MLP($J$=4) are shown in Fig. 5 (c) and (d); the numbers of search points were 202 and 302 respectively. We see the best solution was frequently obtained from different splitting. Finally, 161 search points converged to the same excellent solution.

**Table 5.** Best training error comparison for artificial polynomial data

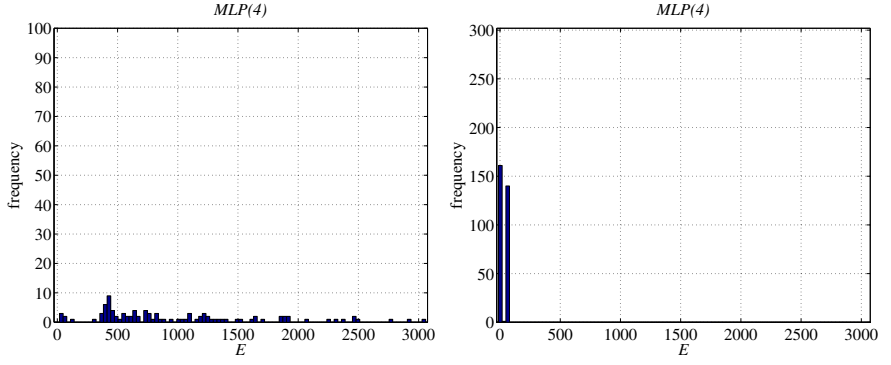
| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 1798.4900 | 1760.8019 | 1760.8019 |
| 2 | 720.2273 | 636.1869 | 636.1869 |
| 3 | 122.6208 | 60.7541 | 60.7541 |
| 4 | 65.4654 | 27.5101 | 2.7737 |




(a) MLP(1) (b) from MLP(1) to MLP(2)

(c) from MLP(2) to MLP(3) (d) from MLP(3) to MLP(4)

**Fig. 5.** Learning process of SSF1.1 for artificial polynomial data




(a) histogram of BPQ solutions (b) histogram of SSF1.1 solutions

**Fig. 6.** Histograms of solutions for artificial polynomial data

For comparison we ran adaptive BP and BPQ 100 times each. Table 5 compares the best training error $E$ for each $J$. Both BPQ and SSF1.1 achieved the same best training error for $J$=1, 2, and 3, but SSF1.1 found much better solution than BPQ for

**Table 6.** CPU time comparison for artificial polynomial data (sec)

| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 284.42 | 18.48 | 17.89 |
| 2 | 272.95 | 38.65 | 10.58 |
| 3 | 298.44 | 89.43 | 73.36 |
| 4 | 309.17 | 127.38 | 41.85 |
| total | 1164.98 | 273.93 | 143.68 |

$J$=4. Adaptive BP found very poor solutions for each $J$. Figure 6 compares histograms of BPQ and SSF1.1 solutions. SSF1.1 reached the best solution 161 times out of 302, while BPQ could not find the best solution for any 100 runs. Moreover, many solutions of SSF1.1 are located quite close to the best solution, while BPQ solutions are very widely distributed. We see SSF1.1 could find the excellent solution much more stably.

Table 6 compares CPU time for artificial polynomial data. SSF1.1 was about twice slower than BPQ mainly because the search points got larger as $J$ increased. Adaptive BP was again slow and all runs were terminated by reaching the iteration upper bound.

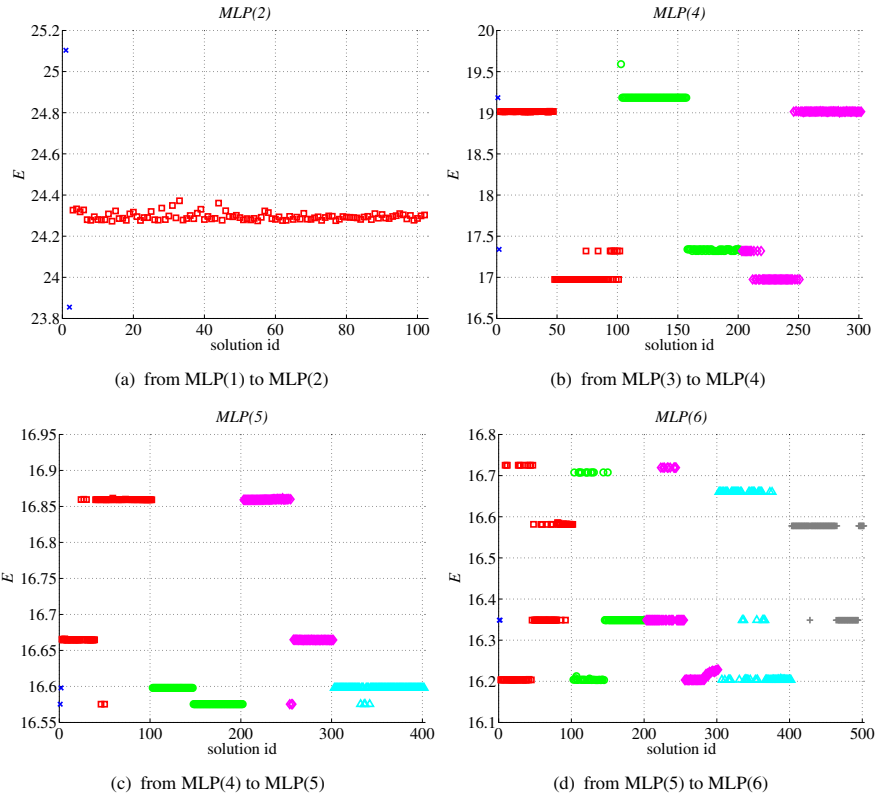
### 4.4 Experiment of Polynomial-Type MLP Using Real Data

As a real data set for polynomial-type MLP we used ball bearings data (Journal of Statistics Education). The objective is to estimate fatigue (L10) using load (P), the number of balls (Z), and diameter (D). Before learning, variables were normalized as $x_k/max(x_k)$ and $(y - \bar{y})/std(y)$. The sample size is 210, and we set as $J_{max} = 6$.

Figure 7 shows the result of SSF1.1. We repeated MLP($J$=1) learning 100 times and obtained two solutions; The better solution was used for the next step. The result for MLP($J$=2) is shown in Fig. 7 (a). We have two search points for reducibility mapping $\alpha\beta$ search, and 100 points for $\gamma$ interpolation and extrapolation. The best solution obtained by $\alpha\beta$ search was used for the next step. The results for MLP($J$=4), MLP($J$=5) and MLP($J$=6) are shown in Fig. 7 (b), (c) and (d); the numbers of search points were 302, 402 and 502 respectively. We see the best solution was frequently obtained from different splitting. Finally, 158 search points converged to the same excellent solution. We ran adaptive BP and BPQ 100 times each. Table 7 compares the best training error $E$. BPQ and SSF1.1 achieved the same best $E$ for $J$=1, 3, 4, and 5, but SSF1.1 outperformed BPQ for $J$=2 and 6. Adaptive BP found poor solutions for each $J$ and showed rugged change of $E$ as $J$ increased. Figure 8 compares histograms of BPQ and SSF1.1 solutions. SSF1.1 reached the best solution 158 times out of 502, while BPQ

**Table 7.** Best training error comparison for ball bearings data

| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 32.2456 | 29.6157 | 29.6157 |
| 2 | 27.9588 | 23.8523 | 23.8549 |
| 3 | 27.6056 | 19.5758 | 19.5758 |
| 4 | 28.4923 | 16.9736 | 16.9736 |
| 5 | 31.0674 | 16.5754 | 16.5754 |
| 6 | 29.2479 | 16.2055 | 16.2030 |



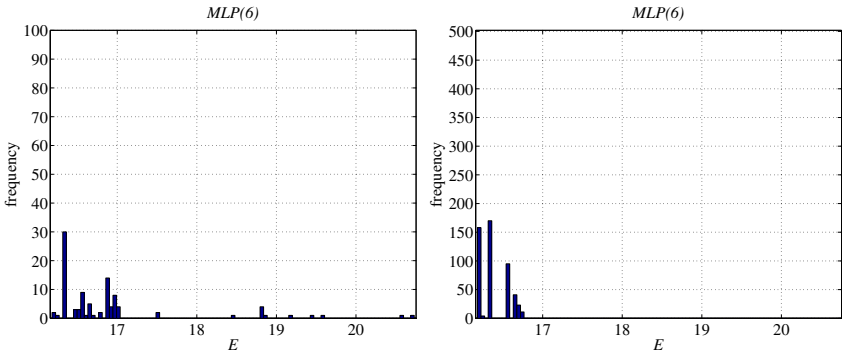

(a) from MLP(1) to MLP(2)

(b) from MLP(3) to MLP(4)

(c) from MLP(4) to MLP(5)

(d) from MLP(5) to MLP(6)

**Fig. 7.** Learning process of SSF1.1 for ball bearings data




(a) histogram of BPQ solutions

(b) histogram of SSF1.1 solutions

**Fig. 8.** Histograms of solutions for ball bearings data

did only twice out of 100. Moreover, most solutions of SSF1.1 are located close to the best, while BPQ solutions scatter widely. We see SSF1.1 found the excellent solution 15.7 times more stably.

Table 8 compares CPU time for ball bearings data. SSF1.1 was about 7 times slower than BPQ mainly because the search points got larger as $J$ increased. Adaptive BP was slow and all runs stopped reaching the iteration upper bound.

**Table 8.** CPU time comparison for ball bearings data (sec)

| $J$ | BP | BPQ | SSF1.1 |
|---|---|---|---|
| 1 | 274.35 | 5.60 | 5.71 |
| 2 | 262.64 | 12.14 | 14.11 |
| 3 | 279.27 | 33.89 | 29.83 |
| 4 | 290.10 | 77.84 | 351.41 |
| 5 | 309.05 | 111.51 | 1405.61 |
| 6 | 325.50 | 128.66 | 806.67 |
| total | 1740.90 | 369.63 | 2613.33 |

## 5 Conclusions

This paper proposed a new MLP learning method called SSF1.1, which makes good use of the whole singular regions. It begins with MLP($J$=1) and gradually increases the number of hidden units one by one. Our various experiments using sigmoidal and polynomial-type MLP showed that, compared with existing methods such as BP or quasi-Newton method, SSF1.1 quite stably and successively found solutions excellent enough for MLP($J$). In the future we plan to improve our method by reducing time complexity and apply it to a wide variety of data.

**Acknowledgements.** This work was supported by Grants-in-Aid for Scientific Research (C) 22500212 and Chubu University Grant 22IS27A.

## References

1. Amari, S.: Natural gradient works efficiently in learning. Neural Computation 10(2), 251–276 (1998)
2. Amari, S., Park, H., Fukumizu, K.: Adaptive method of realizing natural gradient learning for multilayer perceptrons. Neural Computation 12(6), 1399–1409 (2000)
3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. John Wiley & Sons, Inc. (2001)
4. Fukumizu, K., Amari, S.: Local minima and plateaus in hierarchical structure of multilayer perceptrons. Neural Networks 1(3), 317–327 (2000)
5. Hamey, L.G.C.: XOR has no local minima: a case study in neural network error surface. Neural Networks 11(4), 669–681 (1998)
6. Minnett, R.C.J., Smith, A.T., Lennon Jr., W.C., Hecht-Nielsen, R.: Neural network tomography: network replication from output surface geometry. Neural Networks 24(5), 484–492 (2011)
7. Luenberger, D.G.: Linear and nonlinear programming. Addison-Wesley (1984)
8. Nakano, R., Saito, K.: Discovering Polynomials to Fit Multivariate Data Having Numeric and Nominal Variables. In: Arikawa, S., Shinohara, A. (eds.) Progress in Discovery Science. LNCS (LNAI), vol. 2281, pp. 482–493. Springer, Heidelberg (2002)

9. Nakano, R., Satoh, S., Ohwaki, T.: Learning method utilizing singular region of multilayer perceptron. In: Proc. 3rd Int. Conf. on Neural Computation Theory and Applications, pp. 106–111 (2011)
10. Watanabe, S.: Algebraic geometry and statistical learning theory. Cambridge Univ. Press (2009)
11. Saito, K., Nakano, R.: Partial BFGS update and efficient step-length calculation for three-layer neural networks. Neural Computation 9(1), 239–257 (1997)
12. Sussmann, H.J.: Uniqueness of the weights for minimal feedforward nets with a given input-output map. Neural Networks 5(4), 589–593 (1992)

# Interacting Individually and Collectively Treated Neurons for Improved Visualization

Ryotaro Kamimura

IT Education Center, 1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292 Japan
ryo@keyaki.cc.u-tokai.ac.jp

**Abstract.** In this paper, we propose a new type of learning method in which neurons are treated individually and collectively. In addition, the collectivity is defined in terms of distance and similarity between neurons. We applied the method to the self-organizing maps, because our method makes it possible to control flexibly a process of cooperation between neurons. Then, we applied the method with the self-organizing maps to the visualization of the pound-yen exchange rates. We succeeded in producing clearer class structure. The entire period of the exchange rates was divided into three distinct periods.

## 1 Introduction

In this chapter, we propose a new type of neural learning method, based upon the separation and interaction of individually and collectively treated neurons. Neurons have been treated individually or collectively, depending upon learning methods. However, little attention has been paid to the importance of the separation of individually and collectively treated neurons. We try to show the importance of distinction between two types of neurons by using self-organizing maps. In the self-organizing maps, much attention has been paid to cooperation or collectivity of neurons. The introduction of individuality of neurons has much influence on learning and final visualization performance. In addition, the distinction also makes it possible to construct collectively treated neurons, taking into account the characteristics of individually treated neurons. We here show the influence of distinction in terms of visualization and a new concept of collectivity.

First, distinction between individually and collectively treated neurons makes it possible to visualize SOM knowledge more explicitly. The self-organizing maps [1], [2] have been used for many applications, because of the visualization performance. Paradoxically, we have had much difficulty in visualizing SOM's knowledge. Due to this difficulty, there have been many attempts to develop methods to visualizse SOM's knowledge, for example, non-linear projection methods [3], U-matrix and its variants [4], [5], coloring, component planes [6], [7], [8], visualization-oriented learning algorithms [9], [10], [11]. One of the main reasons for this difficulty in visualization lies in focus upon cooperation among neurons in SOM. In the SOM, neurons are forced to cooperate with each other. Neighboring neurons are forced to behave in the same way as much as possible. To separate classes in the data or to make class boundaries clearer, it is necessary to find discontinuity between neurons. At this point, the introduction of individuality of neurons plays an important role in visualizing class boundaries.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 277–290.
DOI: 10.1007/978-3-642-35638-4_18 

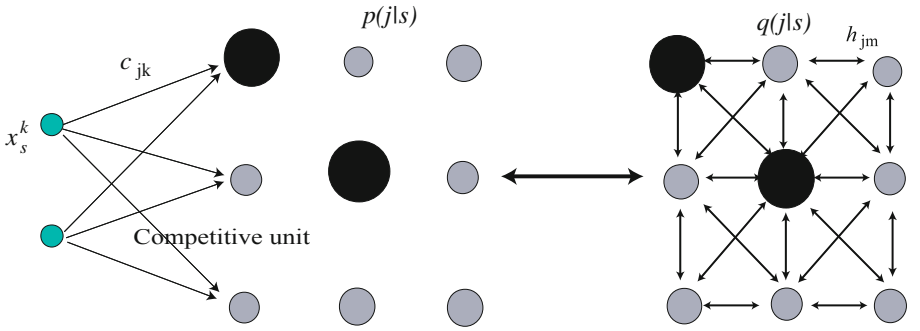



**Fig. 1.** Individually (a) and collectively (b) treated neurons. Some connection weights were omitted for simple representation.

The individuality can be used to make it possible to disconnect neurons and make class boundaries clearer. The individuality can be used to weaken cooperation between neurons and to produce boundaries between neurons.

Second, the distinction also makes it possible to define more exactly the collectivity of neurons. We can create collectively treated neurons not by collecting all neighboring neurons' activities but by weighting the neighboring neurons. By this weighting, collectively treated neurons can take into account the characteristics of individually treated neurons. For this purpose, we introduce similarity between neurons in addition to closeness between neurons. In the conventional SOM, much attention has been paid to closeness between neurons. If two neurons are nearby located, they should cooperate with each other more strongly. However, we can say that even if two neurons are nearby located, but if they are not similar to each other, for example, in terms of neuron's outputs, they should less strongly cooperate with each other. When we take into account similarity between neurons, two neurons cooperate more strongly with each other, only if two neurons are nearby located and similar to each other. The separation of individuality and collectivity and the introduction of similarity can be used to visualize SOM' knowledge more clearly.

We first explain how to compute individually treated neurons. Then, we compute collectively treated neurons by taking into account closeness and similarity between neurons. We define the re-estimation formula to obtain connection weights by minimizing *KL* divergence and free energy. We applied the method to the visualization of pound-yen exchange rates for the entire period of 2011. The new method showed very clear class structure. The method could divide the entire period into three distinct ones. The exceptional cases with high exchange rates turned out to be treated separately.

## 2 Theory and Computational Methods

**Individually Treated Neurons.** We have shown that there are two types of neurons, namely, individually and collectively treated neurons. Figure 1 (a) shows an example of individually treated neurons in which all neurons are disconnected. On the other hand,

Figure 1(b) shows a collectively treated neuron in which all neurons cooperate with other. One of the most fundamental ways to cooperate with each other is that when two neurons fire in the same way, two neurons fire more strongly.

Let us explain how to compute outputs from competitive units and input patterns in Figure 1 (a). The $s$th input pattern of total $S$ patterns can be represented by

$$\mathbf{x}^s = [x_1^s, x_2^s, \cdots, x_L^s]^T, \quad s = 1, 2, \cdots, S. \tag{1}$$

Connection weights into the $j$th competitive unit of total $M$ units are computed by

$$\mathbf{c}_j = [c_{j1}, c_{j2}, \cdots, c_{jL}]^T, \quad j = 1, 2, \ldots, M. \tag{2}$$

The $j$th competitive unit output can be computed by

$$v_j^s = \exp\left\{-\frac{1}{2}(\mathbf{x}^s - \mathbf{c}_j)^T \mathbf{\Lambda} (\mathbf{x}^s - \mathbf{c}_j)\right\}, \tag{3}$$

where $\mathbf{x}^s$ and $\mathbf{w}_j$ are supposed to represent $L$-dimensional input and weight column vectors, where $L$ denotes the number of input units. The $L \times L$ matrix $\mathbf{\Lambda}$ is called a "scaling matrix," and the $kl$th element of the matrix denoted by $(\mathbf{\Lambda})_{kl}$ is defined by

$$(\mathbf{\Lambda})_{kl} = \frac{\delta_{kl}}{\sigma_\beta^2}, \quad k, l = 1, 2, \cdots, L. \tag{4}$$

where $\sigma_\beta$ is a spread parameter. The output is increased when connection weights become closer to input patterns.

**Collectively Treated Neurons.** Using those individually treated neurons, we can compute collectively treated neurons by taking into account distance and similarity between neurons. First, distance between neurons is computed by distance between neurons on the map. Relations between the $j$th neuron and $m$th neuron $h_{jm}$ are defined by

$$h_{jm} = \exp\left(\frac{\|\mathbf{r}_j - \mathbf{r}_m\|^2}{2\sigma_\gamma^2}\right), \tag{5}$$

where $\mathbf{r}_j$ and $\mathbf{r}_m$ denote the position of the $j$th and the $m$th unit on the output space and $\sigma_\gamma$ is a spread parameter. By using this neighborhood (distance) function, we can define a collectively treated neuron

$$y_j^s = \sum_{m=1}^{M} h_{jm} v_m^s. \tag{6}$$

This equation shows that when neurons are closer and similar to each other, they fire more strongly as shown in Figure 2(a). This definition of collectively treated neurons is close to that by the conventional SOM.

In addition to this distance between neurons, we can take into account similarity between neurons. Figure 2 (a) shows a concept of similarity between neurons. In the

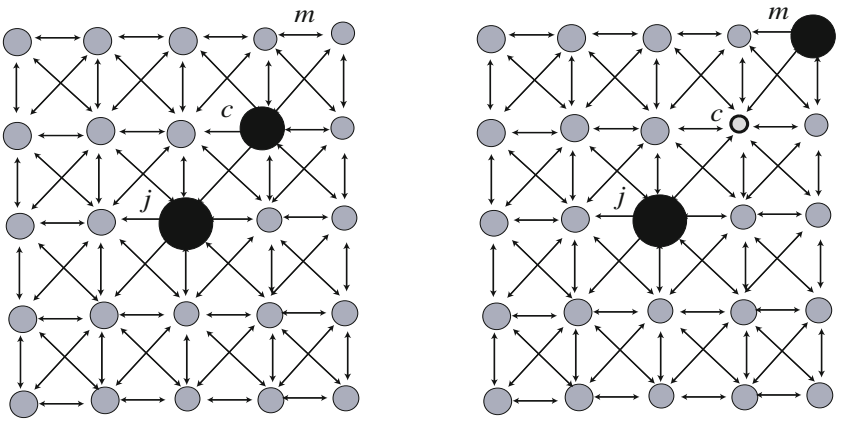



**Fig. 2.** Concept of distance (a) and similarity (b). Some connection weights were omitted for simple representation.

figure, the $c$th neuron is closer to the $j$th neuron and they are strongly connected by the standard self-organizing maps. In our model, the $m$th neuron fire strongly as the $j$th neuron do in Figure 2 (b). Thus, two neurons are similar to each other in terms of firing rates or outputs. They are strongly connected in spite of distance between two neurons. We can take into account similarity between neurons by

$$y_j^s = \sum_{m=1}^{M} v_j^s h_{jm} v_m^s. \tag{7}$$

This equation shows that when $j$th neuron and the $m$th neuron are close and similar to each other, collectively treated neurons fire more strongly. Thus, the collectively treated neurons can take into account distance as well as similarity between neurons. The firing probability of the collectively treated neurons can be obtained by

$$q(j \mid s) = \frac{y_j^s}{\sum_{m=1}^{M} y_m^s}. \tag{8}$$

**Re-estimation Formula.** We must decrease distance between individually and collectively treated neurons as much as possible. For this purpose, we introduce the Kullback-Leibler divergence between individually and collectively treated neurons

$$KL = \sum_{s=1}^{S} p(s) \sum_{j=1}^{M} p(j \mid s) \log \frac{p(j \mid s)}{q(j \mid s)}. \tag{9}$$

This $KL$ divergence should be as small as possible. We can minimize this divergence with a constraint

$$E = \sum_{s=1}^{S} p(s) \sum_{j=1}^{M} p^*(j \mid s) \|\mathbf{x}^s - \mathbf{w}_j\|^2. \tag{10}$$

By minimizing the KL divergence, we have

$$p(j \mid s) = \frac{q(j \mid s)v_j^s}{\sum_{m=1}^{M} q(m \mid s)v_m^s}. \tag{11}$$

This equation shows that individually treated neurons' outputs should be weighted by the outputs from the corresponding collectively treated neurons.

For obtaining connection weights, we introduce the free energy. The free energy is obtained by putting $q(j \mid s)$ into the $KL$ divergence. It can be defined by

$$F = -2\sigma^2 \sum_{s=1}^{S} p(s) \log \sum_{j=1}^{M} \exp\left\{-\frac{1}{2}(\mathbf{x}^s - \mathbf{c}_j)^T \mathbf{\Lambda}(\mathbf{x}^s - \mathbf{c}_j)\right\}. \tag{12}$$

This equation can be expanded as

$$\begin{aligned} F = &\sum_{s=1}^{S} p(s) \sum_{j=1}^{M} p(j \mid s)\|\mathbf{x}^s - \mathbf{w}_j\|^2 \\ &+2\sigma_\beta^2 \sum_{s=1}^{S} p(s) \sum_{j=1}^{M} p^*(j \mid s) \log \frac{p(j \mid s)}{q(j \mid s)}. \end{aligned} \tag{13}$$

Thus, the free energy can be used to decrease $KL$ divergence as well as quantization errors. By differentiating the free energy, we have the re-estimation formula

$$\mathbf{w}_j = \frac{\sum_{s=1}^{S} p(j \mid s)\mathbf{x}^s}{\sum_{s=1}^{S} p(j \mid s)}. \tag{14}$$

This re-estimation formula is repeated until a criterion for convergence is met.

## 3 Results and Discussion

### 3.1 Experimental Results

**Experimental Setting and Data Description.** We here present experimental results on the pound-yen exchange rates. Our objective is not to estimate the future exchange rates but to visualize the past one-year exchange rate fluctuation. We can easily check how well our method can visualize the exchange rates, because the exchange rates are linear and easily interpreted. We aim to show how well our method captures the characteristics of exchange rates whose linear characteristics make it possible to estimate the utility of our method. The well-known SOM toolbox of Vesanto et al. [12] was used, because the final results of the SOM have been very stable. For easy reproduction of our results, we used the well-known and simple error measures for quantification evaluation, namely, quantization and topographic errors. The quantization error is the average distance from each data vector to its BMU (best-matching unit). The topographic error is the percentage of data vectors for which the BMU and the second-BMU are not neighboring units [13].

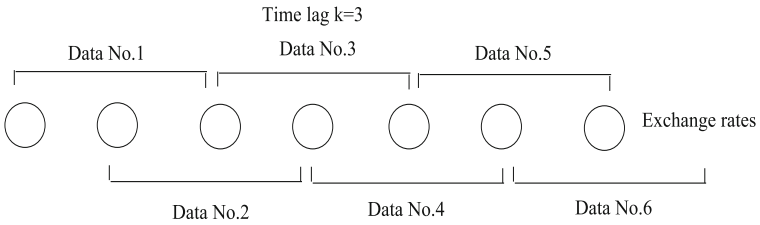



**Fig. 3.** Data description when the time lag is three

**Parameter Setting.** In the experiment, we used the pound-yen exchange rates. The data was composed of the previous $k$ values of the exchange rates as shown in Figure 3. Thus, we must determine the time lag value of $k$. To determine the time lag, we used mutual information between competitive units and input patterns of collectively treated neurons

$$I = \sum_{s=1}^{S} p(s) \sum_{j=1}^{M} q(j \mid s) \log \frac{q(j \mid s)}{q(j)}. \tag{15}$$

Mutual information is an important criterion to determine the time lag, because it shows how organized collectively treated neurons are. We must try to increase mutual information on input patterns as much as possible by changing the time lag. Figure 4 shows mutual information as a function of the time lag. As can be seen in the figure, mutual information gradually increased and reached its peak when the time lag was three. Then, mutual information remained almost unchanged. Figure 5 shows the U-matrices when the time lag was changed from one (a) to four (d). When the time lag was one in Figure 5(a), a class boundary was located vertically, which was not shared by the other values of the time lag. When the time lag was increased to two in Figure 5(b), a wide class boundary in the middle of the map appeared. When the time lag was increased to three in Figure 5(c), two major class boundaries were produced with the other minor one on the lower side. When the time lag was four in Figure 5(d), the most stable pattern of the U-matrix could be obtained. When the time lag was four, mutual information became stable as shown in Figure 4. Thus, we chose the time lag as four for the experiment.

**Quantitative Evaluation.** Quantization errors decreased and mutual information increased when the parameter $\beta$ was increased. However, topographic errors did not constantly decrease or increase. Figure 6(a) shows quantization errors when the parameter $\beta$ was changed from one to 20. The quantization errors decreased gradually and reached the final value of 0.702, which was less than 0.917 by the conventional SOM. When the parameter $\beta$ was further increased beyond 20, qunatization errors tended to decrease. Figure 6(b) shows topographic errors as a function of the parameter $\beta$. The topographic errors gradually increased and reached the peak for $\beta = 9$, and then decreased gradually. The minimum value of 0.129 ($\beta = 18$) was lower than 0.212 by the conventional SOM. Figure 6(c) shows mutual information as a function of the parameter $\beta$. When the parameter $\beta$ was increased, information gradually increased. However, the maximum value of 0.216 was lower than 0.228 by the conventional SOM. Correlation coefficient

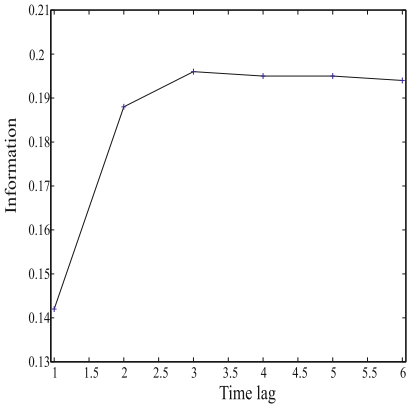



**Fig. 4.** Mutual information of collectively treated neurons as a function of the time lag

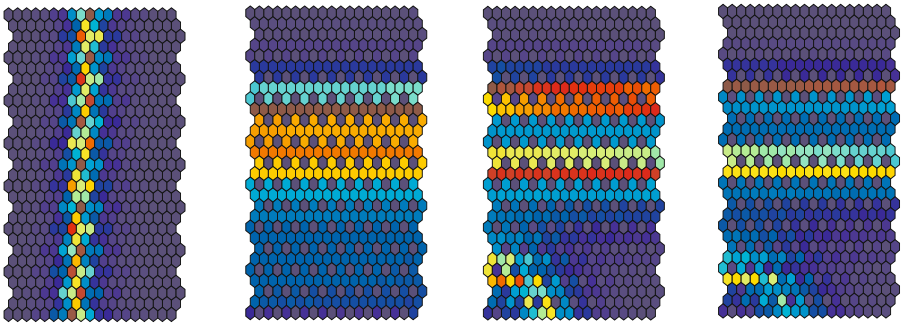
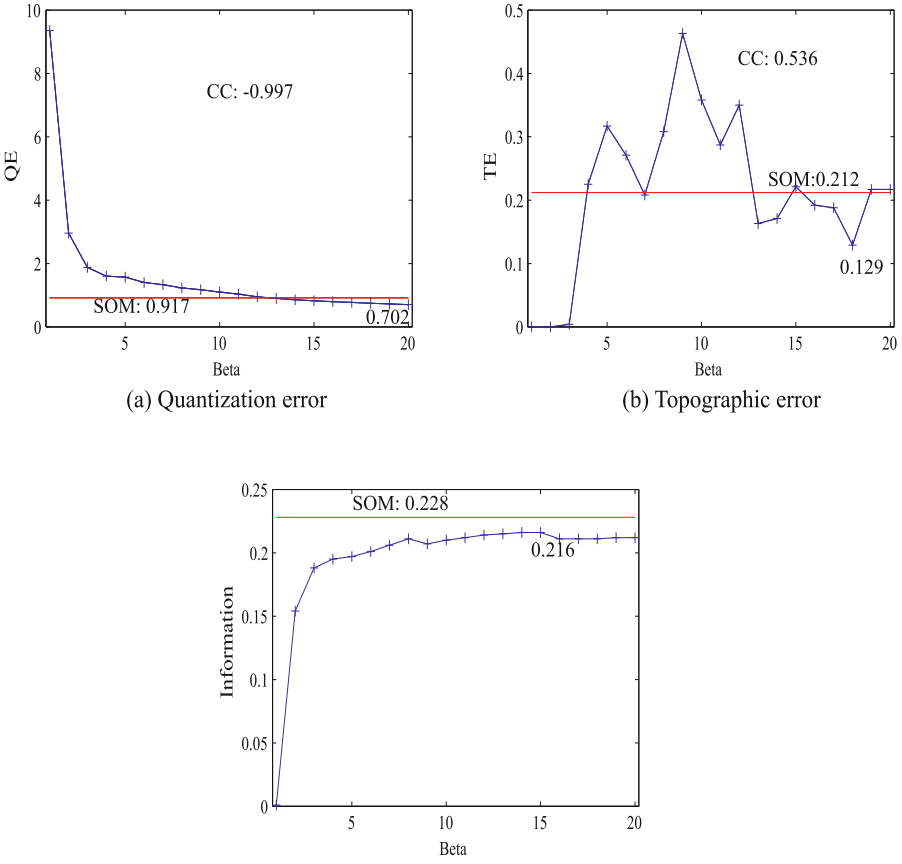



**Fig. 5.** U-matrices when the time lag was changed from one to four. The parameter $\beta$ was four.

between quantization errors and information was -0.997, while correlation coefficient between information and topographic errors was 0.536. This means that quantization errors decreased when mutual information increased.

Experimental results showed that quantization errors could be decreased and mutual information could also be increased only by increasing the parameter $\beta$. However, we had difficulty in decreasing the topographic errors. Thus, fidelity to input patterns in terms of topographic errors could not be easily controlled. We must carefully change the parameter $\beta$ to compromise between quantization and topographic errors.

**Visual Evaluation.** Our method showed very clear U-matrices in which clearer class boundaries could be seen. Figure 7 (a) shows the U-matrix by the conventional SOM. Though two class boundaries in warmer color seems to be present, those class boundaries were wide and ambiguous. It was difficult to see clear class boundaries on the matrix. Figure 7(b) shows the U-matrix when the parameter $\beta$ was one. A huge class boundary in the middle of the matrix was detected. When the parameter $\beta$ was increased to two, the huge class boundary became sharper in Figure 7 (c). When the parameter




(a) Quantization error (b) Topographic error

(c) Information

**Fig. 6.** Quantization (a), topographic (b) errors and information (c). The symbol CC represents the correlation coefficient between information and quantization and topographic errors.

$\beta$ was four in Figure 7 (d), three class boundaries appeared. When the parameter $\beta$ was increased to six in Figure 7(e), three class boundaries deteriorated. In particular, a class boundary in the middle of the matrix became weak. When the parameter $\beta$ was increased from 8 in Figure 7 (f) to 20 in Figure 7 (i), many minor class boundaries were produced.

These results showed that clearer class boundaries on the U-matrices could be obtained. However, the U-matrices were dependent upon the parameter $\beta$. When the parameter $\beta$ was increased and mutual information was increased, the U-matrices became more detailed. Thus, we must carefully choose the parameter $\beta$ and control mutual information to obtain the clearest U-matrix.

**Interpretation.** Though we had difficulty in interpreting the U-matrix of the conventional SOM, the interpretation of the U-matrices obtained by our method corresponded to our intuition on the pound-yen exchange rate fluctuation. Figure 8 shows the

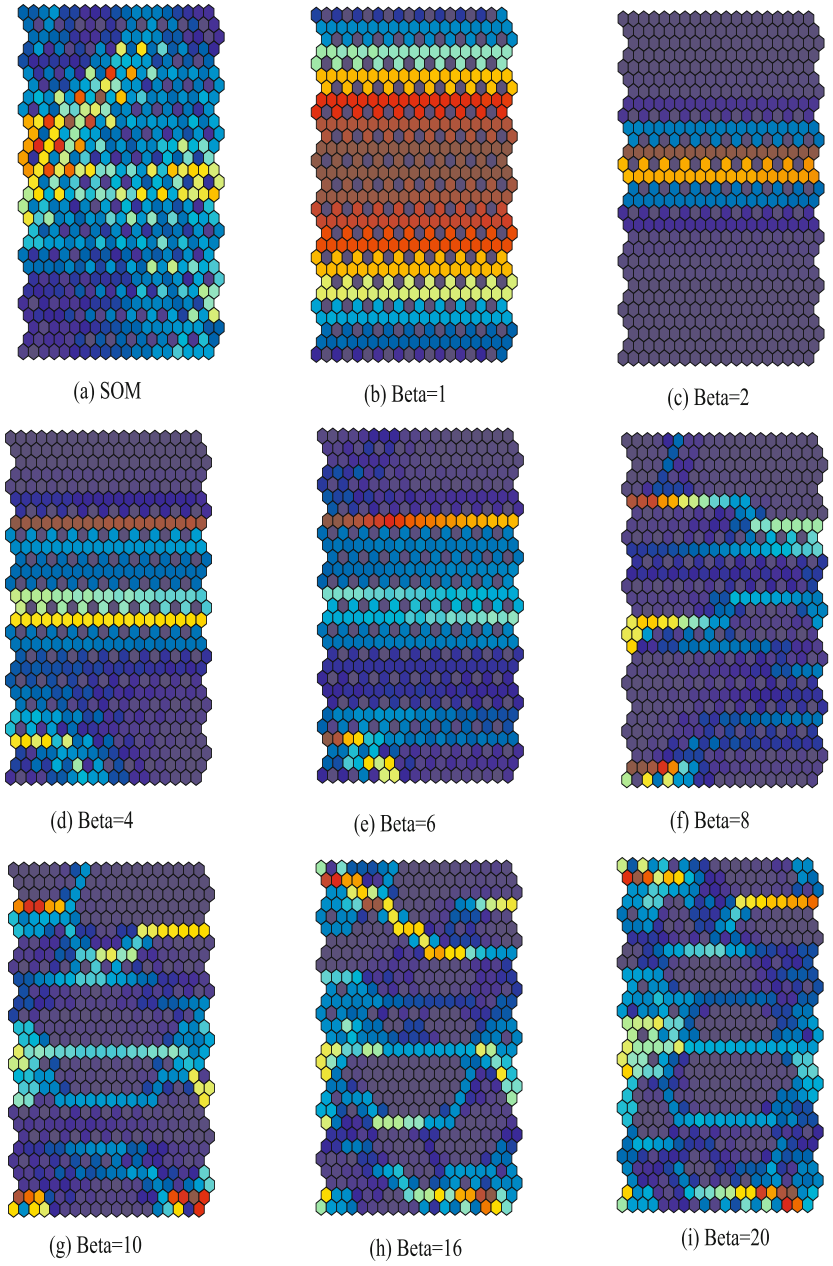
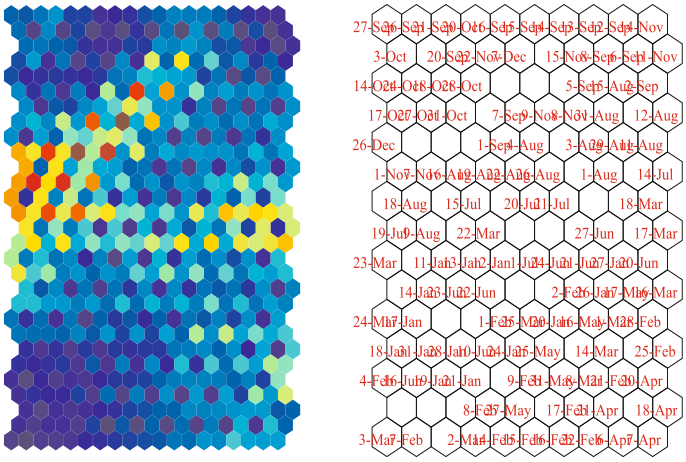



**Fig. 7.** U-matrices by the conventional SOM (a) and our method with 15 by 10 map whose parameter $\beta$ was increased from one (b) to 20 (i) for the pound-yen exchange rate

U-matrix (a) and labels (b) by the conventional SOM. As can be seen in the figure, huge and sparse class boundaries prevented us from identifying explicit class structure. Figure 9 (a) shows U-matrix (a1) and labels (a2) when the parameter $\beta$ was four. We could



(a) U-matrix (b) Labels

**Fig. 8.** U-matrices (a) and labels (b) by the conventional SOM for the pound-yen exchange rate

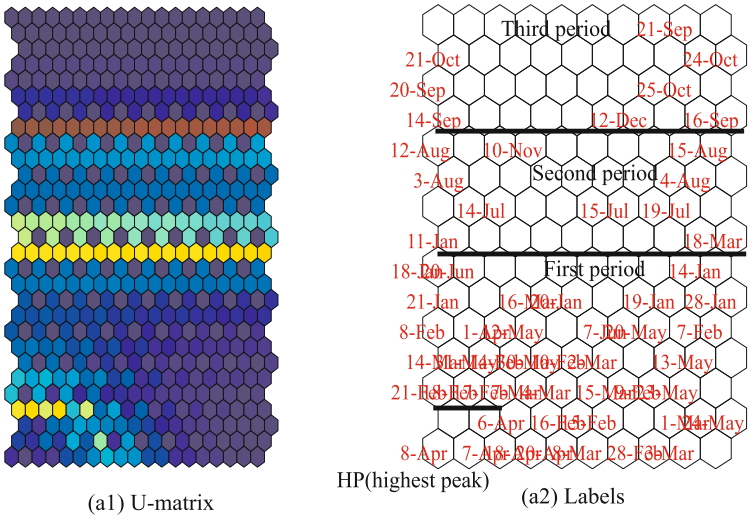
see clear three class boundaries. The lowest small class boundary represents the period with the highest peak in Figure 10. The lower part corresponds to the first period (Janurary-June). The middle part is the second part (July-August). The upper part corresponds to the third part (September to December). When the parameter $\beta$ was increased to six in Figure 9(b), the class boundary in the middle became unclear. Data points were distributed more evenly. This means that the boundary between the first and the second period is weak. In Figure 10, we can see gradual change between the first and the second period.

Experimental results showed that our method could divide the period into three ones and the exceptional cases were treated separately. This interpretation corresponds to our intuition on the rate fluctuation.

### 3.2 Discussion

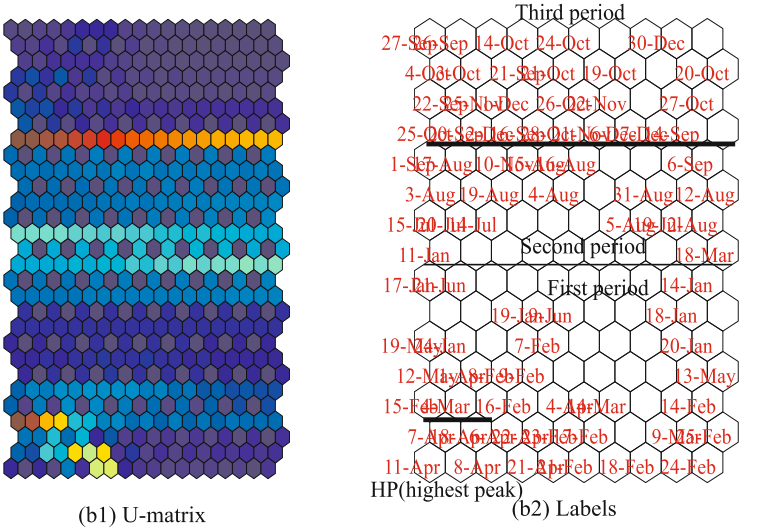
**Validity of Methods and Experimental Results.** We have proposed a new type of learning method in which individually and collectively treated neurons interact with each other. The separation of two types of neurons can be used to clarify class structure in the self-organizing maps. The SOM has been exclusively concerned with the collective behavior of neurons or cooperation between neurons. This focus on the cooperation has made it difficult for the SOM to be applied to the detection of class boundaries, because the effect of cooperation is to reduce discontinuity between neurons. By separating the individually treated neurons, we can control cooperation between neurons and produce discontinuity between neurons leading to class boundaries. In addition to distance between neurons, we have introduced similarity between neurons. Two neurons cooperate strongly with each other only if they are nearby located and at the same time they fire to input patterns in the same way.

We applied the method to the visualization of pound-yen exchange rates during 2011. Quantization errors decreased when the parameter $\beta$ was increased. Topographic errors




(a1) U-matrix (a2) Labels

(a) Beta=4




(b1) U-matrix (b2) Labels

(b) Beta=6

**Fig. 9.** U-matrices (a) and labels (b) for $\beta$=4 and 6 for the pound-yen exchange rate

did not decrease when the parameter $\beta$ was increased. However, smaller topographic errors were obtained for larger and smaller values of the parameter. Our method could divide the entire period into three periods. In addition, the highest peak was treated differently. This result confirmed our intuition of the pound-yen exchange rates.

Experimental results showed that explicit class structure could be obtained by the interaction of individually and collectively treated neurons. The interpretation of the structure revealed the main characteristics of pound-yen exchange rate fluctuation. Fidelity to input patterns in terms of quantization errors could be increased just by

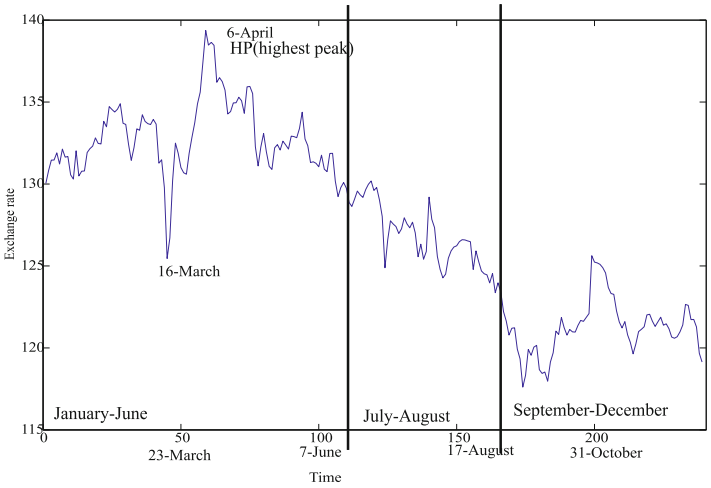



**Fig. 10.** Pound-yen exchange rates during 2012

increasing mutual information and by increasing the parameter $\beta$. However, the other fidelity to input patterns in terms of topographic errors could not be easily improved. The results showed a possibility of explicit class structure at the expense of fidelity to input patterns.

**Problems of the Method.** Though our method has shown better performance in clarifying class structure, two problems should be solved for our method to be practically applicable, namely, the choice of the parameter and criterion for the clarity of class structure. First, we had much difficulty in determining the appropriate information or parameter $\beta$. Quantization errors decreased when the parameter $\beta$ was increased. However, we observed large topographic errors for some values of the parameter $\beta$. When the parameter $\beta$ was small, topographic errors were small. In addition, when the parameter $\beta$ was large, topographic errors were also small. However, when the parameter $\beta$ was in an intermediate level, topographic errors were large. In this intermediate level, clearer U-matrices were obtained as shown in Figure 7. This suggests that clearer class structure could be obtained only at the expense of fidelity to input patterns.

Second, no criteria to describe the clarity of class structure have existed. Related to the first problem, we have had much difficulty in describing the clarity of class structure. We have had used mutual information as one of the possible criteria. However, as shown in 7, when the parameter $\beta$ was increased and mutual information was increased, the U-matrices became more complicated and detailed. Detailed and complex representations do not necessarily correspond to the clarity of class structure. Thus, we need to develop methods to describe the clarity of class structure.

**Possibility of the Method.** The possibility of the method can be described in terms of SOM's visualization, time-lag analysis, improved performance in estimation and spatial representation.

First, the method can be used to improve the SOM's visualization performance. The SOM has received good reputation for its ability for visualization. Paradoxically, we have had much difficulty in visualization SOM's knowledge. Because of this difficulty, a number of visualization techniques for SOM have been developed as discussed in the introduction section. Our method can be used to provide these techniques with more easily interpretable knowledge. The easily interpretable representations can be obtained by the interaction of individually and collectively treated neurons. When individually and collectively treated neurons are close to each other, they are enhanced to fire more strongly. In addition, if individually and collectively treated neurons respond to input patterns in the same way, they fire more strongly even if they are not so close to each other in terms of distance between neurons. The enhancement by distance and similarity between neurons makes it possible to produce explicit class boundaries.

Second, we can give a new insight to the problem of the time lag in the time-series analysis. The time lag is a useful way to describe the time series. However, it is impossible to show how many lags must be necessary. Our method showed the peak value around the lag=4. Because this paper is not concerned with the real estimation of pound-yen exchange rates, it is impossible at the moment to say that this value of lag is related to the estimation. However, the results shows a possibility of the determination of optimal time lag for good estimation.

Related to the optimal time lag, our method can be used to improve the estimation performance of neural network applied to the time-series. Our method can be used to explain how neural networks represent knowledge in input patterns. The representation obtained by our method for the pound-yen exchange rates corresponds to our intuition. Thus, we can expect that this method can be used to improve the future trend of exchange rates.

Finally, we have a possibility that linear and time-series representations can be transformed into spatial representations, which are more easily interpreted. Time series analysis has been applied to many actual data. However, the main focus is on improved performance in estimating targets and little attention has been paid to obtained internal representations. We have been concerned with the interpretation of obtained internal representations. In particular, we try to interpret how and why neural networks tries to capture the time-dependent data. As discussed in the experimental results, we succeeded in clarifying the meaning of representations. In other words, we succeeded in transforming linear representations into spatial ones for easy interpretation. We think that it is possible to extract characteristics independent of the time.

## 4 Conclusions

In this paper, we have proposed a new type of neural learning method. In the method, the distinction of two types of neurons, namely, individually and collectively treated neurons, has been made. In addition, the distinction of two types of neurons has made it possible to introduce similarity between neurons in addition to distance between neurons to compute collectively treated neurons. By the introduction of similarity, neurons cooperate more strongly with each other when neurons are close to each other and they are similar to each other in terms of responses to input patterns.

We have applied the method to the self-organizing maps, because much attention has been paid to the collectivity of neurons. The introduction of individually and collectively treated neurons has made it possible to control the process of cooperation. In the self-organizing maps, if two neurons are close to each other, the neurons fire more strongly. In our method, in addition to distance between neurons, similarity between neurons was taken into account. If two neurons are close and similar to each other, they fire more strongly. The introduction of individually treated neurons can be used to make class boundaries clearer.

We applied the method to the visualization of pound-yen exchange rates during 2011. We observed that quantization errors decreased gradually when the parameter $\beta$ was increased. Mutual information increase turned out to be correlated with quantization errors. Though topographic errors did not constantly decreased when the parameter $\beta$ was increased. We observed lower errors when the parameter $\beta$ were larger. We succeeded in making class boundaries clearer and the method could divide the period into three ones. In addition, we could observe that an exceptional period with high exchange rates was treated differently. The next step of our method is to examine how and to what extent the target estimation is improved. In other words, we should examine how well our method can be used to estimate the future exchange rates.

## References

1. Kohonen, T.: Self-Organizing Maps. Springer (1995)
2. Kohonen, T.: The self-organization map. Proceedings of the IEEE 78, 1464–1480 (1990)
3. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers C-18, 401–409 (1969)
4. Ultsch, A., Siemon, H.P.: Kohonen self-organization feature maps for exploratory data analysis. In: Proceedings of International Neural Network Conference, pp. 305–308. Kulwer Academic Publisher, Dordrecht (1990)
5. Ultsch, A.: U*-matrix: a tool to visualize clusters in high dimensional data. Technical Report 36, Department of Computer Science, University of Marburg (2003)
6. Vesanto, J.: SOM-based data visualization methods. Intelligent Data Analysis 3, 111–126 (1999)
7. Kaski, S., Nikkila, J., Kohonen, T.: Methods for interpreting a self-organized map in data analysis. In: Proceedings of European Symposium on Artificial Neural Networks, Bruges, Belgium (1998)
8. Mao, I., Jain, A.K.: Artificial neural networks for feature extraction and multivariate data projection. IEEE Transactions on Neural Networks 6, 296–317 (1995)
9. Yin, H.: ViSOM-a novel method for multivariate data projection and structure visualization. IEEE Transactions on Neural Networks 13, 237–243 (2002)
10. Su, M.C., Chang, H.T.: A new model of self-organizing neural networks and its application in data projection. IEEE Transactions on Neural Networks 123, 153–158 (2001)
11. Xu, L., Xu, Y., Chow, T.W.: PolSOM-a new method for multidimentional data visualization. Pattern Recognition 43, 1668–1675 (2010)
12. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: SOM toolbox for Matlab. Technical report, Laboratory of Computer and Information Science, Helsinki University of Technology (2000)
13. Kiviluoto, K.: Topology preservation in self-organizing maps. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 294–299 (1996)

# Ultrasonic Motor Control Based on Recurrent Fuzzy Neural Network Controller and General Regression Neural Network Controller

Tien-Chi Chen[1], Tsai-Jiun Ren[2], and Yi-Wei Lou[1]

[1] Department of Electrical Engineering, Kun Shan University, Tainan, Taiwan
tchichen@mail.ksu.edu.tw
[2] Department of Information Engineering, Kun Shan University, Tainan, Taiwan
cyrusren@mail.ksu.edu.tw

**Abstract.** The travelling-wave ultrasonic motor (TWUSM) has been used in industrial, medical, robotic and automotive applications. However, the TWUSM has the nonlinear characteristic and dead-zone problem which varies with many driving conditions. A novel control scheme, recurrent fuzzy neural network controller (RFNNC) and general regression neural network controller (GRNNC), for a TWUSM control is presented in this paper. The RFNNC provides real-time control such that the TWUSM output can tightly track the reference command. The adaptive updated RFNNC law is derived using Lyapunov theorem such that the system stability can be absolute. The GRNNC is appended to the RFNNC to compensate for the TWUSM dead-zone using a predefined set. The experimental results are shown to demonstrate the effectiveness of the proposed control scheme.

**Keywords:** Travelling-wave ultrasonic motor, TWUSM, Recurrent fuzzy neural network controller, RFNNC, Lyapunov theorem, General regression neural network controller, GRNNC, Dead-zone.

## 1 Introduction

The TWUSM is a new type of motor that is driven using the ultrasonic vibration force of piezoelectric elements. It has excellent performance and many useful features [1], such as high torque at low speed, quiet operation, light weight and compact size, quick response, wide velocity range, high efficiency, simple structure, easy production process and no electro-magnetic interference [2-3]. The TWUSM can be used in many industries such as industrial, medical, automotive, aerospace science and accurate positioning actuators [4].

The TWUSM is a new type of actuator with different control technique and operating principles than conventional electro-magnetic motors. Because the TWUSM is composed of piezoelectric ceramics instead of electro-magnetic windings in the motor structure [5], the TWUSM driving principles are based on the ultrasonic vibration of piezoelectric elements and mechanical frictional force [6].

The TWUSM motor dynamic model is very complicated with nonlinear characteristics, which vary with many driving conditions. The TWUSM parameters

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 291–305.
DOI: 10.1007/978-3-642-35638-4_19 

are nonlinear and time varying due to the increasing temperature and different motor drive operating conditions. These parameters include driving frequency, source voltage and load torque [1]. The TWUSM control characteristics are very complex to analyze and accurately model [7].

In general, the TWUSM drive and digital control system apply three independent control methods which are the drive frequency control, supplied voltage control and applied voltage phase difference control. In the phase difference control method the motor shows a variable dead-zone in the control input (phase difference of applied voltages) against the operating frequency. The dead-zone is due to a large static friction torque appearing at low speed. It is therefore difficult to design a perfect angle controller that can provide accurate control at all times. According to practical control issues, many speed controllers based on PI (proportional plus integral) controller using mathematical models of the motor have been reported.

Because the PI controller control algorithms are simple and the controllers have advantages such as high-stability margin and high-reliability when the controllers are tuned properly, the PI controller can be used to drive common motors. However, the PI controller cannot maintain these virtues at all times. The ultrasonic motor has nonlinear speed characteristics which vary with drive operating conditions. In order to overcome these difficulties, a dynamic controller with adjustable parameters and online learning algorithms is suggested for unknown or uncertain dynamic systems [8-9].

In the past few years there has been much research on neural network (NN) applications in order to deal with the nonlinearities and uncertainties in control systems [10-12]. According to NN structures, the NN can be classified mainly as feed-forward neural network (FNN) and recurrent neural network (RNN) [13]. It is well known that the FNN is capable of closely approximating continuous functions. The FNN conducts static mapping without the aid of delays. The FNN is unable to represent dynamic mapping. Although the FNN presented in much research is used to deal with delay and dynamic problems, The FNN requires a large number of neurons to express dynamic responses [14]. The weight calculations are not updated quickly and the function approximation is sensitive to the training data.

The RNN [15], on the other hand has superior capabilities compared to the FNN. The RNN exhibits dynamic response and information storing ability for later use. Since the recurrent neuron has an internal feedback loop, it captures the dynamic response of a system without external feedback through long delays. Thus, the RNN is a dynamic mapping and displays good control performance in the presence of unknowable and time-varying model dynamics [16]. As a result the RNN is better suited for dynamic systems than the FNN.

If the number of hidden neurons too many, the computation load becomes heavy so that the RNN is not suitable for online practical applications. If the number of hidden neurons is too few the learning performance may not be good enough to achieve the desired control performance. To solve this problem we propose a novel controller, the RFNNC, to maintain high accuracy.

The RFNNC has a number of attractive advantages compared to recurrent neural network control. For example, it has superior modelling performance due to local modelling and the fuzzy partition of the input space, linguistic dynamic fuzzy rule description, a learning based training example structure and parsimonious models with smaller parametric complexity [17]. The RFNNC combines fuzzy reasoning capability to handle uncertain information and the artificial recurrent neural network

capability to learn processes to deal with the nonlinearities and uncertainties that frustrate the TWUSM.

The RFNNC still presents a challenge considering the TWUSM as a plant. In the proposed RFNNC, the controller is effective in handling the small characteristic variations in the motor due to RFNNC connecting weight updating. However, the RFNNC is not able to fully compensate for the dead-zone effect and therefore the dynamic response deteriorates [18]. For these reasons an angle control scheme for the TWUSM with dead-zone compensation based on the RFNNC is presented in this research. The GRNNC is adopted to determine the dead-zone compensating input and decouple the RFNNC output. Because of the saturation reverse effect, phase difference control is not adequate for precise angle control. Therefore, the drive frequency must also be implemented, leading to a more accurate control strategy. The GRNNC based on RFNNC applies both the driving frequency and phase difference constructions as a dual-mode control method. The proposed controller can take the nonlinearity into account and compensate for the TWUSM dead zone. This approach also provides robust performance against parameter variations. The usefulness and validity of the proposed control scheme is examined through experimental results. The experimental results reveal that the GRNNC based on the RFNNC maintains stable performance under different motion conditions.

## 2 The Control Scheme

The TWUSM nonlinear dynamic system is expressed as:

$$\ddot{\theta} = f(\theta) + g(\theta)u(t) + d(t) \tag{1}$$

where $f(\cdot)$ and $g(\cdot)$ are unknown functions that are bounded. *u(t)* is the control input, *d(t)* is the external disturbance, and $\theta$ is rotor angle displacement of the TWUSM.

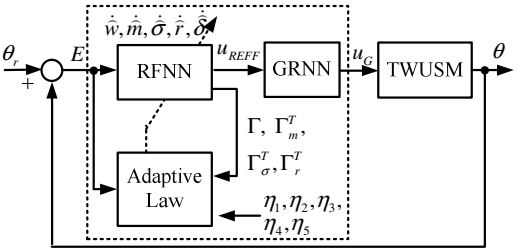



**Fig. 1.** The proposed control structure

The proposed control scheme, illustrated in Fig. 1, is composed of two main blocks, RFNNC and GRNNC. The RFNNC provides real-time control such that the TWUSM output can track the reference command $\theta_r$. The back-propagation algorithm is applied in the RFNNC to automatically adjust the parameters on-line. The RFNNC adaptive laws are derived using the Lyapunov Theorem such that the system stability can be absolute. $\Gamma$, $\Gamma_m^T$, $\Gamma_\sigma^T$, $\Gamma_r^T$ are the adaptive update law training

parameters and $\eta_1$, $\eta_2$, $\eta_3$, $\eta_4$, $\eta_5$ are the learning rates. The GRNNC is appended to the RFNNC to compensate for the TWUSM dead-zone using a predefined set. The GRNNC is designed to avoid the TWUSM dead-zone response.

### 2.1 Recurrent Fuzzy Neural Networks Controller

A controller is designed such that the TWUSM output can track the reference command. The tracking error vector is first defined as

$$E = \left[e,\ \dot{e}\right]^T \tag{2}$$

where $e = \theta_r - \theta$ is the angle tracking error. From (1) and (2), an ideal controller can be chosen as

$$u^*(t) = \frac{1}{g_n(\theta)}[\ddot{\theta}_r - f_n(\theta) - d_n(t) + K^T E] \tag{3}$$

where $K = \left[k_2, k_1\right]^T$, $k_1$ and $k_2$ are positive constants. Applying (2) to (3), the error dynamics can be expressed as

$$\ddot{e} + k_1\dot{e} + k_2 e = 0 \tag{4}$$

If $K$ is chosen to correspond to Hurwitz polynomial coefficients, it is a polynomial whose roots lie strictly in the open left half of the complex plane. A result is then achieved where $\lim_{t\to\infty} e(t) = 0$ for any initial conditions. Nevertheless, the functions $f(\theta)$ and $g(\theta)$ are not accurately known and the external load disturbances are perturbed. The ideal controller $u^*(t)$ cannot thus be practically implemented. Therefore, the RFNNC will be designed to approximate this ideal controller.

Figure 2 shows the four-layer RFNNC structure, which is comprised of an input layer, membership layer, rule layer and output layer. The superscript of symbol $y$ means the ordinal number of the layer, and the subscript of symbol $y$ means its number. The symbol $w$ expresses the weight of the signals. The RFNNC model is summarized as follows:

**(1) Input Layer.** The RFNNC inputs are $x_e^1 = e$ and $x_{\dot{e}}^1 = \dot{e}$. The input layer outputs are $y_{e,i}^1$ and $y_{\dot{e},i}^1$, which are equal to the inputs:

$$y_{e,i}^1 = x_e^1; \quad i = 1 \sim 3 \tag{5}$$

$$y_{\dot{e},i}^1 = x_{\dot{e}}^1; \quad i = 1 \sim 3 \tag{6}$$

**(2) Membership Layer.** There are three membership functions for $e$ and $\dot{e}$, respectively. The three signals are sent to calculate the degree belonging to the specified fuzzy set. The outputs $y_{e,i}^2$ and $y_{\dot{e},i}^2$ are as follows:

$$y_{e,i}^2 = \exp\left(-\left(\frac{y_{e,i}^1 - m_{e,i}}{\sigma_{e,i}}\right)^2\right); \quad i = 1 \sim 3 \tag{7}$$

$$y_{\dot{e},j}^{2} = \exp\left( -\left( \frac{y_{\dot{e},j}^{1} - m_{\dot{e},j}}{\sigma_{\dot{e},j}} \right)^{2} \right); \quad i = 1 \sim 3 \tag{8}$$

where $m$ and $\sigma$ are the mean and standard deviation of the Gaussian function. They express different RFNNC membership functions so the layer output can represent the degree the input belongs to the fuzzy rule.

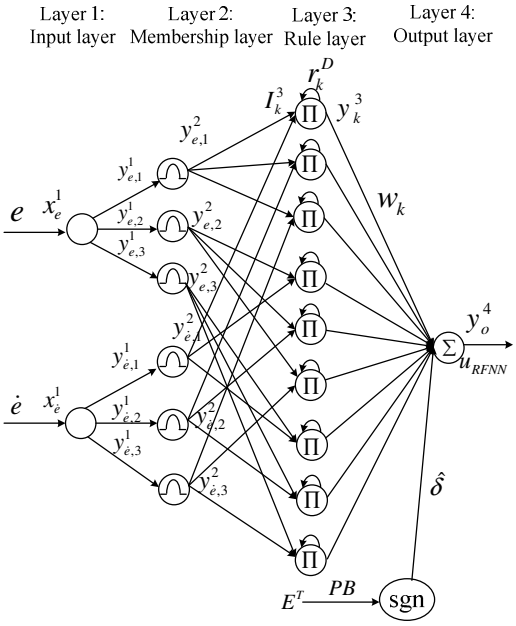



**Fig. 2.** The four-layer RFNNC structure

**(3) Rule Layer.** The outputs $y_k^3$ of the rule layer can be expressed as

$$y_k^3(t) = (1 + \frac{1}{1 + 100 \cdot \exp^{-10 r_k^D y_k^3(t-1)}}) y_{e,i}^2(t) y_{\dot{e},j}^2(t) \tag{9}$$

where $k = 3 \times (i-1) + j$, $i = 1 \sim 3$, $j = 1 \sim 3$ and $k = 1 \sim 9$. $r_k^D$ are the weights. The value of $y_k^3$ is always positive and between zero and two.

**(4) Output Layer.** The output $y_o^4$ of the RFNNC can be expressed as

$$\begin{aligned} u_{RFNN} = y_o^4 &= \sum_{k=1}^{9} w_k y_k^3 + \hat{\delta} \mathrm{sgn}(\mathrm{E}^{\mathrm{T}} \mathrm{PB}) \\ &= w^T \Gamma(x, m, \sigma, r) + \hat{\delta} \mathrm{sgn}(\mathrm{E}^{\mathrm{T}} \mathrm{PB}) \end{aligned} \tag{10}$$

where $\Gamma(x, m, \sigma, r) = \begin{bmatrix} y_1^3 & y_2^3 & \cdots & y_9^3 \end{bmatrix}^T$ fuzzy rule function vector, and $w = \begin{bmatrix} w_1 & w_2 & \cdots & w_9 \end{bmatrix}^T$ adjustable output weight vector, $\delta$ a small positive constant, and $E = [e, \dot{e}]^T$.

Assume that an optimal RFNNC exists to approximate the ideal control law such that

$$u^* = u^*_{RFNN}(e, w^*, m^*, \sigma^*, r^*) + \varepsilon = w^{*T}\Gamma^* + \varepsilon \tag{11}$$

where $\varepsilon$ is a minimum reconstructed error, $w^*$, $m^*$, $\sigma^*$, $r^*$ and $\Gamma^*$ are optimal parameters of $w$, $m$, $\sigma$, $r$ and $\Gamma$, respectively. Thus, the RFNNC control law is assumed to take the following form:

$$u = u_{RFNN} = \hat{w}^T\hat{\Gamma} + \hat{\delta}\mathrm{sgn}(\mathrm{E^T PB}) \tag{12}$$

where $\hat{w}$, $\hat{m}$, $\hat{\sigma}$, $\hat{r}$ and $\hat{\Gamma}$ are estimations of the optimal parameters, provided by algorithm tuning to be introduced later. Subtracting (12) from (11), an approximation error $\tilde{u}$ is obtained as

$$\begin{aligned}\tilde{u} &= u^* - u = w^{*T}\Gamma^* + \varepsilon - \hat{w}^T\hat{\Gamma} - \hat{\delta}\mathrm{sgn}(\mathrm{E^T PB}) \\ &= \tilde{w}^T\Gamma^* + \hat{w}^T\tilde{\Gamma} + \varepsilon - \hat{\delta}\mathrm{sgn}(\mathrm{E^T PB})\end{aligned} \tag{13}$$

where $\tilde{w} = w^* - \hat{w}$ and $\tilde{\Gamma} = \Gamma^* - \hat{\Gamma}$. The linearization technique transforms the multidimensional receptive-field basis functions into a partially linear form such that the expansion of $\tilde{\Gamma}$ in Taylor series becomes

$$\tilde{\Gamma} = \begin{bmatrix} \tilde{y}_1^3 & \dots & \tilde{y}_9^3 \end{bmatrix}^T = \Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r} + O_v \tag{14}$$

where $\tilde{y}_k^3 = y_k^{3*} - \hat{y}_k^3$, $y_k^{3*}$ the optimal parameter of $\hat{y}_k^3$, $\hat{y}_k^3$ the estimated parameter of $y_k^{3*}$, $\tilde{m} = m^* - \hat{m}$, $\tilde{\sigma} = \sigma^* - \hat{\sigma}$, $\tilde{r} = r^* - \hat{r}$, $O_v$ higher-order terms, $\Gamma_m = \left[\partial y_1^3/\partial m \ \dots \ \partial y_9^3/\partial m\right]^T|_{m=\hat{m}}$, $\Gamma_\sigma = \left[\partial y_1^3/\partial \sigma \ \dots \ \partial y_9^3/\partial \sigma\right]^T|_{\sigma=\hat{\sigma}}$ and $\Gamma_r = \left[\partial y_1^3/\partial r \ \dots \ \partial y_9^3/\partial r\right]^T|_{r=\hat{r}}$.

Equation (14) can be rewritten as

$$\Gamma^* = \hat{\Gamma} + \Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r} + O_v \tag{15}$$

Substituting (15) into (13), it can be rewritten as:

$$\begin{aligned}\tilde{u} &= \tilde{w}^T(\hat{\Gamma} + \Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r} + O_v) + \hat{w}^T(\Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r} + O_v) + \varepsilon - \hat{\delta}\mathrm{sgn}(\mathrm{E^T PB}) \\ &= \tilde{w}^T\hat{\Gamma} + \hat{w}^T(\Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r}) - \hat{\delta}\mathrm{sgn}(\mathrm{E^T PB}) + D\end{aligned} \tag{16}$$

where $D = \tilde{w}^T(\Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r}) + w^{*T}O_v + \varepsilon$ is the uncertainty term, and this term is assumed to be bounded with a small positive constant $\delta$ ($\text{let } |D| \le \delta$). From (1), (4) and (16), an error equation is obtained

$$\begin{aligned}\dot{E} &= AE + B(u^* - u) = AE + B\tilde{u} \\ &= AE + B\left[\tilde{w}^T\hat{\Gamma} + \hat{w}^T(\Gamma_m\tilde{m} + \Gamma_\sigma\tilde{\sigma} + \Gamma_r\tilde{r}) - \hat{\delta}\mathrm{sgn}(\mathrm{E^T PB}) + D\right]\end{aligned} \tag{17}$$

Consider the dynamic system represented by (1), if the RFNNC is designed as (12) with the adaptation laws for networks parameters shown in (18)–(22), the stability of the proposed RFNNC can be guaranteed. where $\eta_1$, $\eta_2$, $\eta_3$, $\eta_4$ and $\eta_5$ are strictly positive constants.

$$\dot{\hat{w}} = \eta_1 \hat{\Gamma} E^T PB \tag{18}$$

$$\dot{\hat{m}} = \eta_2 \Gamma_m^T \hat{w} E^T PB \tag{19}$$

$$\dot{\hat{\sigma}} = \eta_3 \Gamma_\sigma^T \hat{w} E^T PB \tag{20}$$

$$\dot{\hat{r}} = \eta_4 \Gamma_r^T \hat{w} E^T PB \tag{21}$$

$$\dot{\hat{\delta}} = \eta_5 \left| E^T PB \right| \tag{22}$$

**Proof**

Define a Lyapunov function candidate as

$$V(t) = \frac{1}{2} E^T PE + \frac{1}{2\eta_1} tr(\tilde{w}^T \tilde{w}) + \frac{1}{2\eta_2} \tilde{m}^T \tilde{m} + \frac{1}{2\eta_3} \tilde{\sigma}^T \tilde{\sigma} + \frac{1}{2\eta_4} \tilde{r}^T \tilde{r} + \frac{1}{2\eta_5} \tilde{\delta}^2 \tag{23}$$

where $P$ is a symmetric positive definite matrix which satisfies the following Lyapunov equation

$$A^T P + PA = -Q \tag{24}$$

where $Q$ is a positive definite matrix. Here, the uncertainty bound estimation error is defined as $\tilde{\delta} = \delta - \hat{\delta}$. Taking the Lyapunov function differential (23) and using (16) and (24), it is concluded that

$$\begin{aligned} \dot{V}(t) = & -\frac{1}{2} E^T QE + E^T PB \left[ \tilde{w}^T \hat{\Gamma} + \hat{w}^T (\Gamma_m \tilde{m} + \Gamma_\sigma \tilde{\sigma} + \Gamma_r \tilde{r}) - u_C + D \right] \\ & - \frac{1}{\eta_1} \tilde{w}^T \dot{\hat{w}} - \frac{1}{\eta_2} \dot{\hat{m}}^T \tilde{m} - \frac{1}{\eta_3} \dot{\hat{\sigma}}^T \tilde{\sigma} - \frac{1}{\eta_4} \dot{\hat{r}}^T \tilde{r} - \frac{1}{\eta_5} \tilde{\delta} \dot{\hat{\delta}} \end{aligned} \tag{25}$$

Take (18)-(22) into (25), the derivative of $V$ can be rewritten as

$$\begin{aligned} \dot{V}(t) &= -\frac{1}{2} E^T QE + E^T PBD - E^T PBu_C - \frac{1}{\eta_5} (\delta - \hat{\delta}) \dot{\hat{\delta}} \\ &\le -\frac{1}{2} E^T QE - \left| E^T PB \right| (\delta - |D|) \le 0 \end{aligned} \tag{26}$$

Therefore regardless what the situation is, the derivative of $V$ respect to time is smaller than zero. $\dot{V}(t) \le 0$ is negative semi-definite (i.e., $\dot{V}(t) \le \dot{V}(0)$), which implies $E$, $\tilde{w}$, $\tilde{m}$, $\tilde{\sigma}$, $\tilde{\delta}$ and $\tilde{r}$ are bounded. Let function $F(t) = E^T QE / 2 \le -\dot{V}(t)$, and integrate function with respect to time.

Because $V(0)$ is bounded, and $V(t)$ is bounded, the following result is obtained:

$$\lim_{t \to \infty} \int_0^t F(\tau) d\tau < \infty \tag{27}$$

Since $\dot{F}(t)$ is bounded, by Barbalat's Lemma it can be shown that $\lim_{t \to \infty} F(t) = 0$. This implies that $E(t)$ will converge to zero as $t \to \infty$. As a result, the stability of the proposed control system can be guaranteed.

### 2.2 Convergence Analysis of RFNNC

Although the stability of the adaptive RFNNC can be guaranteed, the parameters $\hat{w}$, $\hat{m}$, $\hat{\sigma}$ and $\hat{r}$ in (18)–(21) cannot be guaranteed within a bound value. The RFNNC output is bounded, whether the means, the standard deviation of the Gaussian function and weights are bounded. The constraint sets $\overline{w}$, $\overline{m}$, $\overline{\sigma}$ and $\overline{r}$ are defined respectively

$$U_w = \left\{ \left\| \hat{w} \right\| \le \overline{w} \right\} \tag{28}$$

$$U_m = \left\{ \left\| \hat{m} \right\| \le \overline{m} \right\} \tag{29}$$

$$U_\sigma = \left\{ \left\| \hat{\sigma} \right\| \le \overline{\sigma} \right\} \tag{30}$$

$$U_r = \left\{ \left\| \hat{r} \right\| \le \overline{r} \right\} \tag{31}$$

where $\| \cdot \|$ is a two-norm of vector, $\overline{w}$, $\overline{m}$, $\overline{\sigma}$ and $\overline{r}$ are positive constants, and the adaptive laws (18)-(21) can be modified as follows

$$\dot{\hat{w}} = \begin{cases} \eta_1 \hat{\Gamma} E^T PB, & \text{if } \left\| \hat{w} \right\| < \overline{w} \text{ or } \left( \left\| \hat{w} \right\| = \overline{w} \text{ and } E^T PB\hat{w}^T \hat{\Gamma} \le 0 \right) \\ \eta_1 \hat{\Gamma} E^T PB - \eta_1 \hat{\Gamma} E^T PB \dfrac{\hat{w}\hat{w}^T}{\left\| \hat{w} \right\|^2}, & \text{if } \left\| \hat{w} \right\| = \overline{w} \text{ and } E^T PB\hat{w}^T \hat{\Gamma} > 0 \end{cases} \tag{32}$$

$$\dot{\hat{m}} = \begin{cases} \eta_2 \Gamma_m^T \hat{w} E^T PB, & \text{if } \left\| \hat{m} \right\| < \overline{m} \text{ or } \left( \left\| \hat{m} \right\| = \overline{m} \text{ and } E^T PB\hat{w}^T \Gamma_m \hat{m} \le 0 \right) \\ \eta_2 \Gamma_m^T \hat{w} E^T PB - \eta_2 \Gamma_m^T \hat{w} E^T PB \dfrac{\hat{m}\hat{m}^T}{\left\| \hat{m} \right\|^2}, & \text{if } \left\| \hat{m} \right\| = \overline{m} \text{ and } E^T PB\hat{w}^T \Gamma_m \hat{m} > 0 \end{cases} \tag{33}$$

$$\dot{\hat{\sigma}} = \begin{cases} \eta_3 \Gamma_\sigma^T \hat{w} E^T PB, & \text{if } \left\| \hat{\sigma} \right\| < \overline{\sigma} \text{ or } \left( \left\| \hat{\sigma} \right\| = \overline{\sigma} \text{ and } E^T PB\hat{w}^T \Gamma_\sigma \hat{\sigma} \le 0 \right) \\ \eta_3 \Gamma_\sigma^T \hat{w} E^T PB - \eta_3 \Gamma_\sigma^T \hat{w} E^T PB \dfrac{\hat{\sigma}\hat{\sigma}^T}{\left\| \hat{\sigma} \right\|^2}, & \text{if } \left\| \hat{\sigma} \right\| = \overline{\sigma} \text{ and } E^T PB\hat{w}^T \Gamma_\sigma \hat{\sigma} > 0 \end{cases} \tag{34}$$

$$\dot{\hat{r}} = \begin{cases} \eta_4 \Gamma_r^T \hat{w} E^T PB, & \text{if } \left\| \hat{r} \right\| < \overline{r} \text{ or } \left( \left\| \hat{r} \right\| = \overline{r} \text{ and } E^T PB\hat{w}^T \Gamma_r \hat{r} \le 0 \right) \\ \eta_4 \Gamma_r^T \hat{w} E^T PB - \eta_4 \Gamma_r^T \hat{w} E^T PB \dfrac{\hat{r}\hat{r}^T}{\left\| \hat{r} \right\|^2}, & \text{if } \left\| \hat{r} \right\| = \overline{r} \text{ and } E^T PB\hat{w}^T \Gamma_r \hat{r} > 0 \end{cases} \tag{35}$$

If the initial values $\hat{w}(0) \in U_w$, $\hat{m}(0) \in U_m$, $\hat{\sigma}(0) \in U_\sigma$ and $\hat{r}(0) \in U_r$ then the adaptive laws (32)-(35)guarantee that $\hat{w}(t) \in U_w$, $\hat{m}(t) \in U_m$, $\hat{\sigma}(t) \in U_\sigma$ and $\hat{r}(t) \in U_r$ for all $t \ge 0$.

Define a Lyapunov function as

$$v_w = \frac{1}{2} \hat{w}^T \hat{w} \tag{36}$$

The derivative of the Lyapunov function is presented as

$$\dot{v}_w = \hat{w}^T \dot{\hat{w}} \tag{37}$$

Assume the first line of (32) is true, either $\left\| \hat{w} \right\| < \overline{w}$ or $\left( \left\| \hat{w} \right\| = \overline{w} \text{ and } E^T PB\hat{w}^T \hat{\Gamma} \le 0 \right)$. Substituting the first line of (32) into (37), which becomes $\dot{v}_w = \eta_1 E^T PB\hat{w}^T \hat{\Gamma} \le 0$. As a

result, $\|\hat{w}\| \le \bar{w}$ is guaranteed. In addition, when $\|\hat{w}\| = \bar{w}$ *and* $E^T PB\hat{w}^T\hat{\Gamma} > 0$, $\dot{v}_w = \eta_1 E^T PB\hat{w}^T\hat{\Gamma} - \eta_1 E^T PB\dfrac{\hat{w}^T\hat{w}}{\|\hat{w}\|^2}\hat{w}^T\hat{\Gamma} = 0$. That $\|\hat{w}\| \le \bar{w}$ can be also assured. Thereby, the initial value of $\hat{w}$ is bounded, $\|\hat{w}\|$ is bounded by the constraint set $\bar{w}$ for $t \ge 0$. Similarly, it can be proved that $\|\hat{m}\|$ is bounded by the constraint set $\bar{m}$, $\|\hat{\sigma}\|$ is bounded by the constraint set $\bar{\sigma}$ and $\|\hat{r}\|$ is bounded by the constraint set $\bar{r}$ for $t \ge 0$.

When the condition $\|\hat{w}\| < \bar{w}$ or $\left(\|\hat{w}\| = \bar{w}\ and\ E^T PB\hat{w}^T\hat{\Gamma} \le 0\right)$, $\|\hat{m}\| < \bar{m}$ or $\left(\|\hat{m}\| = \bar{m}\ and\ E^T PB\hat{w}^T\Gamma_m\hat{m} \le 0\right)$, $\|\hat{\sigma}\| < \bar{\sigma}$ or $\left(\|\hat{\sigma}\| = \bar{\sigma}\ and\ E^T PB\hat{w}^T\Gamma_\sigma\hat{\sigma} \le 0\right)$, $\|\hat{r}\| < \bar{r}$ or $\left(\|\hat{r}\| = \bar{r}\ and\ E^T PB\hat{w}^T\Gamma_r\hat{r} \le 0\right)$, the stability analysis the same as (33), (34) and (35). In the other situation, the condition $\|\hat{w}\| = \bar{w}$ and $E^T PB\hat{w}^T\hat{\Gamma} > 0$, $\|\hat{m}\| = \bar{m}$ and $E^T PB\hat{w}^T\Gamma_m\hat{m} > 0$, $\|\hat{\sigma}\| = \bar{\sigma}$ and $E^T PB\hat{w}^T\Gamma_\sigma\hat{\sigma} > 0$, $\|\hat{r}\| = \bar{r}$ and $E^T PB\hat{w}^T\Gamma_r\hat{r} > 0$ is occurred, the Lyapunov function can be rewritten as follows

$$\begin{aligned}
\dot{v}_w &= -\frac{1}{2}E^T QE + E^T PB\left(\tilde{w}^T\hat{\Gamma} + \hat{w}^T\Gamma_m\tilde{m} + \hat{w}^T\Gamma_\sigma\tilde{\sigma} + \hat{w}^T\Gamma_r\tilde{r}) + D - u_C\right) - \frac{1}{\eta_1}\tilde{w}^T\dot{\hat{w}} \\
&\quad - \frac{1}{\eta_2}\dot{\hat{m}}^T\tilde{m} - \frac{1}{\eta_3}\dot{\hat{\sigma}}^T\tilde{\sigma} - \frac{1}{\eta_4}\dot{\hat{r}}^T\tilde{r} - \frac{1}{\eta_5}\tilde{\delta}\dot{\hat{\delta}} \\
&= -\frac{1}{2}E^T QE + E^T PB\left(D - u_C\right) + E^T PB\frac{\tilde{w}^T\hat{w}}{\|\hat{w}\|^2}\hat{w}^T\hat{\Gamma} + (\Gamma_m^T\hat{w})^T E^T PB\hat{m}\frac{\hat{m}^T\tilde{m}}{\|\hat{m}\|^2} \\
&\quad + (\Gamma_\sigma^T\hat{w})^T E^T PB\hat{\sigma}\frac{\hat{\sigma}^T\tilde{\sigma}}{\|\hat{\sigma}\|^2} + (\Gamma_r^T\hat{w})^T E^T PB\hat{r}\frac{\hat{r}^T\tilde{r}}{\|\hat{r}\|^2} - \frac{1}{\eta_5}\tilde{\delta}\dot{\hat{\delta}}
\end{aligned} \tag{38}$$

Equation $\tilde{w}^T\hat{w} = \left(\|w^*\|^2 - \|\hat{w}\|^2 - \|\tilde{w}\|^2\right)/2 < 0$, which is according to $\|\hat{w}\| = \bar{w} > \|\hat{w}^*\|$. Similarly, $\|\hat{m}\| = \bar{m} > \|\hat{m}^*\|$, $\|\hat{\sigma}\| = \bar{\sigma} > \|\hat{\sigma}^*\|$ and $\|\hat{r}\| = \bar{r} > \|\hat{r}^*\|$ can be proven. It is finally obtained as

$$\begin{aligned}
\dot{v}_w &= -\frac{1}{2}E^T QE + E^T PBD - E^T PBu_C + E^T PB\frac{\tilde{w}^T\hat{w}}{\|\hat{w}\|^2}\hat{w}^T\hat{\Gamma} + \Gamma_m^T\hat{w}E^T PB\hat{m}\frac{\hat{m}^T\tilde{m}}{\|\hat{m}\|^2} \\
&\quad + \Gamma_\sigma^T\hat{w}E^T PB\hat{\sigma}\frac{\hat{\sigma}^T\tilde{\sigma}}{\|\hat{\sigma}\|^2} + \Gamma_r^T\hat{w}E^T PB\hat{r}\frac{\hat{r}^T\tilde{r}}{\|\hat{r}\|^2} - \frac{1}{\eta_5}\tilde{\delta}\dot{\hat{\delta}} \\
&\le -\frac{1}{2}E^T QE + E^T PB\frac{\left(\|w^*\|^2 - \|\hat{w}\|^2 - \|\tilde{w}\|^2\right)}{\|\hat{w}\|^2}\hat{w}^T\hat{\Gamma} \\
&\quad + \frac{1}{2}(\Gamma_m^T\hat{w})^T E^T PB\hat{m}\frac{\left(\|m^*\|^2 - \|\hat{m}\|^2 - \|\tilde{m}\|^2\right)}{\|\hat{m}\|^2} \\
&\quad + \frac{1}{2}(\Gamma_\sigma^T\hat{w})^T E^T PB\hat{\sigma}\frac{\left(\|\sigma^*\|^2 - \|\hat{\sigma}\|^2 - \|\tilde{\sigma}\|^2\right)}{\|\hat{\sigma}\|^2} \\
&\quad + \frac{1}{2}(\Gamma_r^T\hat{w})^T E^T PB\hat{r}\frac{\left(\|r^*\|^2 - \|\hat{r}\|^2 - \|\tilde{r}\|^2\right)}{\|\hat{r}\|^2} \\
&\le -\frac{1}{2}E^T QE \le 0
\end{aligned} \tag{39}$$

Using the same discussion shown in the previous section, the stability property can also be guaranteed since $E \to 0$ as $t \to 0$.

### 2.3 General Regression Neural Networks Controller

As a common nonlinear problem, a dead-zone often appears in the control system, which not only makes a steady-sate error, it also deteriorates the dynamic quality of the control systems. The GRNNC is proposed to solve this problem. The GRNNC is a powerful regression tool with a dynamic network structure and the training speed is extremely fast. Due to the simplicity of the network structure and ease of implementation, it can be widely applied to a variety of fields.

The GRNNC structure, shown in Fig. 3, is suggested for the system input nonlinear compensation. The input $u$ is the RFNNC output, $W_G^1$ is the weight of the hidden layer, $W_G^2$ is the weight of the output layer, $a$ is the output of the hidden layer, $u_G$ is the output of the output layer.

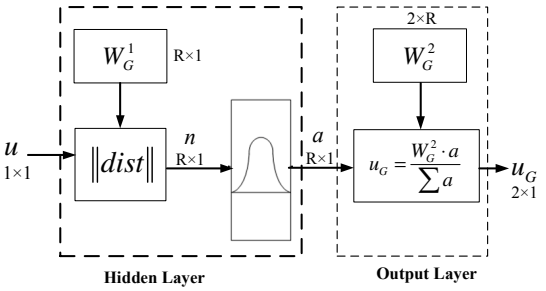



**Fig. 3.** The GRNNC structure

The GRNNC is composed of two layers, the hidden layer and the output layer. The input $u$ of the GRNNC means a torque calculated by the RFNNC. The outcome $n$ of $\|dist\|$ represents the Euclidean distance between the input $u$ and each element of $W_G^1$. $n$ is passed using a Gaussian function. When the Euclidean distance between $u$ and $W_G^1$ is far, the output element $a$ approaches zero. On the other hand, if the Euclidean distance is short the output element $a$ approaches one. The Gaussian function is

$$a = \exp\left(-\left(\frac{n-m}{\sigma}\right)^2\right) \tag{40}$$

where $m$ and $\sigma$ are the center and standard deviation of the Gaussian function, respectively. In order to increase the discrimination and have better performance, the standard deviation $\sigma$ value of the Gaussian function is chosen as low.

The relation function of the output layer can be expressed as

$$u_G = \frac{W_G^2 \cdot a}{\sum a} \tag{41}$$

The output vector of the hidden layer $a$ is multiplied with appropriate weights $W_G^2$ to sum up the output $u_G$ of the GRNNC. The output $u_G$ is composed of frequency control $u_f$ and phase control $u_P$ and expressed as

$$u_G = \begin{bmatrix} u_f & u_p \end{bmatrix}^T \tag{42}$$

Applying the GRNNC, the dead-zone of the TWUSM will be compensated as desired.

## 3 Experiments

Experiments are required to prove the feasibility of the proposed scheme. Figure 4 shows the experimental structure, which includes TMS320F2812 digital signal processor (DSP), TWUSM driver and TWUSM. The TMS320F2812 DSP experiment board is applied as the computing core. The DSP program was coded in C language. After compilation, assembly and link, the execution file is generated by C2000 code composer (CCS). The execution file is executed in the same windows interface.

In these experiments three different controllers were chosen for comparison.

(i) The proposed control scheme, RFNNC and GRNNC.

(ii) The RFNNC only, without GRNNC. The control algorithm of RFNNC only is the same as RFNNC of the proposed control scheme.

(iii) The PI controller. The PI controller is the one of the most used controller in linear system. The control PI controller has important advantages such as a simple structure and easy to design. Therefore, PI controllers are used widely in industrial applications. Owing to the absence of the TWUSM mathematical model, the PI controller parameters are chosen by trial and error in such a way that the optimal performance occurs at rated conditions. A block diagram of the angle control system for an ultrasonic motor using a PI controller is shown in Fig. 5. Where $\theta_r$ and $\theta$ are the command and rotor angle, *e(k)* is the tracking error, $u_f$ is the frequency command, $u_p$ is the phase different command, respectively. The PI controller parameters were selected as $K_P = 1000$ and $K_I = 100$.

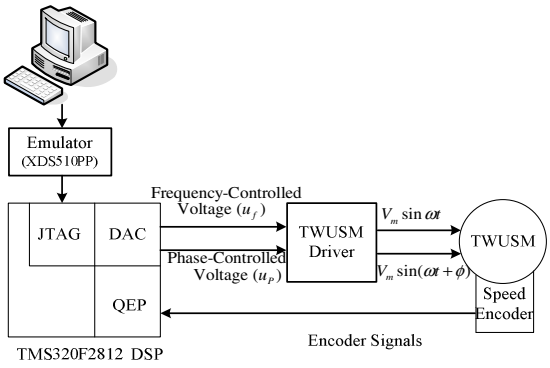



**Fig. 4.** The experiment structure

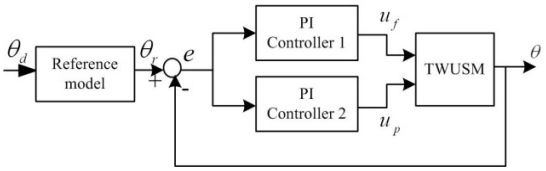



**Fig. 5.** The block diagram of the dual-mode PI control

Figures 6 to 8 show the experimental results for the proposed control scheme, the RFNNC only, and the PI control respectively, for a periodic square angle command from -90 to 90 degrees. Figures 9 to 11 show the experimental results for the proposed control scheme, the RFNNC only, and the PI control respectively, for a sinusoidal angle command from -90 to 90 degrees. Figure (a) shows the TWUSM angle response and speed response. Figure (b) shows the angle error between angle command and angle response.

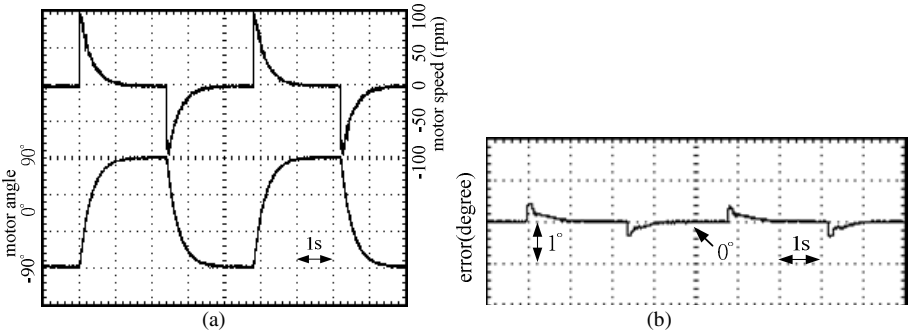



**Fig. 6.** The experimental result for the proposed control scheme for a periodic angle square command from -90 to 90 degrees

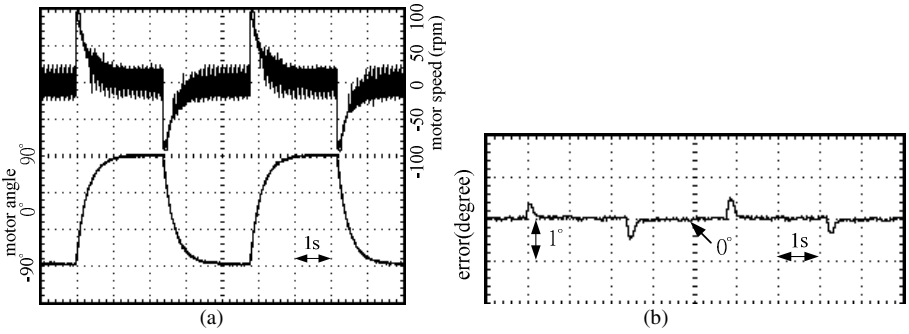



**Fig. 7.** The RFNNC only experimental result for a periodic square angle command from -90 to 90 degrees

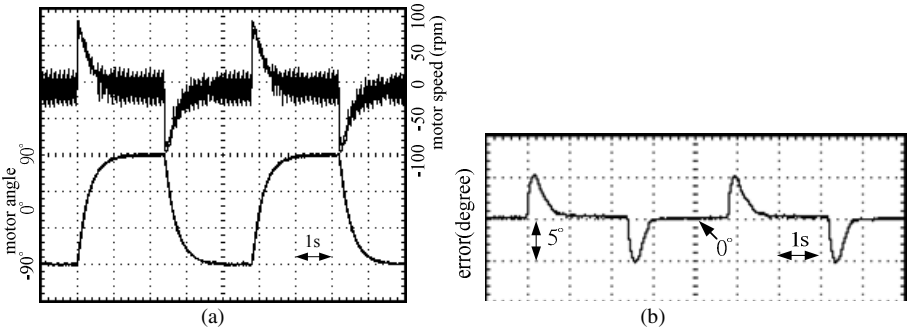



**Fig. 8.** The experimental result for the PI control for a periodic square angle command from -90 to 90 degrees

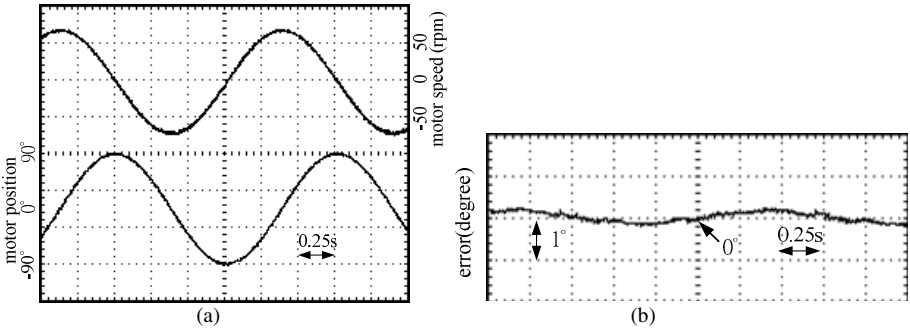



**Fig. 9.** The experimental result for the proposed control scheme for a sinusoidal angle command from -90 to 90 degrees

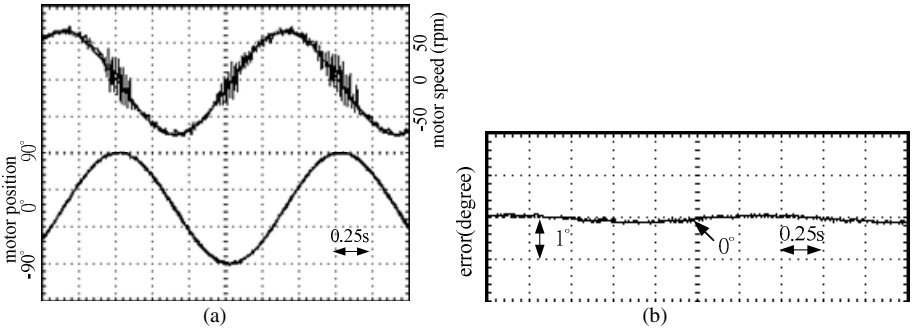



**Fig. 10.** The RFNNC only experimental result for a sinusoidal angle command from -90 to 90 degrees

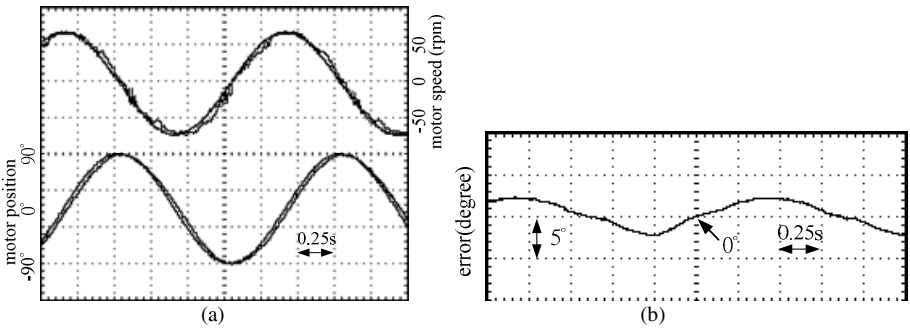



**Fig. 11.** The experimental result for the PI control for a sinusoidal angle command from -90 to 90 degrees

Observing the experimental results for the proposed control scheme in Figs. 6 and 9 the tracking errors for both can converge to an acceptable region and the control performance is excellent. The proposed controller retains control performance and has no dead-zone.

The RFNNC only experimental results in Figs. 7 and 10 show that the tracking error is similar to the proposed control scheme. However, the RFNNC drawbacks interfere with the dead-zone and the motor speed has a serious chattering phenomenon at slow speed near zero.

Figures 8 and 11 illustrate that the PI controller has a chattering phenomenon like the RFNNC only and a larger tracking error.

## 4 Conclusions

This paper presented a proposed control scheme, RFNNC and GRNNC, applied to the TWUSM. Many concepts such as controller design and the stability analysis of the controller are introduced. The experiment results show that the proposed control scheme is feasible and the performance is better than conventional control methods.

The proposed control scheme includes the RFNNC and GRNNC. The RFNNC is designed to track the reference angle. The membership function and weight variables can be updated using adaptive algorithms. Moreover, all parameters proposed RFNNC parameters are tuned in the Lyapunov sense; thus, the system stability can be guaranteed. In the RFNNC a compensated controller is designed to recover the residual part of the approximation error. The GRNNC is appended to the RFNNC to compensate for the TWUSM system dead zone using a predefined set. The GRNNC can successfully avoid the TWUSM dead-zone problem. The experimental results verify that the proposed controller can control the system well.

**Acknowledgements.** The authors would like to express their appreciation to NSC for supporting under contact NSC 97-2221-E-168 -050 -MY3.

## References


1. Sashida, T., Kenjo, T.: An introduction to ultrasonic motors. Clarendon Press, Oxford (1993)
2. Ueha, S., Tomikawa, Y.: Ultrasonic motors theory and applications. Clarendon Press, Oxford (1993)
3. Uchino, K.: Piezoelectric actuators and ultrasonic motors. Kluwer Academic Publishers (1997)
4. Huafeng, L., Chunsheng, Z., Chenglin, G.: Precise position control of ultrasonic motor using fuzzy control with dead-zone compensation. J. of Electrical Engineering 56(1-2), 49–52 (2005)
5. Uchino, K.: Piezoelectric ultrasonic motors: overview. Smart Materials and Structures 7, 273–285 (1998)
6. Chen, T.C., Yu, C.H., Tsai, M.C.: A novel driver with adjustable frequency and phase for travelling-wave type ultrasonic motor. Journal of the Chinese Institute of Engineers 31(4), 709–713 (2008)
7. Hagood, N.W., Mcfarland, A.J.: Modeling of a piezoelectric rotary ultrasonic motor. IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control 42(2), 210–224 (1995)
8. Bal, G., Bekiroglu, E.: Servo speed control of travelling-wave ultrasonic motor using digital signal processor. Sensor and Actuators A 109, 212–219 (2004)
9. Bal, G., Bekiroglu, E.: A highly effective load adaptive servo drive system for speed control of travelling-wave ultrasonic motor. IEEE Trans. on Power Electronics 20(5), 1143–1149 (2005)
10. Alessandri, A., Cervellera, C., Sanguineti, M.: Design of asymptotic estimators: an approach based on neural networks and nonlinear programming. IEEE Trans. on Neural Networks 18(1), 86–96 (2007)
11. Liu, M.: Delayed standard neural network models for control systems. IEEE Trans. on Neural Networks 18(5), 1376–1391 (2007)
12. Abiyev, R.H., Kaynak, O.: Fuzzy wavelet neural networks for identification and control of dynamic plants-A novel structure and a comparative study. IEEE Trans. on Industrial Electronics 55(8), 3133–3140 (2008)
13. Lin, C.M., Hsu, C.F.: Recurrent neural network based adaptive -backstepping control for induction servomotors. IEEE Trans. on Industrial Electronics 52(6), 1677–1684 (2005)
14. Ku, C.C., Lee, K.Y.: Diagonal recurrent neural networks for dynamic systems control. IEEE Trans. on Neural Networks 6(1), 144–156 (1995)
15. Juang, C.F., Huang, R.B., Lin, Y.Y.: A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing. IEEE Trans. on Fuzzy Systems 17(5), 1092–1105 (2009)
16. Stavrakouds, D.G., Theochairs, J.B.: Pipelined recurrent fuzzy neural networks for nonlinear adaptive speech prediction. IEEE Trans. on Systems, Man and Cybernetics, Part B 37(5), 1305–1320 (2007)
17. Lin, C.J., Chen, C.H.: Identification and prediction using recurrent compensatory neuro-fuzzy systems. Fuzzy Sets and Systems 150(2), 307–330 (2005)
18. Senjyu, T., Kashiwagi, T., Uezato, K.: Position control of ultrasonic motors using MRAC with deadzone compensation. IEEE Trans. on Power Electronics 17(2), 265–272 (2002)

# A Hybrid Model for Navigation Satellite Clock Error Prediction

Bo Xu[1], Ying Wang[2], and Xuhai Yang[3]

[1] School of Astronomy & Space Science, Nanjing University, Nanjing 210093, China
xubo@nju.edu.cn
[2] Aerospace System Engineering Shanghai, Shanghai 201108, China
winewy1118@yahoo.com.cn
[3] National Time Service Centre, Xi'an 710600, China
yangxh@nts.ac.cn

**Abstract.** In order to improve navigation satellite clock error prediction accuracy, a hybrid model is proposed in this paper. According to the physics property of atomic clock, the model firstly fits the clock error series by polynomial model. Then it models for polynomial fitting residuals, using functional network. The functional network structure is defined by wavelet de-noising and phase space reconstruction. Finally the GPS satellites are taken for example and four separate predict tests are done, the simulation results show that the proposed method can fit and predict the clock error series effectively, whose predict accuracy is better than those of IGU-P and conventional methods.

**Keywords:** Clock error predict, Functional network, Phase space construction, Chaotic, Hybrid model.

## 1 Introduction

The performance of a navigation satellite is related to the behavior of the atomic clocks hosted on the satellite. The real-time and reliable prediction of the behavior of such clocks is absolutely necessary for providing precise navigation performance and optimizing the interval between uploading of the corrections to the satellite clocks. Take Global Navigation Satellite System (GNSS) for example, the International GNSS Service (IGS), along with a multinational membership of organizations and agencies, provides Global Positioning System (GPS) orbits and clocks, tracking data, and data products online to meet the objectives of a wide range of scientific and engineering applications and studies. The accuracy of the satellite and station clocks is announced to be better than 0.1 ns. The accuracy of orbit is less than 5 cm [1]. In fact, these high-accuracy data are not available in real time but a posteriori, with a delay up to 13 days. The broadcast ephemeris is realized in real time, but the accuracy reaches 5 ns.

Many papers have dealt with the prediction problem. Zhang et al. [2] constructed a model which includes a quadratic polynomial and the periodic terms. Cui and Jiao [3] introduced the grey system into the prediction of the clock error and obtained better results. Xu and Zeng [4] proposed ARIMA (0, 2, q) model to predict the clock error

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 307–316.
DOI: 10.1007/978-3-642-35638-4_20 

and gained a series of important achievements. However, further studies show that there exist some limitations in the classical methods of navigation satellite clock error prediction. On the basis of exploring the limitations of the traditional models, we present a novel research on the navigation satellite clock error prediction based on the hybrid model, which is the combination of polynomials and functional network.

Castillo et al. [5] introduced functional network as a generalization of the standard neural network. The neural networks are basically driven by data, but the functional network may be considered more as problem-driven model than as data-driven model. Functional network has been successfully demonstrated in some sample applications, e.g. to extract information masked by chaos [6], and has been used for nonlinear system identification [7]. It has also been used for predicting fresh and hardened properties of self-compacting concretes [8].

The paper is organized as follows: Section 2 is a brief description of the atomic clock's physical property. Section 3 describes the clock error prediction model, which includes the predict mechanism, mathematical representation of the functional network, and the determination procedure of the functional network structure. In Section 4, four separate tests were carried out on the materials of the GPS satellite clock error. And the results are compared with the conventional grey method (GM), the quadratic polynomial method (QPM); the quadratic polynomial with periodic term method (QPPTM), the autoregressive integrated moving average method (ARIMA) and the Kalman filter method (KFM). The results and some diagrams and discussions from the simulation are also presented in this section. Finally, some conclusions are presented in Section 5.

## 2 The Physical Property of Atomic Clock on Board

The output of the atomic clock can be expressed as,

$$V(t) = \left[V_0 + \varepsilon(t)\right] \sin\left[2\pi f_0 t + \varphi(t)\right] \tag{1}$$

where $V_0$ is the nominal amplitude, $f_0$ is the nominal frequency, $\varepsilon(t)$ is the fluctuation of amplitude and $\varphi(t)$ is the fluctuation of phase. The instantaneous phase of the clock signal is $\phi(t) = 2\pi f_0 t + \varphi(t)$ . Take derivatives of the instantaneous phase $\phi(t)$, and it is the instantaneous angular frequency. Thus the instantaneous frequency can be written as,

$$f(t) = f_0 + \frac{1}{2\pi}\dot{\varphi}(t) \tag{2}$$

where $\dot{\varphi}(t)$ represents instantaneous frequency bias. The relative phase bias $x(t)$ and the relative frequency bias $y(t)$ can be expressed as,

$$x(t) = \frac{\varphi(t)}{2\pi f_0}, y(t) = \frac{\dot{\varphi}(t)}{2\pi f_0} \tag{3}$$

Usually the relative phase bias $x(t)$ can be modeled as,

$$x(t) = x_0 + y_0 t + \frac{1}{2} a t^2 + \psi(t) \tag{4}$$

where, $x_0 = x(0)$, $y_0 = y(0)$ represent the phase bias and frequency bias at initial time $t_0$, $a$ represents the linear rate of the frequency bias $y(t)$. Actually the value of $a$ can be set to 0 for cesium atomic clocks [9].

# 3 The Clock Error Prediction Model Based on Hybrid Model

## 3.1 The Prediction Mechanism

The navigation satellite clock error is a discrete-time series from nonlinear system, the change of which is a comprehensive reflection of the interaction of many factors. Ke et al. [10], starting from the establishment of nonlinear dynamics model for atomic clocks, for the first time introduced chaos theory to the analysis of atomic clock error series and used the fractal theory to describe the complexity of the clock error series.

In this paper, we use polynomial model to extract the clock series trend, and then use the functional network to model the residuals. As the functional network has powerful capacity of parallel processing and nonlinear mapping, we can use it to study the chaotic time series, and then to predict or control. On the other side, the chaotic time series has definite internal regularity, which makes the system seem to have some correlation. This kind of information processing method is just what the functional network excels, while it is difficult for conventional analytical methods. Therefore, we choose the functional network to predict the chaotic time series.

## 3.2 Functional Network

Functional network corresponds to the functional transformation; its topology describes a function transformation system. Generally, a functional network consists of several elements, which includes one layer of input storing neurons, one layer of output storing neurons, one (or more) layers of processing neurons, optional layers of intermediate storing neurons and a set of direct links between them [11]. Figure 1 shows a typical architecture of a functional network.

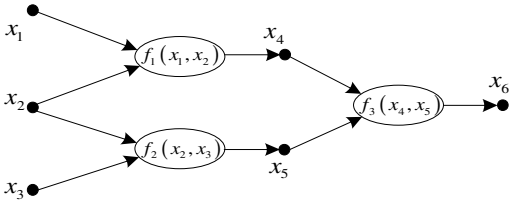



**Fig. 1.** Functional network architecture

To work with functional networks, in addition to the data information, it is important to understand the problem to be solved. Since the selection of the topology is normally based on the properties, which usually lead to a clear and single network

structure. From the different possible functional networks, the separable functional network is a simple family with many applications. It uses a functional expression that combines the separated efforts of input variables.

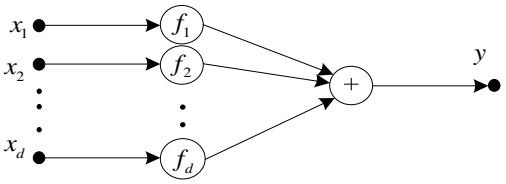



**Fig. 2.** The general separable functional network architecture

Figure 2 depicts the topology of a separable functional network. The relationship between the inputs and outputs can be defined mathematically as follows,

$$y = \sum_{k=1}^{d} \hat{f}_k(x_k) = \sum_{k=1}^{d} \sum_{j=1}^{m_k} a_{kj} \phi_{kj}(x_k) \tag{5}$$

where $a_{kj}$ are parameters of the network, and the functional neurons $f_1, f_2, ..., f_d$ are composed of the linear combination of the basis function family $\Phi_k$. The basis function family can be a polynomial family, a Fourier series or any other set of linearly independent functions.

Assume that we have training data set as $\{(x_{i,1}, x_{i,2}, ..., x_{i,d}; y_i), i = 1, 2, ..., N\}$, where $N$ is the number of training data set and it satisfies $N > d$. The training error for the functional network can be defined as,

$$e_i = y_i - \sum_{k=1}^{d} \sum_{j=1}^{m_k} a_{kj} \varphi_{kj}(x_{i,k}) \tag{6}$$

For a unique representation of the functional network, we must add an initial functional condition,

$$f_k(x_0) = \sum_{j=1}^{m_k} a_{kj} \varphi_{kj}(x_{k0}) = v_{k0}, k = 1, 2, ..., d \tag{7}$$

Using a Lagrange multiplier, the objective function can be written as,

$$Q_\lambda = \sum_{i=1}^{N} \left( y_i - \sum_{k=1}^{d} \sum_{j=1}^{m_k} a_{kj} \varphi_{kj}(x_{i,k}) \right)^2 + \sum_{k=1}^{d} \lambda_k \left( \sum_{j=1}^{m_k} a_{kj} \varphi_{kj}(x_{k0}) - v_{k0} \right) \tag{8}$$

where $\lambda_k$ is a constant. Minimization of the above function $Q_\lambda$ is equivalent to solving the set of derivative equations of $Q_\lambda$ with respect to parameters $a_{kj}$ and multiplier $\lambda_k$. Then, we have

$$\begin{cases} \dfrac{\partial Q_\lambda}{\partial a_{pr}} = -2 \sum_{i=1}^{N} \left( x_{i,d} - \sum_{k=1}^{d} \sum_{j=1}^{m_k} a_{kj} \varphi_{kj}(x_{i,k}) \right) \varphi_{pr}(x_{i,p}) + \lambda_p \varphi_{pr}(x_{p0}) = 0 \\ \dfrac{\partial Q_\lambda}{\partial \lambda_p} = \sum_{j=1}^{m_p} a_{pi} \varphi_{pi}(x_{p0}) - v_{p0}, p = 1, 2, ..., d, r = 1, 2, ..., m_p \end{cases} \tag{9}$$

This leads to a system which composed of linear equations, and the unknown coefficients are parameters $a_{kj}$ and $\lambda_k$. Finally we can get the optimal parameters of the network by solving the equations.

### 3.3 The Determination of Functional Network Structure

To reduce the impact of noise on the chaotic characteristic research of the atomic clock, we carry out noise smooth process with wavelet analysis.

**Wavelet De-noise.** As we all known, wavelet decomposition can separate components of different frequency within the signal. Recently the Daubechies wavelet is widely used because of its good nature of continuous and compactly supported. It has the continuous derivative, and the smoothness quite meets the requirements. Thus we choose Daubechies wavelet for signal decomposing and noise reducing for the clock error residuals.

**Phase Space Reconstruction.** Suppose that there exists a chaotic time series $x=\{x_i \mid i=1,2,...,N\}$, Takens theorem [12] shows that we can use the delay coordinate method to reconstruct phase space of the series. And if the embedding dimension $m$ satisfies $m \geq 2d+1$ ($d$ is the dynamics dimension of the system), then the reconstructed system is equivalent to the original one in the topological sense.

In recent years, studies have shown that the main factors affect the reconstruction quality of the phase space is not only individually selection of the time delay $\tau$ and embedding dimension $m$, but together determining the time delay $\tau$ and embedding dimension $m$. It can be written as $\tau_w=(m-1)\tau$, where $\tau_w$ is the embedding window. In this paper, we choose C-C method to estimate the time delay $\tau$ and the embedding window $\tau_w$ together. This method can avoid subjectivity in calculating the embedding window.

**Chaotic Identification.** If we apply chaotic analysis in the clock error prediction, we must identify whether the clock error series is chaotic or not. Lyapunov exponent is a very important characteristic of chaotic systems, which can measure the divergence degree of the neighboring points in phase space. A positive Lyapunov exponent is a sufficient condition which can prove that the system has entered the chaotic state. Therefore, we calculate the largest Lyapunov exponent of time series with the small data sets algorithm. The small data sets algorithm has high data-using efficiency and it is reliable for small data set.

### 3.4 Model the Clock Error Residuals Based on the Functional Network

In this paper, we firstly extract the trend items with the polynomial method based on the physical property of the atomic clock, and then model the residuals with the functional network. The structure of the network is determined by wavelet de-noising and phase space reconstruction techniques.

Assuming that the result of the phase space reconstruction is $\boldsymbol{X}=\left\{\boldsymbol{X}_i \mid \boldsymbol{X}_i=\left[x_i, x_{i+\tau},...,x_{i+(m-1)\tau}\right]^{\mathrm{T}}, i=1,2,...,M\right\}$, where $\boldsymbol{X}_i$ is the point in the phase

space, $m$ is the embedding dimension and $\tau$ is the time delay. Then the number of input nodes for the functional network equals to the embedding dimensions $m$, and the number of output nodes equals to one. The network's input vector is just the point in phase space, which is $\left\{\boldsymbol{X}_i = \left[x_i, x_{i+\tau}, ..., x_{i+(m-1)\tau}\right]^{\mathrm{T}}\right\}$, and the output vector is the state point of next one step after the corresponding input vector, which is the one-dimensional sequence $\left\{\boldsymbol{Y}_i = x_{i+(m-1)\tau+1}\right\}$. The polynomial function family is selected to be the network's middle layer neuron basis function.

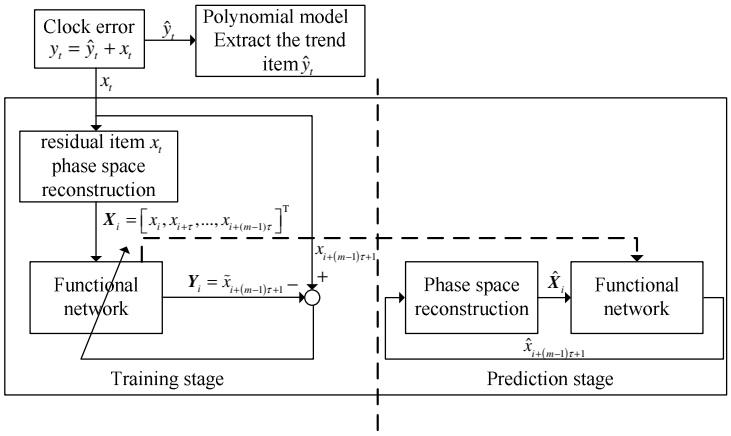



**Fig. 3.** Hybrid model based on combination of polynomial and functional network

During the prediction stage, let $x_{i+(m-1)\tau+1} = \hat{x}_{i+(m-1)\tau+1}$, so we can get a new time series of $\left\{x_{i+1}, x_{i+\tau+1}, ..., x_{i+(m-1)\tau+1}\right\}$ as the network's input vector for next moment. In this way, we can achieve the multi-step prediction for atomic clock error. Based on this modeling thoughts, we build the simulate model shown in Figure 3.

## 4 Application to Navigation Satellite Clock Error Prediction

### 4.1 Selection of Experimental Data

In order to verify the feasibility and effectiveness of the proposed model, we carried out four separate tests on the clock error prediction. In general, the IGS provides three types of GPS clock products; the ultra-rapid products (IGU) with an initial latency of 3 hours, the rapid products (IGR) with a delay of approximately 17 hours latency and the final products with a delay up to 13 days. The first 24 hours of IGU is the measured clock error and their precision is about 0.1~0.2 ns, the last 24 hours is the real-time prediction clock error (IGU-P) and the precision is 3 ns.

The IGU are very suitable to be the training data for 1-day-ahead prediction, whose precision is very high and can be available on a daily basis. Hence, we separately

carried out simulation tests upon the IGU and compared the results of the proposed model with those of the IGU-P and conventional methods.

### 4.2 Experimental Procedures

For the data pre-process, IGS ephemeris from January 7 to February 9, 2009 are chosen to be the training data, and six GPS satellites in orbit are randomly selected for four separate tests, which are 6-hour, 12-hour, 1-day, 7-day and 14-day prediction test. According to the anomalies and missing values of the atomic clock error, firstly we performed integrity check on the data, and then adopted Baarda data detection method in anomaly detection; finally we used the Lagrange interpolation to interpolate these data after the anomalies were removed.

After the data pre-process, we extracted trend item from the clock error series according to its physical property. Here the satellites of Block IIA were fitted by quadratic polynomial and satellites of Block IIR were fitted by one order polynomial. After that, we use wavelet analysis to remove the noise in series. The six GPS satellites trend items extraction are summarized in Table 1, which PRN 11 satellite's trend term extraction and wavelet de-noising is shown in Figure 4.

**Table 1.** The trend item extraction of the IGU clock error

| Type of satellite | Number of satellite | Trend item extraction |
|---|---|---|
| Block IIA | PRN03、PRN05、PRN27 | Quadratic polynomial |
| Block IIR | PRN02、PRN07、PRN11 | One order polynomial |

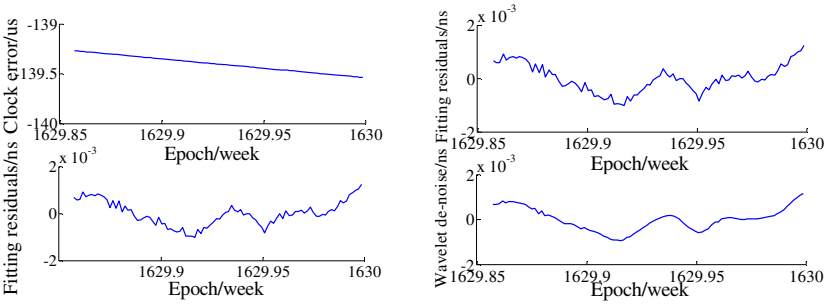



**Fig. 4.** Trend item extraction and wavelet de-noising of PRN11

After extracting the trend item, we calculated the optimal embedding dimensions $m$ and the time delay $\tau$, and then we reconstructed the phase space of the residuals. The $\Delta\overline{S}(\tau) \sim \tau$ and $S_{cor}(\tau) \sim \tau$ curves which are corresponding to PRN11 are shown in Figure 5.

The value of optimal time delay $\tau$ corresponds to the first local minimum point of $\Delta\overline{S}(\tau)$. And we can also find out the optimal embedding window width $\tau_w = (m-1)\tau$, which corresponds to the minimum point of $S_{cor}(\tau)$ in the figures. The optimal embedding dimensions $m$, time delay $\tau$ and largest Lyapunov exponent of all six satellites are summarized in Table 2.

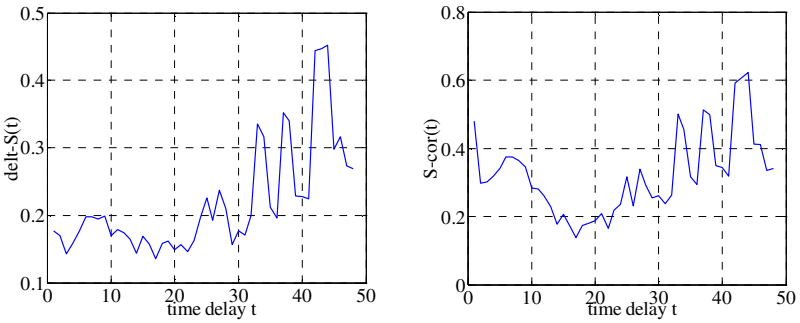



**Fig. 5.** The IGU residual's phase space reconstruction diagram of PRN11

**Table 2.** The IGU residual term's chaos characteristics

| Number of satellite | PRN02 | PRN03 | PRN05 | PRN07 | PRN11 | PRN27 |
|---|---|---|---|---|---|---|
| Embedding dimensions | 6 | 26 | 4 | 5 | 3 | 4 |
| Time delay | 6 | 3 | 20 | 13 | 10 | 8 |
| Largest Lyapunov exponent | 0.27 | 0.31 | 0.29 | 0.31 | 0.23 | 0.13 |

From Table 2, we could find that the largest Lyapunov exponents of GPS satellites are all greater than zero, which means that the clock error residuals are of chaotic characteristic. Therefore, we could determine the structure of the functional network based on the results of phase space reconstruction and model for each navigation satellite separately. Finally, the 24-hour predict curves of the six satellites are shown in Figure 6(a), and the corresponding IGU-P error curves are shown in Figure 6(b).

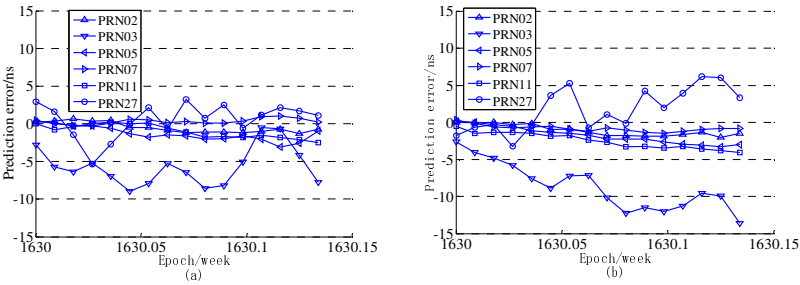



**Fig. 6.** 24 hours prediction error based on hybrid model and IGU-P

Meanwhile, we also carried out four separate tests with conventional models, which are the GM, QPM, QPPTM, ARIMA and KFM. The results of these tests are all summarized in Table 3.

### 4.3 Analysis and Discussions

It can be inferred from Figure 6 and Table 3 that the predict error curves of PRN 02, PRN 05 and PRN 07, PRN 11 convergence within 2 ns with the proposed method, but the predict error of PRN 03 and PRN 27 is relatively large. This is due to that the type

**Table 3.** Comparison among the prediction accuracy of six GPS satellites

| RMS(ns) | | Hybrid model | IGU-P | QPM | QPPTM | GM | ARIMA | KFM |
|---|---|---|---|---|---|---|---|---|
| PRN02 | 3h | 0.29 | 0.19 | 1.67 | 1.57 | 1.12 | 0.23 | 0.16 |
| | 6h | 0.38 | 0.28 | 1.67 | 1.73 | 0.79 | 0.23 | 0.26 |
| | 12h | 0.52 | 0.77 | 1.91 | 1.96 | 1.07 | 0.45 | 0.92 |
| | 24h | 0.81 | 1.33 | 3.30 | 3.28 | 2.35 | 0.52 | 1.65 |
| PRN03 | 3h | 5.50 | 4.11 | 3.83 | 1.63 | 3.57 | 13.14 | 3.35 |
| | 6h | 5.67 | 5.20 | 2.87 | 1.48 | 7.63 | 17.17 | 4.01 |
| | 12h | 6.59 | 6.79 | 2.33 | 1.34 | 12.75 | 24.12 | 3.97 |
| | 24h | 6.46 | 9.55 | 2.94 | 1.82 | 25.04 | 41.36 | 4.69 |
| PRN05 | 3h | 0.23 | 0.17 | 1.27 | 1.26 | 0.59 | 1.79 | 0.17 |
| | 6h | 0.33 | 0.49 | 1.29 | 1.29 | 0.52 | 2.17 | 0.50 |
| | 12h | 1.00 | 1.13 | 1.58 | 1.63 | 1.02 | 3.34 | 1.08 |
| | 24h | 1.64 | 2.17 | 2.86 | 2.85 | 2.13 | 7.09 | 2.14 |
| PRN07 | 3h | 0.20 | 0.29 | 0.91 | 0.83 | 1.99 | 0.30 | 0.50 |
| | 6h | 0.22 | 0.49 | 1.17 | 1.08 | 2.82 | 0.49 | 0.94 |
| | 12h | 0.33 | 0.70 | 2.03 | 2.02 | 4.60 | 1.23 | 1.58 |
| | 24h | 0.50 | 0.92 | 3.76 | 3.77 | 7.84 | 3.54 | 2.38 |
| PRN11 | 3h | 0.41 | 1.12 | 1.44 | 1.22 | 0.50 | 2.90 | 1.16 |
| | 6h | 0.31 | 1.27 | 1.33 | 1.25 | 1.36 | 5.11 | 1.53 |
| | 12h | 0.39 | 1.69 | 1.28 | 1.36 | 3.24 | 10.08 | 2.41 |
| | 24h | 1.41 | 2.79 | 2.08 | 2.05 | 6.20 | 22.39 | 4.46 |
| PRN27 | 3h | 1.10 | 1.21 | 3.85 | 1.39 | 11.84 | 4.44 | 1.94 |
| | 6h | 1.60 | 1.61 | 4.75 | 4.28 | 16.72 | 7.26 | 3.35 |
| | 12h | 2.37 | 2.74 | 6.76 | 6.37 | 27.47 | 17.66 | 6.33 |
| | 24h | 2.16 | 3.53 | 7.43 | 5.88 | 51.66 | 54.09 | 5.91 |

of the atomic clocks onboard for PRN 03 and PRN 27 is cesium atomic clock, and these two satellites are both Block IIA satellites, which are early launched. The old age and other various reasons, such as the physical property of the cesium clock itself, make the predict accuracy of the Block IIA Cesium clock generally lower than others.

In this paper, we use the IGU ephemeris as the training data; the sum of training and predict time for our proposed method is less than five minutes so as to ensure the real-time of the algorithm. If we compare Figure 6(a) and (b), it is not difficult to find out that the proposed method controls the divergent trend of the predict error in a

certain extent. It can be seen from Table 3 that the 3-hour and 6-hour predict accuracy of the clock error with the proposed method is quite equal to those of the IGU-P ephemeris which are released by IGS in real time, the 12-hour and 24-hour predict accuracy of the clock error are higher than those of the IGU-P, and the most significant improvement is 76.9% for the predict error variance.

## 5 Conclusions

This paper presents an alternative method for navigation satellite clock error prediction. The method generally considers the physical property of the atomic clock and the objective laws which are calculated from the residuals; it can avoid the human subjectivity and improve the predict accuracy and credibility. Finally, the GPS satellites are taken for example and four short-term prediction tests are done. The results are compared with those of the IGU-P and conventional methods, and it shows that the convergence and the predict accuracy are both better than others. Therefore, it can be used as a novel method in navigation satellite clock error prediction.

**Acknowledgements.** This work was supported by the National High Technology Research and Development Program of China (Grant No. 2012AA121602) and the National Science Foundation of China (Grant No. 11078001 and 11033004).

## References

1. Delporte, J.: Performance of GPS on board clocks computed by IGS. In: 18th European Frequency and Time Forum, EFTF 2004, Guildford, pp. 201–207 (2004)
2. Zhang, B., Qu, J.K., Yuan, Y.B., et al.: Fitting method for GPS satellites clock errors using wavelet and spectrum analysis. Geomatics and Information Science of Wuhan University 32(8), 715–718 (2007)
3. Cui, X.Q., Jiao, W.H.: Grey system model for the satellite clock error predicting. Geomatics and Information Science of Wuhan University 30(5), 447–450 (2005)
4. Xu, J.Y., Zeng, A.M.: Application of ARIMA(0,2,q) model to prediction of satellite clock error. Journal of Geodesy and Geodynamics 29(5), 116–120 (2009)
5. Castillo, E., Cobo, A., Gutiérrez, J.M., et al.: Functional networks with applications: a neural-based paradigm. Kluwer Academic Publishers, Boston (1999)
6. Castillo, E., Gutiérrez, J.M.: Nonlinear time series modeling and prediction using functional networks. Extracting information masked by chaos. Physics Letters A 244, 71–84 (1998)
7. Li, C.G., Liao, X.F., He, S.B., et al.: Functional network method for the identification of nonlinear system. System Engineering and Electronics 23(11), 50–53 (2001)
8. Tomasiello, S.: A functional network to predict fresh and hardened properties of self-compacting concretes. International Journal for Numerical Methods in Biomedical Engineering 27(6), 840–847 (2011)
9. Martinez, F.G., Waller, P.: GNSS clock prediction and integrity. In: The 22nd European Frequency and Time Forum. IEEE International, Besancon (2009)
10. Ke, X.-Z., Guo, L.-X.: Multi-scale fractal characteristic of atomic clock noise. Chinese Journal of Radio Science 12(4), 396–400 (1997)
11. Castillo, E.: Functional Networks. Neural Processing Letters 7, 151–159 (1998)
12. Takens, F.: Detecting strange attractors in turbulence. Lecture Notes in Mathematics, vol. 898, pp. 361–381 (1981)

# Semi-supervised K-Way Spectral Clustering with Determination of Number of Clusters

Guillaume Wacquet, Émilie Poisson-Caillault, and Pierre-Alexandre Hébert

LISIC - Lab. of Computing, Signal and Image Processing in Côte d'Opale,
Université Lille Nord de France, ULCO, Calais, France
{Guillaume.Wacquet,Emilie.Caillault,
Pierre-Alexandre.Hebert}@lisic.univ-littoral.fr
http://www-lisic.univ-littoral.fr

**Abstract.** In this paper, we propose a new K-way semi-supervised spectral clustering method able to estimate the number of clusters automatically and then to integrate some limited supervisory information. Indeed, spectral clustering can be guided thanks to the provision of prior knowledge. For the automatic determination of the number of clusters, we propose to use a criterion based on an outlier number minimization. Then, the prior knowledge consists of pairwise constraints which indicate whether a pair of objects belongs to a same cluster (*Must-Link* constraints) or not (*Cannot-Link* constraints). The spectral clustering then aims at optimizing a cost function built as a classical *Multiple Normalized Cut* measure, modified in order to penalize the non-respect of these constraints. We show the relevance of the proposed method with some UCI datasets. For experiments, a comparison with other semi-supervised clustering algorithms using pairwise constraints is proposed.

**Keywords:** Spectral embedding, Within-cluster cohesion, Semi-supervised clustering, Pairwise constraints.

## 1 Introduction

The proposed semi-supervised clustering methodology aims at clustering unknown data, considering a real case, when some expert can add some knowledge. More precisely, it is composed of two main steps: first, the build of an initial convenient clustering, without any knowledge, which is then theoritically used by an expert to assess or correct some clustering results, using pairwise constraints. In a second step, a semi-supervised clustering is then used to adjust the initial clustering, by integrating the newly available knowledge.

Among the whole set of clustering methods, we focus on algorithms able to generate discriminant representations, which can then be clustered by simple algorithms, like K-means. We look for a subspace conjointly maximizing within-cluster cohesion and between-clusters separation. Both measures can be gathered in one criterion like the Multiple Normalized Graph Cut ($MNCut$) [7]. This criterion is the basis of spectral clustering algorithms, "in vogue" in the literature, thanks to their effective global optimization and their simplicity of implementation. Both these advantages are due to the

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 317–332.
DOI: 10.1007/978-3-642-35638-4_21 

main step: the eigenvectors extraction from a similarity matrix computed on the dataset [13][4]. Similarity matrix gathers the complete information used by the method, telling for each pair of objects how close they are. Moreover, spectral clustering algorithms are able to deal with complex cases including "non-globular" or non-linearly separable clusters.

As the first step, we propose a methodology to build an optimal partition, using Ng's algorithm [5] which tends to produce particularly well discriminative representation space, to estimate the number of clusters without any kind of knowledges. This determination of number $K$ is based on a cluster representativeness, defined as the proportion of outliers. Indeed, the only $MNCut$ value cannot always guarantee a good partition because of its minimization process. This is the reason why we introduce two additional criteria: the limitation of outliers and the minimization of the number of clusters.

As the second step, the method comes within a real context, where the obtained partition is presented to experts which can validate it and can give some additional informations. Indeed, in recent years, methods incorporating prior knowledge in their clustering process have emerged as both relevant and effective in several applications, such as image segmentation [4], information retrieval or document analysis [2]. The prior knowledge is generally provided in two forms: class labels, and pairwise constraints. Labelling data is a hard and long task. Pairwise constraints simply indicate if two instances must be in the same cluster (*Must-Link*) or not (*Cannot-Link*). They are easier to collect from experts than labels [11]. In this work, pairwise constraints are randomly built from ground-truth labels. Then, we assume that the generated knowledges are true and relevant.

In this paper, we propose a new algorithm able to integrate constraints in the multiclass spectral clustering process, using a penalty term. The proposed method aims at minimizing the Multiple Normalized Cut criterion, while penalizing the non-respect of the given set of constraints. Moreover, a convenient weight, easily interpretable, is introduced in order to balance the $MNCut$ and the penalty term, i.e. the impact of the original data structure and the contribution of the constraints.

The paper is organized into four sections. The first one is theoretical and introduces some basic notations and spectral clustering methods of the literature. In a second section, we present the first step of the proposed method, i.e. a method based on a measure of the representativeness cluster allowing to estimate the number of clusters automatically. The third section presents the second step, i.e. the proposed semi-supervised K-way spectral clustering method able to integrate some prior knowledges. The last section assesses the performances of our method versus some semi-supervised algorithms of the literature on public databases extracted from UCI repository[1]. The results are finally presented, for different proportions of known constrained pairs.

## 2 Graph Embedding and Spectral Clustering

Spectral clustering is generally considered as a clustering method aiming at minimizing a *Normalized Cut* criterion between $K = 2$ clusters ($NCut$), or a *Multiple Normalized Cut* between $K \geq 2$ clusters ($MNCut$) [4][5][7]. The first measure, $NCut$, assesses

[1] http://archive.ics.uci.edu/ml/

how strongly a cluster of points (or vertices in a graph) is linked to the other points, in relation to its own cohesion. The second one deals with multiple clusters ($K \geq 2$) and is set to the average of the $NCut$ measures over the whole clusters.

### 2.1 Notations

In order to prepare the $NCut$ minimization problem formulations, some notations are first introduced, using an usual graph formalism.

- Let $\mathcal{X} = \{x_1, \ldots, x_i, \ldots, x_N\}$ be a set of $N$ objects, to be clustered;
- this set $\mathcal{X}$ is described by a weighted graph $G(V, E, W)$: $V$ is the set of nodes corresponding to the objects; $E$ is the set of edges between the nodes weighted by a matrix $W$ whose elements $w_{ij} = w_{ji} \geq 0$ tell how strongly related objects $x_i$ and $x_j$ are;
- let $D$ be the degree matrix of graph $G$, i.e. a diagonal matrix whose components are equal to the degrees of the nodes: $D_{ii} = \sum_{j=1}^{N} W_{ij}$;
- let $L$ be the unormalized Laplacian matrix of graph $G$ defined as: $L = D - W$;
- let $C = \{C_1, \ldots, C_K\}$ be a partitioning of $\mathcal{X}$ into $K$ non-empty disjoint subsets;
- each group $C_k$ is described by its volume $Vol(C_k) = \sum_{x_i \in C_k} D_{ii}$ and its "cohesion" degree $Cut(C_k, C_k) = \sum_{x_i \in C_k} \sum_{x_j \in C_k} W_{ij}$;
- the Cut between two groups is defined by $Cut(C_k, C_{k'}) = \sum_{x_i \in C_k} \sum_{x_j \in C_{k'}} W_{ij}$.

### 2.2 $MNCut$ Minimisation as Eigenproblem

In a two-class problem, the *Normalized Cut* between subsets $C_1$ and $C_2$ is defined as:

$$NCut(C_1, C_2) = Cut(C_1, C_2) \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right). \tag{1}$$

In a K-way clustering problem, $NCut$ criterion is generalized by the *Multiple Normalized Cut* ($MNCut$):

$$MNCut(C) = \sum_{k=1}^{K} \frac{Cut(C_k, C \backslash C_k)}{Vol(C_k)} = \sum_{k=1}^{K} \left( 1 - \frac{Cut(C_k, C_k)}{Vol(C_k)} \right). \tag{2}$$

Many authors of spectral clustering algorithms have shown that the minimization of $MNCut$ criterion can be achieved by solving an eigenvalue system (or generalized eigenvalue system). Their optimal clustering processing can be resumed in three steps:

1. **Preprocessing:** Computation and normalization of the similarity matrix $W$. The result is generally a normalized Laplacian matrix $\overline{L}$.
2. **Spectral Mapping:** Some $K$ vector solutions of an eigenvalue system such as $\overline{L} z_k = \lambda_k z_k$ based on the matrix issued from Step 1, are computed to form the matrix $Z = [z_1, z_2, \ldots, z_K]$. If the eigenvalues are not distinct, the eigenvectors are chosen such that $z_i^T D z_j = 0 \; for \; i \neq j$. $Z$ is then normalized into a matrix $U$, whose rows are used to map objects.
3. **Partioning:** A grouping algorithm like K-means clusters the points in the spectral space, and assigns the obtained clusters to the corresponding objects.

**Von Luxburg's Algorithm.** In [9], the author generalizes the $NCut$ criterion to the Multiple-$NCut$ ($MNCut$) criterion, and proposes to solve this problem, by considering $K$ vectors $u_k$ (denoting indicator vectors partitioning $\mathcal{X}$ in $K$ clusters), defined as: $u_k \in \{0, \frac{1}{\sqrt{Vol(C_k)}}\}^N$, and $u_{ik} = \frac{1}{\sqrt{Vol(C_k)}} \Leftrightarrow x_i \in C_k$. These indicator vectors are column-wise gathered in matrix $U$.

$$MNCut(\mathcal{C}) = \sum_{k=1}^{K} \left(1 - \frac{Cut(C_k, C_k)}{Vol(C_k)}\right) = \sum_{k=1}^{K} \left(1 - \frac{u_k^t W u_k}{u_k^t D u_k}\right) \tag{3}$$

$$= tr\left(I - (U^t DU)^{-1}(U^t WU)\right) \tag{4}$$

$$= tr\left((U^t DU)^{-1}(U^t (D - W)U)\right) \tag{5}$$

$$= tr\Big((U^t DU)^{-1}\ (U^t D^{\frac{1}{2}}(I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) D^{\frac{1}{2}} U)\Big),\ \textbf{s.t.}\ U^t DU = I \tag{6}$$

$$= tr\left(\left((D^{\frac{1}{2}}U)^t (D^{\frac{1}{2}}U)\right)^{-1} (D^{\frac{1}{2}}U)^t (I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}})(D^{\frac{1}{2}}U)\right) \tag{7}$$

$$= tr\left((Z^t Z)^{-1}(Z^t \overline{L} Z)\right),\ \textbf{s.t.}\ Z^t Z = I. \tag{8}$$

It is then possible to express the problem as:

$$\min_Z MNCut(C) = \min_Z \sum_{k=1}^{K} {z_k}^T \overline{L} z_k,\ \textbf{s.t.}\ {z_k}^T z_k = 1, \tag{9}$$

which can be optimized by solving the following eigensystem:

$$\overline{L} Z = \lambda Z, \tag{10}$$

with $\overline{L} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ the normalized Laplacian matrix and an additional formal condition $U = D^{-\frac{1}{2}} Z$: $U^T DU = I$.

Consequently, the first $K$ eigenvectors of $\overline{L}$ (i.e. with the $K$ smallest eigenvalues) minimize the criterion and allow to estimate the $K$ cluster indicator vectors. In order to retrieve discrete cluster indicator values, the eigenvector extraction is followed by a K-means step on the row of $U = D^{-\frac{1}{2}} Z$.

**Ng et al.'s Algorithm.** The authors [5] proposed an other algorithm based on Weiss [13] and Meila and Shi [4] that also solved the spectral problem (Eq. 9), but without formulating any optimization problem in terms of indicator vectors. They proposed to modify the initial similarity matrix: $w_{ii} = 0$, and to use the $K$ highest eigenvectors $z_k$ of $L_{Ng} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, orthogonal to each others, to map data. Let's remark that these eigenvectors are the $K$ lowest eigenvectors of $I - L_{Ng} = \overline{L}$.

Then, instead of computing a matrix $U = D^{-\frac{1}{2}} Z$ from matrix $Z$ stacking the extracted eigenvectors, they rather project data points in the spectral space on the unit-sphere, by normalizing $Z$ into $U$: $U_{ij} = Z_{ij} / \sqrt{\sum_j Z_{ij}^2}$. Step 3 is K-means too, initialized by points at most orthogonal.

As shown in [10], despite the diversity of the formalisms used to define the indicator vectors, all authors finally solve the same objective function (eq. 9), which involves the same normalized Laplacian matrix $\overline{L}$. However, the final solutions are different because of the chosen normalization step.

## 3 Automatic Estimation of $K$ Based on Outlier Number Minimization

Few authors focused on the automatic estimation of the number of clusters. Moreover, most algorithms have weaknesses. Indeed, some methods require cleaned and structured data in order to obtain a significant and relevant number of clusters $K$ [5] [8]. But the main weakness of most algorithms lies in the fact that they are time-consuming and very complex because of their optimization process [15] [6] [14]. In this section, we propose a new automatic estimation algorithm based on the outlier number minimization.

### 3.1 Representativeness Measure of Cluster

For the proposed methodology, we use the Ng et al.'s spectral clustering algorithm as clustering method [5]. Indeed, as shown in [10], the final projection on the unit-sphere generally gives better clustering performances. The proposed estimation method consists in *a posteriori* constraining the total proportion of outliers (i.e. objects not enough linked to their own cluster). The main idea is then to maximize the representativeness of clusters while minimizing the number of clusters. Indeed, in a real case, it is better to obtain a low value of $K$ in order to make easier an interactive learning (where an expert must analyze each cluster for labelling).

The representativeness measure of cluster $C_k$ of partition $C^K$ is based on the within-cluster averaged similarity measure, per object. More precisely, it is the average proportion of objects whose within-cluster similarity is not far lower than the within-cluster similarity mean. For a given cluster $C_k$, the criterion can so be defined as:

$$PO(C_k) = \frac{\#i \in C_k, \text{ such as } \widetilde{w_i} \leq \mu_k - \alpha.\sigma_k}{|C_k|}. \tag{11}$$

where $\alpha$ is a weighting parameter, $\widetilde{w_i}$ $(i = 1, \ldots, |C_k|)$ is the mean similarity between objects $x_i$ and all objects belonging to the same cluster. The first term $\mu_k$ represents the mean similarity of the cluster $C_k$ and the second term $\sigma_k$ is the standard deviation of the mean similarities. Then, all objects having a mean similarity $\widetilde{w_i}$ lower than $\mu_k - \alpha.\sigma_k$ are considered as outliers. $\alpha$ corresponds to the weight usually used in the outlier selection rule in some boxplot representations (greater than 1).

### 3.2 The Proposed Estimation Algorithm

To build a global representativeness measure of the partition, $PO(C^K)$, the previous measure is averaged over the whole set of clusters $C_k \in C^K$:

$$PO(C^K) = \frac{1}{K} \sum_{j=1}^{K} PO(C_k). \tag{12}$$

We propose to restrict the optimal number of clusters in the range: $[2; K_m]$ with $K_m$ the maximal value of $K$. The goal is to select $K \in \mathbb{N}$ such as:

$$K = arg\min_k\{PO(C^k) < \beta\} \text{ else } K = arg\min_k\{PO(C^k)\}$$
$$\textbf{s.t. } 2 \leq K \leq K_m. \tag{13}$$

As shown in Equation 13, the outlier number minimization criterion depends on a user threshold $\beta$ defined according to the kind of application (the smaller $\beta$ is, the more compact clusters will be). Then, we set the optimal $K$ as the smallest value which gives a proportion of outliers lower than an user threshold. The final automatic method is presented in Algorithm 1.

**Algorithm 1.** Automatic estimation algorithm

*Inputs: similarity matrix $W$, maximal value $K_m$, weight $\alpha$, threshold $\beta$*

1. Set $w_{ii} = 0$.
2. Compute the degree matrix $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$.
3. Compute the normalized Laplacian matrix $\overline{L}$: $\overline{L} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
4. Extract the largest eigenvector of $\overline{L}$.
5. FOR $k = 2 : K_m$
   - Extract the $k^{th}$ largest eigenvector of $\overline{L}$.
   - Compute the matrix $Z$ by stocking the $k$ largest eigenvectors in columns.
   - Normalize rows of $Z$ to have unit-length.
   - Compute partition $C^k$ by applying K-means algorithm with $k$ on the $k$ largest eigenvectors.
   - Compute the proportion of outliers $PO$ in partition $C^k$.
   - IF $PO < \beta$:
     - Set $K_f = k$ and $C^f = C^k$.
     - Stop loop FOR.
   - IF $k = K_m$:
     - $K_f = arg\min_k PO(C^k)$ and $C^f$ is the corresponding partition.

*Output: final value $K_f$, final partition $C^f$*

## 4 Automatic Semi-supervised K-Way Spectral Clustering Algorithm

We now focus on additional knowledge, formalized as pairwise constraints. The set $\mathcal{X}$ is now completed with the following two sets of pairs of objects [11]:

- pairs of objects that must belong to different clusters: $\{x_i, x_j\} \in \mathcal{CL}$, the *Cannot-Link* set of pairs (with $\{x_i, x_j\} \subseteq \mathcal{X}$);
- pairs of objects that must belong to the same cluster: $\{x_i, x_j\} \in \mathcal{ML}$, the *Must-Link* set of pairs (with $\{x_i, x_j\} \subseteq \mathcal{X}$).

In this section, we propose an automatic semi-supervised spectral clustering algorithm using pairwise constraints (denoted SSSC). Here, the objective function consists in the combination of the criterion of classical spectral clustering ($MNCut$) and a criterion based on the constraints.

### 4.1 Weighting of the Contribution of Constraints

In the literature, the Multiple Normalized Cut criterion can be expressed:

- from the clusters indicator vector $f_k$ such as $f_k \in \{a, b\}$ (where $\{a, b\}$ can take the values $\{0, 1\}$ or $\{-1, +1\}$);
- from the eigenvectors $z$ of the normalized Laplacian matrix $\overline{L} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.

Our work focus on the second alternative. Moreover, most of the spectral clustering methods, post-transform these vectors, either by a $D^{-\frac{1}{2}}$ pre-multiplication, or by a projection on the unit-sphere. We consider here this last choice, as in different previously presented methods [5][3].

Thanks to this final projection, we decide to make the penalty cost depend on the angles between spectral projections given by the $K$ eigenvectors. Penalty term PC is defined by dot products between constrained points, considering that this measure suits well to the alteration of angles:

$$J_{PC} = -\frac{1}{|\mathcal{CL}|} \sum_{\{x_i, x_j\} \in \mathcal{CL}} \sum_{k=1}^{K} z_{ik}.z_{jk} + \frac{1}{|\mathcal{ML}|} \sum_{\{x_i, x_j\} \in \mathcal{ML}} \sum_{k=1}^{K} z_{ik}.z_{jk}. \tag{14}$$

This criterion depends on the sets of constraints $\mathcal{ML}$ and $\mathcal{CL}$. Then, we express $J_{PC}$ as a matrix product, by using a weighting matrix $Q$, defined as:

$$Q_{ij} = Q_{ji} = \begin{cases} -\frac{1}{|\mathcal{CL}|} & \text{if } \{x_i, x_j\} \in \mathcal{CL}, \\ +\frac{1}{|\mathcal{ML}|} & \text{if } \{x_i, x_j\} \in \mathcal{ML}, \\ 0 & \text{else.} \end{cases} \tag{15}$$

The optimization criterion of the contribution of constraints $J_{PC}$ can be written as:

$$J_{PC} = \frac{1}{2} \sum_{i,j} \sum_{k=1}^{K} z_{ik} z_{jk} Q_{ij} = \sum_{k=1}^{K} z_k^T Q z_k. \tag{16}$$

### 4.2 Constrained Multiple Normalized Cut

The criterion $J_{PC}$ is combined with the multiple normalized cut criterion $MNCut$ in order to define a spectral optimization problem using pairwise constraints. The global objective function is then defined as:

$$J = MNCut - J_{PC}. \tag{17}$$

The minimization of this objective function allows to obtain a spectral projection reflecting both the original data structure and the proposed constraints. The constrained optimization problem can be written as:

$$\min_Z J(G,Z) = \min_Z \sum_{k=1}^{K} z_k^T \overline{L} z_k - z_k^T Q z_k = \min_Z \sum_{k=1}^{K} z_k^T (\overline{L} - Q) z_k, \text{ s.t. } {z_k}^T z_k = 1. \tag{18}$$

The optimization of the global objective function $J$ consists in the extraction of eigenvectors of the matrix $(\overline{L} - Q)$. This problem is clearly related to the classical spectral clustering's one, but with a normalized Laplacian matrix $\overline{L}$ penalized by a matrix $Q$, built from the $\mathcal{ML}$ and $\mathcal{CL}$ sets.

### 4.3 Setting the Balance between Contributions of Normalized Cut and Constraints

We propose to introduce a parameter $\gamma$ in order to weight the impact of constraints on the original data structure. In addition, we propose a normalization making $J$ easier to interpret. The $MNCut$ expression $z_k^T \overline{L} z_k$ belonging to $[0,1]$ (because $\overline{L}$ is supposed positive semidefinite) and the penalty one $z_k^T Q z_k$ belonging to $[\lambda_{Qmin}, \lambda_{Qmax}]$, we propose to normalize matrix $Q$ using its minimal and maximal eigenvalues $\lambda_{Qmin}$ and $\lambda_{Qmax}$:

$$\overline{Q} = \frac{Q - \lambda_{Qmin}}{\lambda_{Qmax} - \lambda_{Qmin}}. \tag{19}$$

Thanks to balancing term $\gamma$, criterion $J$ now belongs to $[0,1]$, and the final problem is set as:

$$\min_Z J(G,Z) = \min_Z \sum_{k=1}^{K} ((1-\gamma).{z_k}^T \overline{L} z_k - \gamma.{z_k}^T \overline{Q} z_k), \quad \text{s.t. } {z_k}^T z_k = 1. \tag{20}$$

The optimization of the global objective function $J$ can come down to the resolution of a standard eigenproblem:

$$((1-\gamma).\overline{L} - \gamma.\overline{Q})z = \lambda z, \tag{21}$$

i.e. the extraction of eigenvectors of the matrix $(1-\gamma).\overline{L} - \gamma.\overline{Q}$. The final method is presented in Algorithm 2.

### 4.4 Retained Solution

The vectors $z$ obtained from the resolution of the standard eigensystem (21), are projected on the unit-sphere ($U_{ij} = \frac{Z_{ij}}{(\sum_j {Z_{ij}}^2)^{\frac{1}{2}}}$). Then the retained solution is the second

smallest eigenvector in the case $K = 2$. Indeed, the first vector $(u_1)$ is constant and represents a trivial solution. The final partition is then obtained by partitioning the data thanks to the sign of values in $u_2$.

In case $K > 2$, the usage of $K$ eigenvectors is maintained as generally, considering that the constant vector $u_1$ has no impact on the spectral subspace building. These $K$ first eigenvectors are then used in order to cluster the data thanks to K-means algorithm. The algorithm in its K-way variant is resumed below (cf. Algorithm 2).

**Algorithm 2.** Automatic Semi-Supervised K-way Spectral Clustering

*Inputs: similarity matrix $W$, maximal value $K_m$, weight $\alpha$, threshold $\beta$, weight $\gamma$*

**Spectral Projection Step**

1. Apply Algorithm 1 with $K_m$, $\alpha$ and $\beta$, and set $K = K_f$.
2. Compute the constraints weighting matrix $Q$:

$$Q_{ij} = \begin{cases} -\frac{1}{|\mathcal{CL}|} & \text{if } \{x_i, x_j\} \in \mathcal{CL}, \\ +\frac{1}{|\mathcal{ML}|} & \text{if } \{x_i, x_j\} \in \mathcal{ML}, \\ 0 & \text{else.} \end{cases} \tag{22}$$

3. Compute the minimum and maximum eigenvalues (denoted $\lambda_{Qmin}$ and $\lambda_{Qmax}$) of $Q$.
4. Compute the constraints weighting matrix $\overline{Q}$: $\overline{Q} = \frac{Q - \lambda_{Qmin}}{\lambda_{Qmax} - \lambda_{Qmin}}$
5. Compute the degree diagonal matrix $D \in \Re^{N \times N}$: $D_{ii} = \sum_j w_{ij}$.
6. Compute the normalized Laplacian matrix: $\overline{L} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
7. Find, the $K$ lowest eigenvectors $\{z_1, \ldots, z_K\}$ of matrix:

$$(1 - \gamma)\overline{L} + \gamma\overline{Q}, \tag{23}$$

and form the matrix $Z = [z_1, \ldots, z_K] \in \Re^{N \times K}$.

8. Normalize the rows of $Z$ to be unit-lengthed (projection on the unit-sphere).

$$U_{ij} = \frac{Z_{ij}}{(\sum_j {Z_{ij}}^2)^{\frac{1}{2}}} \tag{24}$$

**Spectral clustering step**

1. 9. Apply a $K$-means clustering on the data matrix $U$.
2. 10. Cluster each point of $\mathcal{X}$ as its corresponding point in $U$ was clustered. *Output: final partition $C^K$*

### 4.5 Weighting of the "Must-Link" and "Cannot-Link" Contributions

The proposed algorithm is able to integrate weights on the constraints sets. These parameters allow to refine the weights of "Cannot-Link" constraints in relation to "Must-Link" constraints, and vice-versa. The criterion based on the pairwise constraints can be written as:

$$J_{PC} = -\frac{\Psi_{CL}}{|\mathcal{CL}|} \sum_{\{x_i, x_j\} \in \mathcal{CL}} \sum_{k=1}^{K} z_{ik}.z_{jk} + \frac{\Psi_{ML}}{|\mathcal{ML}|} \sum_{\{x_i, x_j\} \in \mathcal{ML}} \sum_{k=1}^{K} z_{ik}.z_{jk}. \tag{25}$$

The weights $\Psi_{CL}$ and $\Psi_{ML}$ can be used in order to balance the contributions between "Must-Link" and "Cannot-Link" pairwise constraints. In [16], the authors integrate similar weighting coefficients in their constrained principal components analysis method. The expression of the $J_{PC}$ criterion as a matrix product, is then realized by defining a weighting matrix $Q$:

$$Q_{ij} = Q_{ji} = \begin{cases} -\frac{\Psi_{CL}}{|\mathcal{CL}|} & \text{if } \{x_i, x_j\} \in \mathcal{CL}, \\ +\frac{\Psi_{ML}}{|\mathcal{ML}|} & \text{if } \{x_i, x_j\} \in \mathcal{ML}. \\ 0 & \text{else.} \end{cases} \tag{26}$$

The optimization of the global objective function $J$ is then similar to the previous case with $\Psi_{CL} = \Psi_{ML} = 1$, i.e. it is necessary to extract the $K$ first eigenvectors of the matrix $(1-\gamma).\overline{L} - \gamma.\overline{Q}$.

### 4.6 Comparison with the Semi-supervised Methods of Literature

In this section, we compare the proposed semi-supervised algorithm with similar methods from the literature, in order to highlight the contributions of our method. We focus on semi-supervised clustering methods dealing with pairwise constraints. We consider too some linear pairwise constrained methods, whose discriminant projections can be conveniently followed by a K-means step.

**Linear Constrained Methods.** Two main projection constrained methods are used in the literature: the first one based on a constrained principal component analysis representation ("Semi-Supervised Dimensionality Reduction", denoted SSDR [16]) and the second one based on a constrained projection which preserves the local neighborhood ("Constrained Locality Preserving Projection", denoted CLPP [1]).

SSDR sets similar weights for unconstrained objects. CLPP method gives an interesting data visualization tool for globular clusters of objects but this method does not allow to weight differently the contribution of unconstrained object, objects in "Must-Link" and objects in "Cannot-Link" (+1 and -1 respectively).

In addition to obtain a non linear data representation, the proposed algorithm offers the advantage to integrate weights for each objects, thanks to the similarity matrix, and to weight differently the contribution of "Must-Link" and "Cannot-Link" constraints.

**Non Linear Constrained Methods.** Two kinds of methods deal with non linear constrained data: one based on a direct modification of similarity values and one based on a global optimization of a criterion including the satisfaction of pairwise constraints.

Kamvar et al. proposed a method in the first category ("Spectral Learning" denoted SL [3]) which sets similarity values to 1 for objects in "Must-Link" and to 0 for objects in "Cannot-Link". Consequently, this approach does not allow to take into account the original local data structure.

For the second category, Wang and Davidson aim at minimizing $MNCut$ criterion subject to the satisfaction of constraints, thanks to a Lagrangian formulation. Then, the authors proposed a "Flexible Constrained Spectral Clustering" algorithm

(denoted FCSC [12]), which integrates a weighting parameter for constraints. This optimization can lead to no solutions for some values of weighting parameter, in a multi-class problem. The relevance of our method consists in always obtaining a solution for all values of the weighting parameter.

These different comparisons allows to show the contributions of the proposed semi-supervised method, and the relevance of the optimization technique used.

## 5 Experimental Results

In this section, our 2-steps methodology is applied on public benchmarks belonging to UCI repository. For each dataset, some pairwise constraints are generated from the known labels, and results obtained from Algorithm 2 are analyzed using objective evaluation measures like $MNCut$, satisfied constraints rates, or Rand Index. These results are then compared with outputs of a set of similar methods.

### 5.1 Algorithms for Comparison

For all experiments, the proposed embedding algorithm is compared with the following seven clustering methods:

- SC: the *Spectral Clustering* Ng's algorithm [5] (cf. 2.2), as a reference unsupervised method, in order to assess the impact of the added pairwise constraints;
- SSDR: the semi-supervised dimensionality reduction method [16];
- CLPP: the constrained locality preserving projection algorithm [1];
- SL: the semi-supervised *Spectral Learning* algorithm [3];
- FCSC: the original *Flexible Constrained Spectral Clustering* method [12], weighted by the value $\theta$ obtained from the rule given by the authors: $\theta = \lambda_{\max} \times Vol(G) \times \left(0.5 + 0.4 \times \frac{\#\,\text{Constraints}}{N^2}\right)$;
- FCSC-$\theta$: a variant of FCSC, where the weight $\theta$ is chosen *a posteriori* in the range $(\lambda_{\min} Vol(G), \lambda_{\max} Vol(G))$ introduced by the authors, using an exhaustive search;
- FCSC-$\theta$SP: a variant of FCSC-$\theta$, which consists in incorporating the projection on the unit-sphere step.

In order to facilitate the comparison of the methods, we decide to apply the K-means algorithm as partitioning step in the obtained subspaces from all methods. Moreover, some homogenisations were done in order to not promote our SSSC method. Then, except for methods FCSC and FCSC-$\theta$, the projection step on the unit-sphere is applied. We showed in [10] that this step allows to improve the obtained results. Moreover, in all FCSC variants except the original one, the weighting matrix used for experiments is the one defined in Algorithm 2. The weights of each kind of constraints are then similar and depend on the number of contraints defined.

For SSSC and FCSC variants (except the original), the weight of the penalty term $\theta$ or $\gamma$ is *a posteriori* optimized, by discretizing its definition interval into 100 equidistant values, and choosing the one which maximizes the criterion:

$$E = (1 - MNCut) + \mathcal{ML}_{satisfied} + \mathcal{CL}_{satisfied}, \tag{27}$$

where $\mathcal{ML}_{satisfied}$ and $\mathcal{CL}_{satisfied}$ are the respective rates of satisfied $\mathcal{ML}$ and $\mathcal{CL}$ constraints.

### 5.2 Application to UCI Datasets

In this section, our automatic semi-supervised K-way spectral clustering method is applied to some datasets well-known in the classification world (UCI datasets). For the proposed automatic estimation method, and for all experiments, we decide to set the weighting parameter $\alpha = 2.5$ (as some boxplot representations) and the threshold $\beta = 1\%$. Moreover, we propose to search the value of the number of clusters in the range $[2; 20]$, with $K_m = 20$.

For each example, some given proportions of objets are first randomly selected, so as to build sets of labelled objects. Then, they are used to deduce both $\mathcal{CL}$ and $\mathcal{ML}$ constraints sets. For each percentage tested, the previous sets of constraints are enlarged with new informations. The quality of the obtained clusterings is measured by Rand index, which reflects the similarity between the complete known partition (ground truth) and the one obtained, depending on the number of pairs of points similarly classified in the two partitions [11]. The performance scores are averaged over 10 repetitions of the constraints generation process.

Table 1(a) presents the six datasets used. We chose these databases because the distributions of classes are not uniform. For example, the "Ecoli" dataset contains 8 classes: the first one ("Cytoplasm") is composed of 143 objects, and the last one ("Inner membrane, cleavable signal sequence") is only composed of 2 objects. For each dataset, the similarity matrix is built using a Gaussian kernel: $w_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$ where $\sigma$ is the scale parameter equal to the mean of the variances of features.

**Table 1.** (a) Datasets used for experiments; (b) Performance scores on datasets, with $\beta = 1\%$

| **Dataset** | **Nb. Objects** | **Nb. Features** | **True** $K$ |
|---|---|---|---|
| Hepatitis | 80 | 19 | 2 |
| Ionosphere | 351 | 34 | 2 |
| Dermatology | 366 | 34 | 6 |
| Glass | 214 | 9 | 6 |
| Ecoli | 336 | 7 | 8 |
| Multiple Features | 2000 | 649 | 10 |

| $MNCut$ | **PO** | **Estimated** $K$ | **RI** |
|---|---|---|---|
| 0.09 | 0 | 2 | 0.61 |
| 0.10 | 0.28 | 2 | 0.51 |
| 0.14 | 0.55 | 4 | 0.65 |
| 0.24 | 0.47 | 7 | 0.74 |
| 0.28 | 0.89 | 7 | 0.81 |
| 0.22 | 0.98 | 12 | 0.92 |

The results presented in Table 1(b) show that the proposed estimation method seems consistent with the ground-truth partition and is able to estimate a satisfying value of $K$ with a low $MNCut$ values. This experiment shows that the proposed method succeeds in conjointly optimizing both proportion of outliers (lower than 1%) and number of clusters, in a efficient way. The obtained partition is then consistent and can be presented to experts in order to collect pairwise constraints.

Figure 1 shows the performance measures of all the methods applied on these UCI datasets, in terms of Rand index, i.e. the rate of pairwise relations equal to the real ones, with our estimated $K$ ($K = 7$). As it can be observed:

– Globally, methods like SSSC and some FCSC variants achieve to significantly improves the basic spectral clustering (corresponding to abscissa 0). Increasing the number of constraints globally improves the performances, and this increase is faster

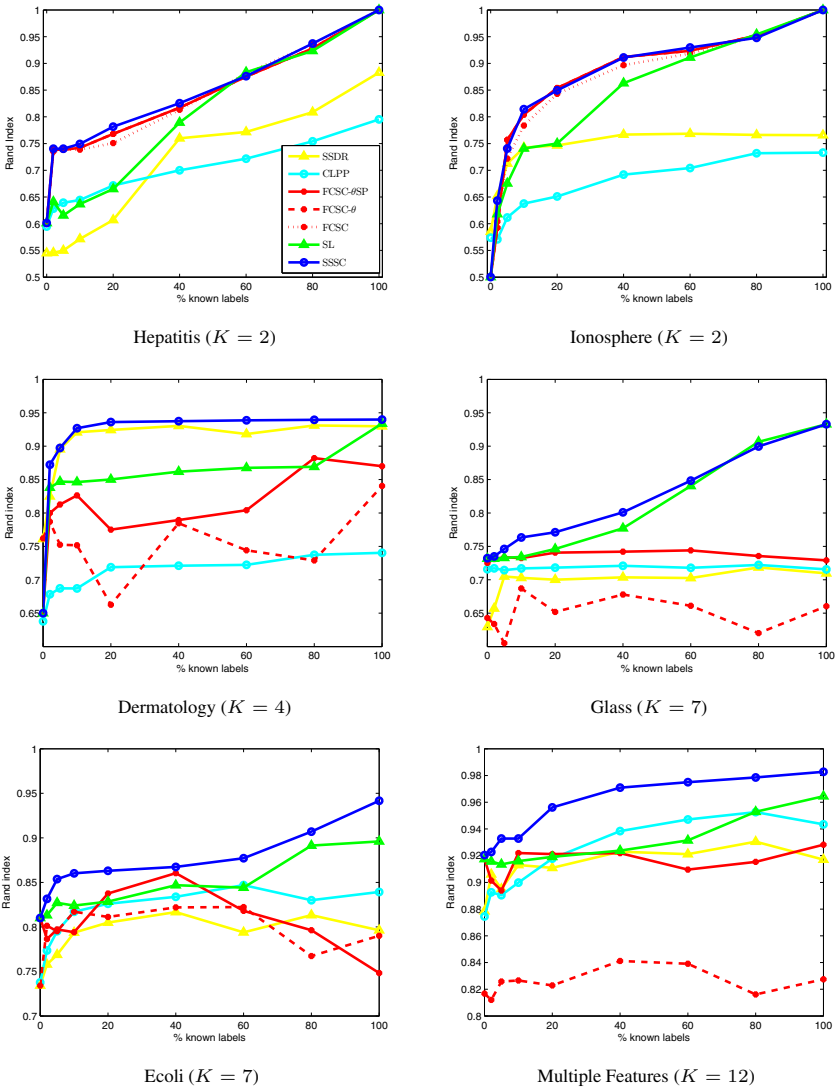



**Fig. 1.** Rand Index according to the percentage of known labels, on UCI datasets

between abscissa 0% and 5%. This means that best methods are able to improve the clustering with small amongs of pairwise constraints.

– With $K = 2$, the best results are obtained from methods SSSC and all FCSC variants: their Rand indexes are the highest. Indeed, they do not decrease with the number of constraints added. SL shows quite lower performances. SL becomes interesting, only with high numbers of constraints: weigths 0 and 1 seem too low (in absolute value) to impact the clustering.
– With $K > 2$, SSSC gets better performances than all others methods. SL gives second best results. Then the methods FCSC-$\theta$ gives very low Rand indexes: both

**Table 2.** Evaluation measures on "Ecoli" dataset (with an estimated $K = 7$) with different numbers of constraints

| % known labels | Methods | % $ML$ | % $CL$ | % Total | $MNCut$ | Rand Index |
|---|---|---|---|---|---|---|
| 0 | SSDR | / | / | / | 0.52 | 0.73 |
| | CLPP | / | / | / | 0.54 | 0.74 |
| | SL | / | / | / | **0.28** | **0.81** |
| | FCSC | / | / | / | 0.36 | 0.73 |
| | FCSC-$\theta$ | / | / | / | 0.36 | 0.73 |
| | FCSC-$\theta$SP | / | / | / | **0.28** | **0.81** |
| | **SSSC** | / | / | / | **0.28** | **0.81** |
| 5 | SSDR | 52.7 | 85.8 | 69.3 | 0.57 | 0.77 |
| | CLPP | 60.3 | 86.6 | 73.5 | 0.54 | 0.80 |
| | SL | 66.3 | 95.0 | 80.7 | **0.36** | 0.83 |
| | FCSC | / | / | / | / | / |
| | FCSC-$\theta$ | 71.9 | 82.9 | 77.4 | 0.55 | 0.80 |
| | FCSC-$\theta$SP | 64.3 | 81.3 | 72.8 | 0.58 | 0.80 |
| | **SSSC** | **99.0** | **97.5** | **98.3** | 0.38 | **0.85** |

weigths and projection steps are required to assure good performances. FCSC original method does not appear, because the constrained problem is not solved with the proposed $\theta$ value.

The proposed methodology with automatic estimation of $K$, proposition of a partition in order to generate pairwise constraints and integration of these constraints in the spectral clustering algorithm, seems efficient and relevant for this dataset.

### 5.3 Detailed Results on "Ecoli" Dataset

Table 2 shows some performance indicators of the different methods applied on a specific example, *Ecoli*, whose number of clusters $K$ is estimated to 7. In each category, *percentage of known labels* by *performance indicator*, the best result is printed in bold type. The proposed method thus appears to be very competitive versus the other methods tested. Indeed, for this dataset and for 5% of known labels, SSSC method reaches the highest rates of satisfied constraints (over 98%), while keeping a satisfactory $MNCut$ value (0.38) and a higher Rand index than other methods: final result for SSSC is then closer to the optimal clustering than other methods.

More precisely, for a small percentage of known labels, the total proportion of satisfied constraints ($ML$ and $CL$) for SSSC is better than for the others methods while remaining a small $MNCut$. Moreover, this value is coherent with the one obtained for the basic spectral clustering (corresponding to 0% of known labels and equal to 0.28) and is smaller than for the linear methods (SSDR and CLPP) and the three FCSC algorithms.

With 2% of known labels, SSSC is also better than the other methods and allows to satisfy all pairwise constraints. The same observation can be made with a fixed number of clusters given by the ground-truth on this dataset ($K = 8$). For this last case, the results obtained are presented in Table 3.

**Table 3.** Evaluation measures on "Ecoli" dataset (with the true $K = 8$) with different numbers of constraints

| % known labels | Methods | % $ML$ | % $CL$ | % Total | $MNCut$ | Rand Index |
|---|---|---|---|---|---|---|
| 0 | SSDR | / | / | / | 0.61 | 0.74 |
| | CLPP | / | / | / | 0.61 | 0.76 |
| | SL | / | / | / | **0.30** | **0.81** |
| | FCSC | / | / | / | 0.52 | 0.79 |
| | FCSC-$\theta$ | / | / | / | 0.52 | 0.79 |
| | FCSC-$\theta$SP | / | / | / | **0.30** | **0.81** |
| | **SSSC** | / | / | / | **0.30** | **0.81** |
| 5 | SSDR | 34.3 | 89.6 | 62.0 | 0.65 | 0.76 |
| | CLPP | 41.5 | 89.4 | 65.5 | 0.67 | 0.79 |
| | SL | 50.8 | 93.2 | 72.0 | **0.41** | 0.82 |
| | FCSC | / | / | / | / | / |
| | FCSC-$\theta$ | 57.5 | 86.1 | 71.8 | 0.60 | 0.79 |
| | FCSC-$\theta$SP | 87.2 | 85.7 | 86.5 | 0.65 | 0.82 |
| | **SSSC** | **99.0** | **100.0** | **99.5** | 0.49 | **0.84** |

The comparison between the results presented in Tables 2 and 3 shows that the proposed algorithm seems consistent with the ground-truth partitions. Indeed, for each proportion of known labels and for all methods tested, Rand indexes are roughly the same. However, we can note that, for 5% of known labels and with an estimated $K$, the SSDR, CLPP, SL, FCSC-$\theta$ and SSSC algorithms obtain lower $MNCut$ values and higher Rand indexes than with the true number of clusters. Then, the automatic estimation method does not negatively impact the final results obtained from the proposed semi-supervised algorithm, as shown in [10].

## 6 Conclusions

In this paper, we proposed an efficient K-way spectral algorithm able to determinate the number of clusters and using "Cannot-Link" and "Must-Link" pairwise constraints as semi-supervised information. The proposed criterion allowing to determinate the optimal value $K$ consists in a measure of representativeness clusters. This last one is defined as the averaged proportion of objects considered as outliers by a simple detection rule based upon their own within-cluster mean similarity. This rule requires the setting of an acceptable rate of outliers, depending of the kind of application.

The method is used in a real-like semi-supervised context, where the pairwise constraints could be provided by experts from an initial partition built in an unsupervised case. The estimated number of clusters and the additional knowledges are then introduced as inputs in a semi-supervised spectral clustering method. Like in its unsupervised version, the clustering problem is set as an optimization problem, consisting in minimizing an objective function proportional to the Multiple Normalized Cut measure. This measure is here balanced by a weighted penalty term assessing the non-satisfaction of the given pairwise constraints.

Some experiments and comparisons with similar methods have been carried on some UCI benchmarks. First, our automatic estimation method seems to be consistent with

the true number of clusters (given by the ground-truth partition) because of its good unsupervised and supervised performance scores. Second, in a semi-supervised case, the results illustrated that the proposed method is able to rapidly adjust the initial clustering to a more convenient one, satisfying the given constraints, even with quite low numbers of constraints. Moreover, its clustering often achieves the highest satisfied constraints rates in the two-class and multi-class cases, while keeping low $MNCut$ values.

## References

1. Cevikalp, H., Verbeek, J.: Semi-supervised dimensionality reduction using pairwise equivalence constraints. In: International Conference on Computer Vision Theory and Applications, pp. 489–496 (2008)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2006)
3. Kamvar, S., Klein, D., Manning, C.: Spectral Learning. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 561–566 (2003)
4. Meila, M., Shi, J.: Learning segmentation by random walks. In: NIPS12 Neural Information Processing Systems, pp. 873–879 (2000)
5. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS14 Neural Information Processing Systems, pp. 849–856 (2002)
6. Sanguinetti, G., Laidler, J., Lawrence, N.: Automatic determination of the number of clusters using spectral algorithms. IEEE Machine Learning for Signal Processing, 28–30 (2005)
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI Transactions on Pattern Analysis and Machine Intelligence, 888–905 (2000)
8. Shortreed, S., Meila, M.: Unsupervised spectral learning. In: Proceedings of the Twenty-First Conference Annual on Uncertainty in Artificial Intelligence, pp. 543–416 (2005)
9. Von Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing, 395–416 (2007)
10. Wacquet, G., Hébert, P.-A., Caillault, E., Hamad, D.: Semi-Supervised K-Way Spectral Clustering using Pairwise Constraints. In: NCTA, International Conference on Neural Computation Theory and Applications, pp. 72–81 (2011)
11. Wagstaff, K., Cardie, C.: Clustering with Instance-level Constraints. In: ICML International Conference on Machine Learning, pp. 1103–1110 (2002)
12. Wang, X., Davidson, I.: Flexible Constrained Spectral Clustering. In: KDD International Conference on Knowledge Discovery and Data Mining, pp. 563–572 (2010)
13. Weiss, Y.: Segmentation using eigenvectors: an unifying view. In: IEEE International Conference on Computer Vision, pp. 975–982 (1999)
14. White, S., Smyth, P.: A Spectral Clustering Approach to Finding Communities in Graphs. In: SIAM International Conference on Data Mining (2005)
15. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS Advances in Neural Information Processing Systems, pp. 1601–1608 (2004)
16. Zhang, D., Zhou, Z.-H., Chen, S.: Semi-supervised dimensionality reduction. In: SIAM 7th International Conference on Data Mining, pp. 629–634 (2007)

# Control of an Industrial PA10-7CE Redundant Robot Using a Decentralized Neural Approach

Ramon Garcia-Hernandez[1], Edgar N. Sanchez[2], Miguel A. Llama[3], and Jose A. Ruz-Hernandez[1]

[1] Facultad de Ingenieria, Universidad Autonoma del Carmen, Calle 56 No. 4, Col. Benito Juarez, C.P 24180, Cd. del Carmen, Campeche, Mexico
[2] Department of Electrical Engineering, CINVESTAV Unidad Guadalajara, Av. del Bosque 1145, C.P. 45019, Col. El Bajío, Zapopan, Jalisco, Mexico
[3] Division de Estudios de Posgrado e Investigacion, Instituto Tecnologico de la Laguna, Blvd. Revolucion y Cuauhtemoc S/N, C.P. 27000, Torreon, Coahuila, Mexico
rghernandez@pampano.unacar.mx, sanchez@gdl.cinvestav.mx, mllama@itlalaguna.edu.mx

**Abstract.** This paper presents a discrete-time decentralized control strategy for trajectory tracking of a seven degrees of freedom (DOF) redundant robot. A high order neural network (HONN) is used to approximate a decentralized control law designed by the backstepping technique as applied to a block strict feedback form (BSFF). The neural network learning is performed online using Kalman filtering. The motion of each joint is controlled independently using only local angular position and velocity measurements. The proposed controller is validated via simulations.

**Keywords:** Decentralized control, High-order neural networks, Extended Kalman filter, Backstepping, Industrial robot.

## 1 Introduction

Nowadays, Robotic arms are employed in a wide range of applications such as in manufacturing to move materials, parts, and tools of various types. Future applications will include nonmanufacturing tasks, as in construction, exploration of space, and medical care. In this context, a variety of control schemes have been proposed in order to guarantee efficient trajectory tracking and stability [1], [2]. Fast advance in computational technology offers new ways for implementing control algorithms within the approach of a centralized control design. However, there is a great challenge to obtain an efficient control for this class of systems, due to its highly nonlinear complex dynamics, the presence of strong interconnections, parameters difficult to determine, and unmodeled dynamics. Considering only the most important terms, the mathematical model obtained requires control algorithms with great number of mathematical operations, which affect the feasibility of real-time implementations.

On the other hand, within the area of control systems theory, for more than three decades, an alternative approach has been developed considering a global system as a set of interconnected subsystems, for which it is possible to design independent controllers, considering only local variables to each subsystem: the so called decentralized

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 333–346.
DOI: 10.1007/978-3-642-35638-4_22 

control [3]. Decentralized control has been applied in robotics, mainly in cooperative multiple mobile robots and robot manipulators, where it is natural to consider each mobile robot or each part of the manipulator as a subsystem of the whole system. For robot manipulators each joint and the respective link is considered as a subsystem in order to develop local controllers, which just consider local angular position and angular velocity measurements, and compensate the interconnection effects, usually assumed as disturbances. The resulting controllers are easy to implement for real-time applications [4].

In [5], a decentralized control of robot manipulators is developed, decoupling the dynamic model of the manipulator in a set of linear subsystems with uncertainties; simulation results for a robot of two joints are shown. In [6], an approach of decentralized neural identification and control for robots manipulators is presented using models in discrete-time. In [7], a decentralized control for robot manipulators is reported; it is based on the estimation of each joint dynamics, using feedforward neural networks.

In recent literature about adaptive and robust control, numerous approaches have been proposed for the design of nonlinear control systems. Among these, adaptive backstepping constitutes a major design methodology [8]. The idea behind the backstepping approach is that some appropriate functions of state variables are selected recursively as virtual control inputs for lower dimension subsystems of the overall system. Each backstepping stage results in a new virtual control design from the preceding stages; when the procedure ends, a feedback design for the true control input results, which achieves the original design objective.

In this paper, the authors propose a decentralized approach in order to design a suitable controller for each subsystem. Afterwards, each local controller is approximated by a high order neural network (HONN) [9]. The neural network (NN) training is performed on-line by means of an extended Kalman filter (EKF) [10], and the controllers are designed for each joint, using only local angular position and velocity measurements. Simulations for the proposed control scheme using a Mitsubishi PA10-7CE robot arm are presented.

## 2 Discrete-Time Decentralized Systems

Let consider a class of discrete-time nonlinear perturbed and interconnected system which can be presented in the block strict feedback form (BSFF) [8] consisting of $r$ blocks

$$\begin{aligned}
\chi_i^1(k+1) &= f_i^1\big(\chi_i^1\big) + B_i^1\big(\chi_i^1\big)\chi_i^2 + \Gamma_{i\ell}^1 \\
\chi_i^2(k+1) &= f_i^2\big(\chi_i^1, \chi_i^2\big) + B_i^2\big(\chi_i^1, \chi_i^2\big)\chi_i^3 + \Gamma_{i\ell}^2 \\
&\vdots \\
\chi_i^{r-1}(k+1) &= f_i^{r-1}\big(\chi_i^1, \chi_i^2, \ldots, \chi_i^{r-1}\big) \\
&\quad + B_i^{r-1}\big(\chi_i^1, \chi_i^2, \ldots, \chi_i^{r-1}\big)\chi_i^r + \Gamma_{i\ell}^{r-1} \\
\chi_i^r(k+1) &= f_i^r\big(\chi_i\big) + B_i^r\big(\chi_i\big)u_i + \Gamma_{i\ell}^r
\end{aligned} \tag{1}$$

where $\chi_i \in \Re^{n_i}$, $\chi_i = \left[\chi_i^{1\top}\ \chi_i^{2\top}\ \cdots\ \chi_i^{r\top}\right]^\top$ and $\chi_i^j \in \Re^{n_{ij}\times 1}$, $\chi_i^j = \left[\chi_{i1}^j\ \chi_{i2}^j\ \cdots\ \chi_{il}^j\right]^\top$, $i = 1,\ldots,N$; $j = 1,\ldots,r$; $l = 1,\ldots,n_{ij}$; $N$ is the number of subsystems, $u_i \in \Re^{m_i}$ is the input vector, the rank of $B_i^j = n_{ij}$, $\sum_{j=1}^r n_{ij} = n_i$, $\forall \chi_i^j \in D_{\chi_i^j} \subset \Re^{n_{ij}}$. We assume that $f_i^j$, $B_i^j$ and $\Gamma_i^j$ are smooth and bounded functions, $f_i^j(0) = 0$ and $B_i^j(0) = 0$. The integers $n_{i1} \le n_{i2} \le \cdots \le n_{ij} \le m_i$ define the different subsystem structures. The interconnection terms are given by

$$\begin{aligned}
\Gamma_{i\ell}^1 &= \sum_{\ell=1,\ \ell\neq i}^{N} \gamma_{i\ell}^1\left(\chi_\ell^1\right)\\
\Gamma_{i\ell}^2 &= \sum_{\ell=1,\ \ell\neq i}^{N} \gamma_{i\ell}^2\left(\chi_\ell^1, \chi_\ell^2\right)\\
&\vdots\\
\Gamma_{i\ell}^{r-1} &= \sum_{\ell=1,\ \ell\neq i}^{N} \gamma_{i\ell}^{r-1}\left(\chi_\ell^1, \chi_\ell^2, \ldots, \chi_\ell^{r-1}\right)\\
\Gamma_{i\ell}^{r} &= \sum_{\ell=1,\ \ell\neq i}^{N} \gamma_{i\ell}^{r}\left(\chi_\ell\right)
\end{aligned} \tag{2}$$

where $\chi_\ell$ represents the state vector of the $\ell$-th subsystem with $1 \le \ell \le N$ and $\ell \neq i$.

Interconnection terms (2) reflect the interaction between the $i$-th subsystem and the other ones.

## 3 High-Order Neural Networks

### 3.1 Discrete-Time HONN

Let consider the HONN described by

$$\begin{aligned}
\phi(w,z) &= w^\top S(z)\\
S(z) &= [s_1^\top(z), s_2^\top(z), \cdots, s_m^\top(z)]\\
s_i(z) &= \left[\prod_{j\in I_1}[s(z_j)]^{d_j(i_1)} \cdots \prod_{j\in I_m}[s(z_j)]^{d_j(i_m)}\right]^\top\\
i &= 1, 2, \cdots, L
\end{aligned} \tag{3}$$

where $z = [z_1, z_2, \cdots, z_p]^\top \in \Omega_z \subset \Re^p$, $p$ is a positive integer which denotes the number of external inputs, $L$ denotes the neural network node number, $\phi \in \Re^m$, $\{I_1, I_2, \cdots, I_L\}$ is a collection of not ordered subsets of $\{1, 2, \cdots, p\}$, $S(z) \in \Re^{L\times m}$, $d_j(i_j)$ is a nonnegative integer, $w \in \Re^L$ is an adjustable synaptic weight vector, and $s(z_j)$ is chosen as the hyperbolic tangent function:

$$s(z_j) = \frac{e^{z_j} - e^{-z_j}}{e^{z_j} + e^{-z_j}} \tag{4}$$

For a desired function $u^* \in \Re^m$, assume that there exists an ideal weight vector $w^* \in \Re^L$ such that the smooth function vector $u^*(z)$ can be approximated by an ideal neural network on a compact subset $\Omega_z \subset \Re^q$

$$u^*(z) = w^{*\top} S(z) + \epsilon_z \tag{5}$$

where $\epsilon_z \subset \Re^m$ is the bounded neural network approximation error vector; note that $\|\epsilon_z\|$ can be reduced by increasing the number of the adjustable weights. The ideal weight vector $w^*$ is an artificial quantity required only for analytical purposes [9], [11]. In general, it is assumed that there exists an unknown but constant weight vector $w^*$, whose estimate is $w \in \Re^L$. Hence, it is possible to define:

$$\tilde{w}(k) = w(k) - w^* \tag{6}$$

as the weight estimation error.

### 3.2 EKF Training Algorithm

It is known that Kalman filtering (KF) estimates the state of a linear system with additive state and output white noises [12]. For KF-based neural network training, the network weights become the states to be estimated. In this case, the error between the neural network output and the measured plant output can be considered as additive white noise. Due to the fact that neural network mapping is nonlinear, an EKF-type is required.

The training goal is to find the optimal weight values which minimize the prediction error. We use a EKF-based training algorithm described by:

$$\begin{aligned} K_i^j(k) &= P_i^j(k) H_i^j(k) M_i^j(k) \\ w_i^j(k+1) &= w_i^j(k) + \eta_i^j K_i^j(k) e_i^j(k) \\ P_i^j(k+1) &= P_i^j(k) - K_i^j(k) H_i^{jT}(k) P_i^j(k) + Q_i^j(k) \end{aligned} \tag{7}$$

with

$$M_i^j(k) = [R_i^j(k) + H_i^{jT}(k) P_i^j(k) H_i^j(k)]^{-1} \tag{8}$$

where $P \in \Re^{L \times L}$ is the prediction error covariance matrix, $w \in \Re^L$ is the weight (state) vector, $\eta$ is the rate learning parameter such that $0 \leq \eta \leq 1$, $L$ is the respective number of neural network weights, $x \in \Re^m$ is the measured plant state, $\hat{x} \in \Re^m$ is the neural network output, $K \in \Re^{L \times m}$ is the Kalman gain matrix, $Q \in \Re^{L \times L}$ is the state noise associated covariance matrix, $R \in \Re^{m \times m}$ is the measurement noise associated covariance matrix, and $H \in \Re^{L \times m}$ is a matrix, for which each entry $(H_{ij})$ is the derivative of one of the neural network output $(\hat{x}_i)$, with respect to one neural network weight $(w_j)$, as follows

$$H_{ij}(k) = \left[ \frac{\partial \hat{x}_i(k)}{\partial w_j(k)} \right] \tag{9}$$

where $i = 1, \ldots, m$ and $j = 1, \ldots, L$. Usually $P$ and $Q$ are initialized as diagonal matrices, with entries $P(0)$ and $Q(0)$, respectively. It is important to remark that $H(k)$, $K(k)$, and $P(k)$ for the EKF are bounded [12].

## 4 Controller Design

Once the system in the BSFF is defined, we apply the well-known backstepping technique [8]. We can define the desired virtual controls $(\alpha_i^{j*}(k), i = 1, \ldots, N; j = 1, \ldots, r-1)$ and the ideal practical control $(u^*(k))$ as follows:

$$\begin{aligned}
\alpha_i^{1*}(k) &\triangleq x_i^2(k) = \varphi_i^1(\overline{x}_i^1(k), x_{i\mathrm{d}}(k+r)) \\
\alpha_i^{2*}(k) &\triangleq x_i^3(k) = \varphi_i^2(\overline{x}_i^2(k), \alpha_i^{1*}(k)) \\
&\vdots \\
\alpha_i^{r-1*}(k) &\triangleq x_i^r(k) = \varphi_i^{r-1}(\overline{x}_i^{r-1}(k), \alpha_i^{r-2*}(k)) \\
u_i^*(k) &= \varphi_i^r(x_i(k), \alpha_i^{r-1*}(k)) \\
\chi_i(k) &= x_i^1(k)
\end{aligned} \tag{10}$$

where $\varphi_i^j(\cdot)$ with $1 \leq j \leq r$ are nonlinear smooth functions. It is obvious that the desired virtual controls $\alpha_i^*(k)$ and the ideal control $u_i^*(k)$ will drive the output $\chi_i(k)$ to track the desired signal $x_{i\mathrm{d}}(k)$. Let us approximate the virtual controls and practical control by the following HONN:

$$\begin{aligned}
\alpha_i^j(k) &= w_i^{j\top} S_i^j(z_i^j(k)) \\
u_i(k) &= w_i^{r\top} S_i^r(z_i^r(k)), \quad j = 1, \cdots, r-1
\end{aligned} \tag{11}$$

with

$$\begin{aligned}
z_i^1(k) &= [x_i^1(k), x_{i\mathrm{d}}^1(k+r)]^\top \\
z_i^j(k) &= [\overline{x}_i^j(k), \alpha_i^{j-1}(k)]^\top, \quad j = 1, \cdots, r-1 \\
z_i^r(k) &= [x_i(k), \alpha_i^{r-1}(k)]^\top
\end{aligned}$$

where $w_i^j \in \Re^{L_j}$ are the estimates of ideal constant weights $w_i^{j*}$ and $S_i^j \in \Re^{L_j \times n_j}$ with $j = 1, \ldots, r$. Define the weight estimation error as

$$\tilde{w}_i^j(k) = w_i^j(k) - w_i^{j*}. \tag{12}$$

Then, the corresponding weights updating laws are defined as

$$w_i^j(k+1) = w_i^j(k) + \eta_i^j K_i^j(k) e_i^j(k) \tag{13}$$

with

$$\begin{aligned}
K_i^j(k) &= P_i^j(k) H_i^j(k) M_i^j(k) \\
M_i^j(k) &= [R_i^j(k) + H_i^{j\top}(k) P_i^j(k) H_i^j(k)]^{-1} \\
P_i^j(k+1) &= P_i^j(k) - K_i^j(k) H_i^{j\top}(k) P_i^j(k) + Q_i^j(k)
\end{aligned} \tag{14}$$

$$H_i^j(k) = \left[\frac{\partial \hat{v}_i^j(k)}{\partial w_i^j(k)}\right] \tag{15}$$

and

$$e_i^j(k) = \upsilon_i^j(k) - \hat{\upsilon}_i^j(k) \tag{16}$$

where $\upsilon_i^j(k) \in \Re^{n_j}$ is the desired signal and $\hat{\upsilon}_i^j(k) \in \Re^{n_j}$ is the HONN function approximation defined, respectively as follows

$$\begin{aligned} \upsilon_i^1(k) &= x_{i\mathrm{d}}^1(k) \\ \upsilon_i^2(k) &= x_i^2(k) \\ &\vdots \\ \upsilon_i^r(k) &= x_i^r(k) \end{aligned} \tag{17}$$

and

$$\begin{aligned} \hat{\upsilon}_i^1(k) &= \chi_i^1(k) \\ \hat{\upsilon}_i^2(k) &= \alpha_i^1(k) \\ &\vdots \\ \hat{\upsilon}_i^r(k) &= \alpha_i^{r-1}(k) \end{aligned} \tag{18}$$

$e_i^j(k)$ denotes the error at each step as

$$\begin{aligned} e_i^1(k) &= x_{i\mathrm{d}}^1(k) - \chi_i^1(k) \\ e_i^2(k) &= x_i^2(k) - \alpha_i^1(k) \\ &\vdots \\ e_i^r(k) &= x_i^r(k) - \alpha_i^{r-1}(k). \end{aligned} \tag{19}$$

The whole proposed neural backstepping control scheme is shown in Fig. 1.

## 5 Seven DOF Mitsubishi PA10-7CE Robot Arm

### 5.1 Robot Description

The Mitsubishi PA10-7CE arm is an industrial robot manipulator which completely changes the vision of conventional industrial robots. Its name is an acronym of Portable General-Purpose Intelligent Arm. There exist two versions [13]: the PA10-6C and the PA10-7C, where the suffix digit indicates the number of degrees of freedom of the arm. This work focuses on the study of the PA10-7CE model, which is the enhanced version of the PA10-7C. The PA10 arm is an open architecture robot; it means that it possesses:

- A hierarchical structure with several control levels.
- Communication between levels, via standard interfaces.
- An open general purpose interface in the higher level.

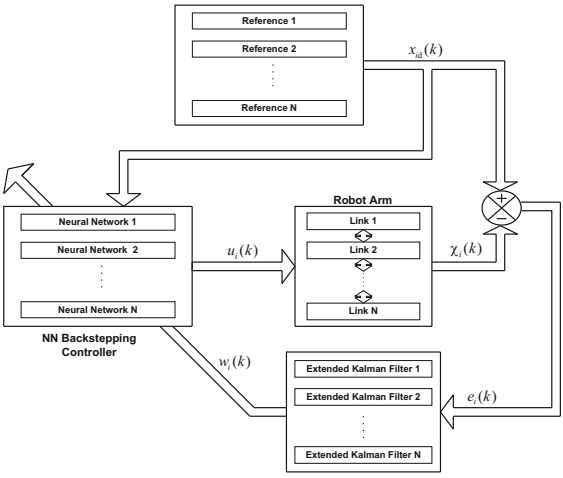



**Fig. 1.** Decentralized neural backstepping control scheme

This scheme allows the user to focus on the programming of the tasks at the PA10 system higher level, without regarding on the operation of the lower levels. The programming can be performed using a high level language, such as Visual BASIC or Visual C++, from a PC with Windows operating system. The PA10 robot is currently the open architecture robot more employed for research [14], [15].The PA10 system is composed of four sections or levels, which conform a hierarchical structure:

Level 4: Operation control section (OCS); formed by the PC and the teaching pendant.

Level 3: Motion control section (MCS); formed by the motion control and optical boards.

Level 2: Servo drives.

Level 1: Robot manipulator.

The PA10 robot is a 7-DOF redundant manipulator with revolute joints. Figure 2 shows a diagram of the PA10 arm, indicating the positive rotation direction and the respective names of each of the joints.

### 5.2 Control Objective

The decentralized discrete-time model for a seven DOF robot arm can be represented as follows

$$\begin{aligned} \chi_i^1(k+1) &= f_i^1(\chi_i^1) + B_i^1(\chi_i^1)\chi_i^2 + \Gamma_i^1 \\ \chi_i^2(k+1) &= f_i^2(\chi_i^1, \chi_i^2) + B_i^2(\chi_i^1, \chi_i^2)u_i(k) + \Gamma_i^2 \end{aligned} \tag{20}$$

where $i = 1, \ldots, 7$; $\chi_i^1(k)$ are the angular positions, $\chi_i^2(k)$ are the angular velocities, $u_i(k)$ represents the applied torque to $i$-th joint respectively. $f_i^j(\cdot)$ and $B_i^j(\cdot)$ depend only on the local variables and $\Gamma_i^j$ are the interconnection effects.

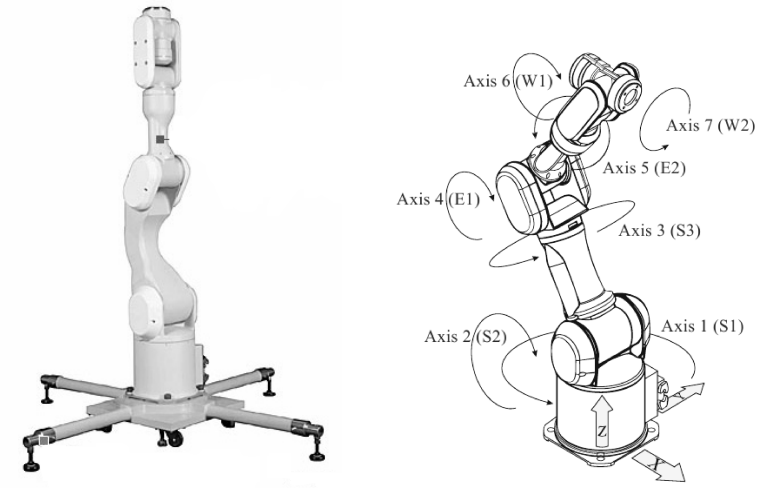



**Fig. 2.** Mitsubishi PA10-7CE robot arm

Let define the following states:

$$x^1(k) = \begin{bmatrix} \chi_1^1 & \chi_2^1 & \chi_3^1 & \chi_4^1 & \chi_5^1 & \chi_6^1 & \chi_7^1 \end{bmatrix}^\top$$

$$x^2(k) = \begin{bmatrix} \chi_1^2 & \chi_2^2 & \chi_3^2 & \chi_4^2 & \chi_5^2 & \chi_6^2 & \chi_7^2 \end{bmatrix}^\top$$

$$u(k) = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{bmatrix}^\top$$

$$x_{\mathrm{d}}^1(k) = \begin{bmatrix} x_{1\mathrm{d}}^1 & x_{2\mathrm{d}}^1 & x_{3\mathrm{d}}^1 & x_{4\mathrm{d}}^1 & x_{5\mathrm{d}}^1 & x_{6\mathrm{d}}^1 & x_{7\mathrm{d}}^1 \end{bmatrix}^\top$$

$$\chi_i^1(k) = x_i^1(k) \tag{21}$$

where $x_{1\mathrm{d}}^1(k)$ to $x_{7\mathrm{d}}^1(k)$ are the desired trajectory signals. The control objective is to drive the output $\chi_i^1(k)$ to track the reference $x_{i\mathrm{d}}^1(k)$. Using (21), the system (20) can be represented in the block strict feedback form as

$$\begin{aligned} x_i^1(k+1) &= f_i^1(x_i^1(k)) + g_i^1(x_i^1(k))x_i^2(k) \\ x_i^2(k+1) &= f_i^2(\overline{x}_i^2(k)) + g_i^2(\overline{x}_i^2(k))u_i(k) \end{aligned} \tag{22}$$

where $\overline{x}_i^2(k) = \begin{bmatrix} x_i^1(k) & x_i^2(k) \end{bmatrix}^\top$, $i = 1, \ldots, 7$, $f_i^1(x_i^1(k))$, $g_i^1(x_i^1(k))$, $f_i^2(\overline{x}_i^2(k))$ and $g_i^2(\overline{x}_i^2(k))$ are assumed to be unknown. To this end, we use a HONN to approximate the desired virtual controls and the ideal practical control described as

$$\begin{aligned} \alpha_i^{1*}(k) \triangleq x_i^2(k) &= \varphi_i^1(x_i^1(k), x_{i\mathrm{d}}^1(k+2)) \\ u_i^*(k) &= \varphi_i^1(x_i^1(k), x_i^2(k), \alpha_i^{1*}(k)) \\ \chi_i^1(k) &= x_i^1(k). \end{aligned} \tag{23}$$

The HONN proposed for this application is as follows:

$$\begin{aligned} \alpha_i^{1*}(k) &= w_i^{1\top} S_i^1(z_i^1(k)) \\ u_i(k) &= w_i^{2\top} S_i^2(z_i^2(k)) \end{aligned} \tag{24}$$

with

$$\begin{aligned} z_i^1(k) &= [x_i^1(k), x_{i\mathrm{d}}^1(k+2)] \\ z_i^2(k) &= [x_i^1(k), x_i^2(k), \alpha_i^1(k)]. \end{aligned} \tag{25}$$

The weights are updated using the EKF (13) - (19) with $i = 1, 2$ and

$$\begin{aligned} e_i^1(k) &= x_{i\mathrm{d}}^1(k) - \chi_i^1(k) \\ e_i^2(k) &= x_i^2(k) - \alpha_i^1(k). \end{aligned} \tag{26}$$

The training is performed on-line using a series-parallel configuration. All the neural network states are initialized in a random way.

## 6 Simulation Results

For simulation, we select the following discrete-time trajectories [16]

$$\begin{aligned} x_{1\mathrm{d}}^1(k) &= c_1(1 - e^{d_1 kT^3})\sin(\omega_1 kT)[\mathrm{rad}] \\ x_{2\mathrm{d}}^1(k) &= c_2(1 - e^{d_2 kT^3})\sin(\omega_2 kT)[\mathrm{rad}] \\ x_{3\mathrm{d}}^1(k) &= c_3(1 - e^{d_3 kT^3})\sin(\omega_3 kT)[\mathrm{rad}] \\ x_{4\mathrm{d}}^1(k) &= c_4(1 - e^{d_4 kT^3})\sin(\omega_4 kT)[\mathrm{rad}] \\ x_{5\mathrm{d}}^1(k) &= c_5(1 - e^{d_5 kT^3})\sin(\omega_5 kT)[\mathrm{rad}] \\ x_{6\mathrm{d}}^1(k) &= c_6(1 - e^{d_6 kT^3})\sin(\omega_6 kT)[\mathrm{rad}] \\ x_{7\mathrm{d}}^1(k) &= c_7(1 - e^{d_7 kT^3})\sin(\omega_7 kT)[\mathrm{rad}] \end{aligned} \tag{27}$$

the selected parameters $c_i$, $d_i$ and $\omega_i$ for desired trajectories of each joint are shown in Table 1. The sampling time is selected as $T = 1$ millisecond.

These selected trajectories (27) incorporate a sinusoidal term to evaluate the performance in presence of relatively fast periodic signals, for which the non-linearities of the robot dynamics are really important.

**Table 1.** Parameters for desired trajectories

| $i$-th Joint | $c_i$ | $d_i$ | $\omega_i$ |
|---|---|---|---|
| 1 | $\pi/2$ | 0.001 | 0.285 rad/s |
| 2 | $\pi/3$ | 0.001 | 0.435 rad/s |
| 3 | $\pi/2$ | 0.01 | 0.555 rad/s |
| 4 | $\pi/3$ | 0.01 | 0.645 rad/s |
| 5 | $\pi/2$ | 0.01 | 0.345 rad/s |
| 6 | $\pi/3$ | 0.01 | 0.615 rad/s |
| 7 | $\pi/2$ | 0.01 | 0.465 rad/s |

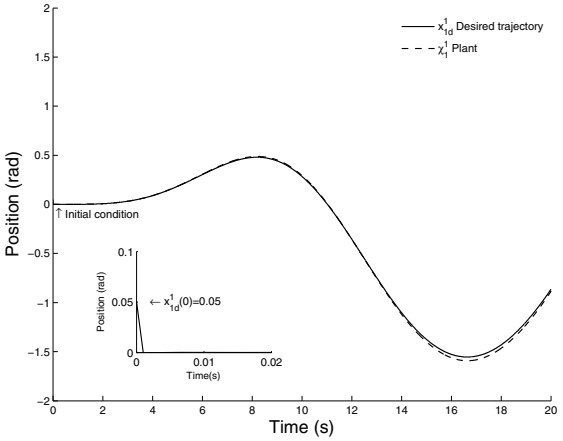



**Fig. 3.** Trajectory tracking for joint 1 $x^1_{1\mathrm{d}}(k)$ (solid line) and $\chi^1_1(k)$ (dashed line)

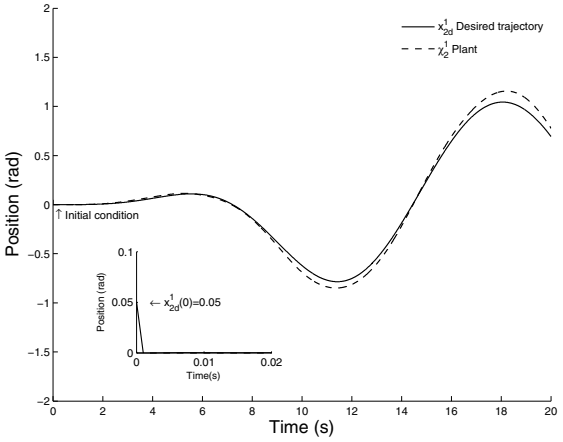



**Fig. 4.** Trajectory tracking for joint 2 $x^1_{2\mathrm{d}}(k)$ (solid line) and $\chi^1_2(k)$ (dashed line)

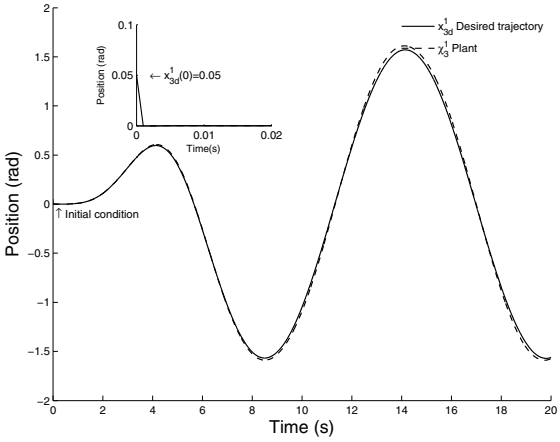



**Fig. 5.** Trajectory tracking for joint 3 $x_{3\mathrm{d}}^{1}(k)$ (solid line) and $\chi_{3}^{1}(k)$ (dashed line)

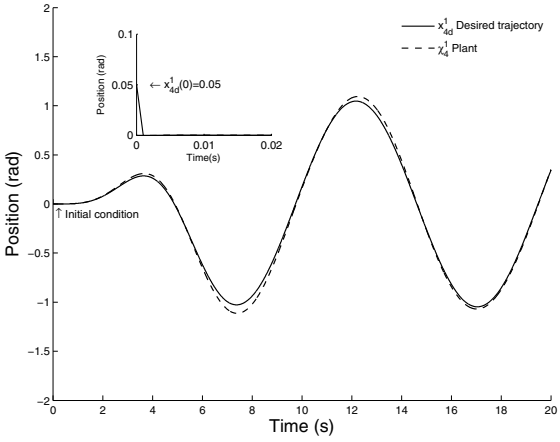



**Fig. 6.** Trajectory tracking for joint 4 $x_{4\mathrm{d}}^{1}(k)$ (solid line) and $\chi_{4}^{1}(k)$ (dashed line)

Simulation results for trajectory tracking using the decentralized neural backstepping control (DNBS) scheme are shown in Figs. 3 to Fig. 9. The initial conditions for the plant are different that those of the desired trajectory. According to these figures, the tracking errors for all joints present a good behavior and remain bounded as shown in Fig. 10.

The applied torques to each joint are always inside of the prescribed limits given by the actuators manufacturer (see Table 2); that is, their absolute values are smaller than the bounds $\tau_1^{\max}$ to $\tau_7^{\max}$, respectively.

**Table 2.** Maximum torques

| Joint | Max Torque |
|---|---|
| 1 | 232 N-m |
| 2 | 232 N-m |
| 3 | 100 N-m |
| 4 | 100 N-m |
| 5 | 14.5 N-m |
| 6 | 14.5 N-m |
| 7 | 14.5 N-m |

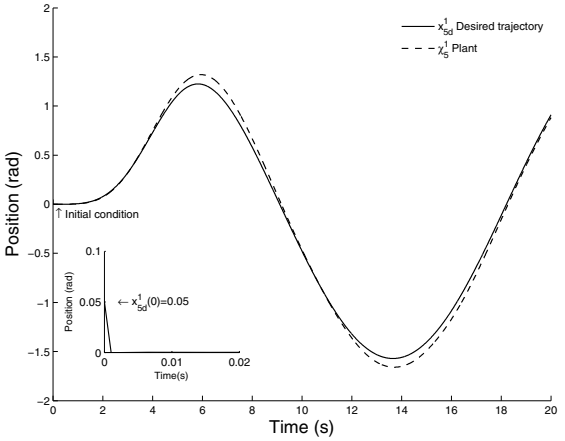



**Fig. 7.** Trajectory tracking for joint 5 $x^1_{5\mathrm{d}}(k)$ (solid line) and $\chi^1_5(k)$ (dashed line)

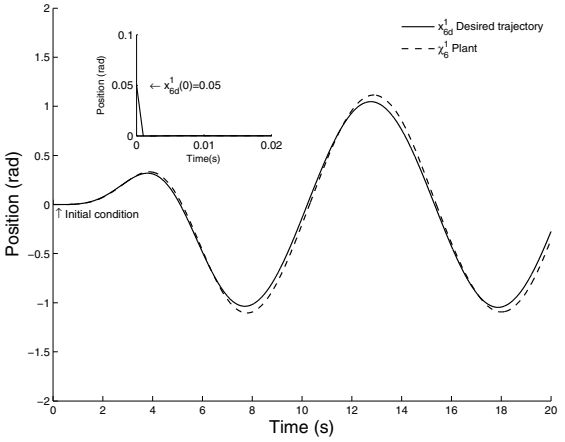



**Fig. 8.** Trajectory tracking for joint 5 $x^1_{5\mathrm{d}}(k)$ (solid line) and $\chi^1_5(k)$ (dashed line)

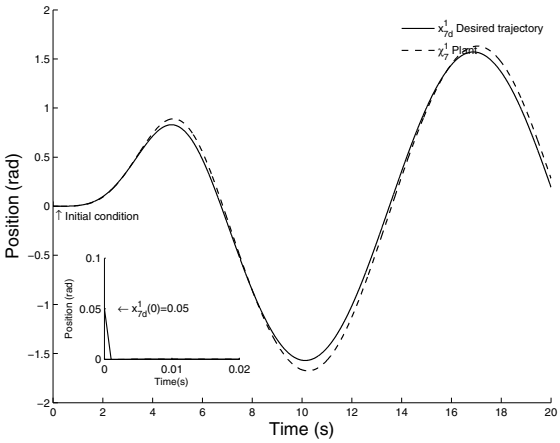



**Fig. 9.** Trajectory tracking for joint 5 $x_{5\mathrm{d}}^{1}(k)$ (solid line) and $\chi_{5}^{1}(k)$ (dashed line)

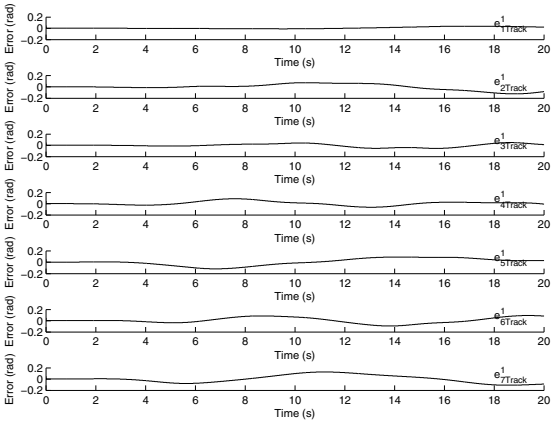



**Fig. 10.** Tracking errors for joints 1 to 7

## 7 Conclusions

In this paper a decentralized neural control scheme based on the backstepping technique is presented. The control law for each joint is approximated by a high order neural network. The training of each neural network is performed on-line using an extended Kalman filter. Simulations results for trajectory tracking using a seven DOF PA10-7CE Mitsubishi robot arm show the effectiveness of the proposed control scheme.

**Acknowledgements.** The first author thanks to Universidad Autonoma del Carmen (UNACAR) and the Programa de Mejoramiento del Profesorado (PROMEP) for supporting this research.

## References


1. Sanchez, E.N., Ricalde, L.J.: Trajectory tracking via adaptive recurrent neural control with input saturation. In: Proc. of International Joint Conference on Neural Networks, Portland, Oregon, pp. 359–364 (2003)
2. Santibañez, V., Kelly, R., Llama, M.A.: A novel global asymptotic stable set-point fuzzy controller with bounded torques for robot manipulators. IEEE Transactions on Fuzzy Systems 13(3), 362–372 (2005)
3. Huang, S., Tan, K.K., Lee, T.H.: Decentralized control design for large-scale systems with strong interconnections using neural networks. IEEE Transactions on Automatic Control 48(5), 805–810 (2003)
4. Liu, M.: Decentralized control of robot manipulators: nonlinear and adaptive approaches. IEEE Transactions on Automatic Control 44(2), 357–363 (1999)
5. Ni, M.L., Er, M.J.: Decentralized control of robot manipulators with coupling and uncertainties. In: Proc. of the American Control Conference, Chicago, Illinois, pp. 3326–3330 (2000)
6. Karakasoglu, A., Sudharsanan, S.I., Sundareshan, M.K.: Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators. IEEE Transactions on Neural Networks 4(6), 919–930 (1993)
7. Safaric, R., Rodic, J.: Decentralized neural-network sliding-mode robot controller. In: Proc. of 26th Annual Conference on the IEEE Industrial Electronics Society, Nagoya, Aichi, Japan, pp. 906–911 (2000)
8. Krstic, M., Kanellakopoulos, I., Kokotovic, P.: Nonlinear and Adaptive Control Design. John Wiley & Sons Inc., New York (1995)
9. Ge, S.S., Zhang, J., Lee, T.H.: Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time. IEEE Transactions on Systems, Man, and Cybernetics Part B 34(4), 1630–1645 (2004)
10. Alanis, A.Y., Sanchez, E.N., Loukianov, A.G.: Discrete-Time Adaptive Backstepping Nonlinear Control via High-Order Neural Networks. IEEE Transactions on Neural Networks 18(4), 1185–1195 (2007)
11. Rovithakis, G.A., Christodoulou, M.A.: Adaptive Control with Recurrent High-Order Neural Networks. Springer, London (2000)
12. Song, Y., Grizzle, J.W.: The extended Kalman filter as local asymptotic observer for discrete-time nonlinear systems. Journal of Mathematical Systems, Estimation and Control 5(1), 59–78 (1995)
13. Higuchi, M., Kawamura, T., Kaikogi, T., Murata, T., Kawaguchi, M.: Mitsubishi clean room robot. Mitsubishi Heavy Industries, Ltd., Tecnical Review (2003)
14. Jamisola, R.S., Maciejewski, A.A., Roberts, R.G.: Failure-tolerant path planning for the PA-10 robot operating amongst obstacles. In: Proceedings of IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, pp. 4995–5000 (2004)
15. Kennedy, C.W., Desai, J.P.: Force feedback using vision. In: Proceedings of IEEE International Conference on Advanced Robotics, Coímbra, Portugal (2003)
16. Ramirez, C.: Dynamic modeling and torque-mode control of the Mitsubishi PA10-7CE robot. Master Dissertation, Instituto Tecnológico de la Laguna, Torreón, Coahuila, Mexico (2008) (in Spanish)

# CMAC Structure Optimization Based on Modified Q-Learning Approach and Its Applications

Weiwei Yu[1], Kurosh Madani[2], and Christophe Sabourin[2]

[1] School of Mechatronic Engineering, Northwestern Polytechnical University, Youyi Xilu 127hao, Xi'an 710072, P.R. China
[2] Signals, Images, and Intelligent Systems Laboratory (LISSI / EA 3956), Paris Est University, Senart Institute of Technology, Avenue Pierre Point, 77127 Lieusaint, France
yuweiwei@nwpu.edu.cn, {madani,sabourin}@u-pe.fr

**Abstract.** Comparing with other neural network based models, CMAC has been applied successfully in many nonlinear control systems because of its computational speed and learning ability. However, for high-dimensional input CMAC in real world applications such as robot, the useable memory is finite or pre-allocated, thus we often have to make our choice between learning accuracy and memory size. This paper discusses how both the number of layer and step quantization influence the approximation quality of CMAC. By experimental enquiry, it is shown that it is possible to decrease the memory size without losing the approximation quality by selecting the adaptive structural parameters. Based on modified Q-learning approach, the CMAC structural parameters can be optimized automatically without increasing the complexity of its structure. The choice of this optimized CMAC structure can achieve a tradeoff between the learning accuracy and finite memory size. At last, this Q-learning based CMAC structure optimization approach is applied on the walk pattern generating for biped robot and workpiece orientation estimation for robot arm assembly respectively.

**Keywords:** CMAC neural network, Structural parameters, Q-learning; Structure optimization.

## 1 Introduction

The Cerebellar Model Articulation Controller (CMAC) is a neural network based model proposed by Albus inspiring from the studies on the human cerebellum [1]. Because of the advantages of simple and effective training properties and fast learning convergence, CMAC has been used in many real-time control systems, pattern recognition and signal processing problems successfully. However, besides its attractive features, the main drawback of CMAC network in realistic applications is related to the required memory size. For the high dimension input space greater than two, on one hand, in order to increase the accuracy of the control, the quantification step usually be chosen as small as possible which will cause the CMAC's memory size become quickly very large. On the other hand, generally in real world applications such as robot and aircraft, the useable memory is finite or pre-allocated. Therefore, we often have to make our choice between accuracy and memory size.

K. Madani et al. (Eds.): *Computational Intelligence*, SCI 465, pp. 347–359.
DOI: 10.1007/978-3-642-35638-4_23 

To solve the problem relating to the size of the memory, the efforts can be classified into three main approaches in general. The first theoretical aspect is developed on how to modify the input space quantization [2,3]. This is based on the idea of the quantization method of input space is a decisive factor of the memory utilization and the more intervals we quantized, the more precise learning we will obtain. However, not only the quantization step but also number of layers determines the learning preciseness and the required memory size. The second approach involves the use of multilayered CMACs of increasing resolutions, demonstrating the properties of generating and pruning the input layers automatically [4,5]. Nevertheless they lack the theoretical proof of the system's learning convergence, which is a desirable attribute for control and function approximation tasks. The third orientation which is most popular focused on incorporating fuzzy logic into CMAC to obtain a new fuzzy neural system model called fuzzy CMAC (FCMAC) to alleviate the required memory size [6,7]. Yet, it rises new problem on how to design an optimal fuzzy sets.

In the above CMAC literatures, there is no one related to the tradeoff problem of limited memory size and learning quality. It is traditionally thought that the more exquisitely the input space is divided, the more accurately the output results of CMAC can be obtained. However, this will certainly cause quickly increasing of memory size, if we do not develop more complex CMAC structure, since the simplicity of structure play an important role in on-line application of neural network, such as robot. In fact, by experimental study of approximation examples, in which several high-dimension functions were selected and several combinations of structural parameters were tested, we found that the learning preciseness and the required memory size are determined by both of the quantization step and number of layers. Thus, adaptive choice of these structural parameters may overcome the above primary limitation. In this way, take aim at CMAC structure be optimized automatically for a given problem, it is possible to decrease the memory size according to the desired performance of CMAC NN.

The paper is organized as follows: In section 2, CMAC model and its structure parameters are concisely overviewed. Section 3 presents the experimental study of the influence of structural parameters on the memory size and approximation quality. In section 4, a Q-learning based structure optimized approach is developed. The proposed approach is applied on the desired joint angle tracking for biped robot and workpiece location for robot arm assembly in section 5 and section 6 respectively. Conclusion and further works are finally set out.

## 2 CMAC ANN Architecture and Structural Parameters

The output $Y$ of the CMAC is computed using two mappings. The first mapping $(X \to A)$ projects the input space point $X = [x_1, x_2]$ into a binary associative vector $A = [a_1, a_2, \ldots, a_{N_c}]$. Each element of $A$ is associated with one detector. When one detector is activated, the corresponding element in $A$ of this detector is equal to 1, otherwise it equals to 0. The second mapping $(A \to Y)$ computes the output $Y$ of the network as a scalar product of the association vector $A$ and the weight vector

$W = [w_1, w_2, \ldots, w_{N_C}]$ according to the relation (3), where $(X)^T$ represents the transpose of the input vector.

$$Y = A(X)^T W \tag{1}$$

The weights of CMAC neural network are updated by using equation (4). $w(t_i)$ and $w(t_{i-1})$ are respectively the weights before and after training at each sample time $t_i$. $N_l$ is the generalization number of each CMAC and $\beta$ is a parameter included in [0 1]. $\Delta e$ is the error between the desired output $Y^d$ of the CMAC and the computed output $Y$ of the corresponding CMAC.

$$W(t_i) = W(t_{i-1}) + \frac{\beta \Delta e}{N_l} \tag{2}$$

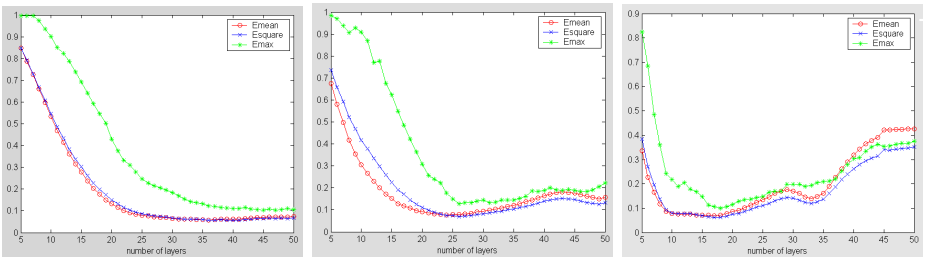
Due to its structure, CMAC is preferable be used to approximate both linear and non-linear functions. If the complexity of its structure is not increased additionally, there are essentially two structural factors ruling the approximation quality. The first one, called "quantization step" $\Delta q$, allows to map a continuous signal into a discrete signal. The second parameter called "generalization parameter" $N_l$ corresponds to the number of layers. These two parameters allow to define the total number of cells $N_C$.

## 3 Impact of Structural Parameters on CMAC ANN

We aim to show the relation between the structural parameters of CMAC NN, the quality of the approximation and the required memory size for a given function. Our study is based on an experimental enquiry, in which several high dimension functions are used in order to test the neural network's approximation abilities. In this section, take FSIN and GUASS functions as examples, simulations for three different step quantization $\Delta q$ are carried out, when the number of layers increases from 5 to 50 for FSIN function, and from 5 to 450 for two dimension GAUSS function. For each of the aforementioned functions, a training set including 100×100 random values selected in the corresponding two-dimensional space, has been constructed. Weights of CMAC are updated using equation (2). When CMAC is totally trained, three modeling errors: mean absolute error $E_{mean}$, mean squared error $E_{square}$ and maximum absolute error $E_{\max}$ are carried out. The overview of the obtained results is shown in Figure 1 and 2 respectively.

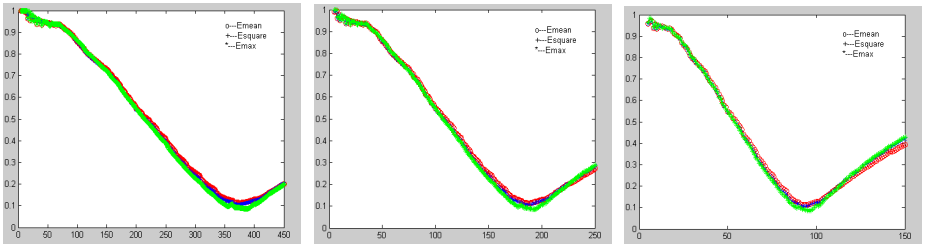
It must be noticed that the modeling error depends on the quantization step $\Delta q$ on the one hand, and the number of layers $N_l$ on the other hand. When $\Delta q$ is relatively small (for example $\Delta q = 0.0025$), errors converges toward a constant value close to the minimum error. But, when the quantization is greater, results show there is an optimal structure when the modeling errors are minimal. However, it must be noticed that for each quantization step, the minimal errors are quasi-identical but for different

number of layers. As the curve trends of the mean absolute error $E_{mean}$, mean squared error $E_{square}$ and maximum absolute error $E_{\max}$ are same, only take $E_{square}$ as an example.




$\Delta q = 0.0025$ $\Delta q = 0.0050$ $\Delta q = 0.0075$

**Fig. 1.** Approximation error according to the number of layers for FSIN function with different step quantization




$\Delta q = 0.0025$ $\Delta q = 0.0050$ $\Delta q = 0.0100$

**Fig. 2.** Approximation error according to the number of layers for GUASS function with different step quantization

**Table 1.** CMAC structure with minimum mean squared error for FSIN function

| $\Delta q$ | $N_l$ | $E_{square}$ | $S_C$ | $N_C$ |
|---|---|---|---|---|
| 0.0025 | 41 | 5.81% | 0.1225 | 4940 |
| 0.0050 | 26 | 6.93% | 0.13 | 2089 |
| 0.0075 | 18 | 6.21% | 0.135 | 1441 |

**Table 2.** CMAC structure with minimum mean squared error for GAUSS function

| $\Delta q$ | $N_l$ | $E_{square}$ | $S_C$ | $N_C$ |
|---|---|---|---|---|
| 0.0025 | 115 | 7.80% | 0.2875 | 7345 |
| 0.0050 | 58 | 8.35% | 0.29 | 3697 |
| 0.0100 | 29 | 8.11% | 0.29 | 1841 |

The mean squared error $E_{square}$ for FSIN function is equal to 5.81% and 6.21% in the case where $\Delta q = 0.0025$ and $\Delta q = 0.0075$ respectively. These chosen results show that the approximation abilities of the CMAC are similar in these two cases.

However, in the points of view of memory size, for $\Delta q = 0.0025$ the required memory size is 4940, 3.5 times greater than when $\Delta q = 0.0075$ ( $N_C = 1441$ ).The experimental enquire simulation results show that an optimal or nearly optimal structure carrying out a minimal modeling error could be achieved. Based on this observation, we try to design the algorithm which based on reinforcement learning, allowing to optimize the structure of the CMAC NN automatically.

# 4 CMAC Structural Parameters' Optimization

## 4.1 Structural Parameters Optimized with Modified Q-Learning Approach

In this section, our goal is to design a structure optimizing strategy allowing adjusting automatically the structural parameters of CMAC NN in order to make a tradeoff between the desirable approximation quality and the limited memory size.

Q-Learning, proposed by Watkins[8], is a very interesting way to use reinforcement learning strategy and is most advanced for which proofs of convergence exist. It does not require the knowledge of probability transitions from a state to another and is model-free. Here, the proposed optimize strategy is based on Q-Learning of temporal differences of order 0, while in our structure optimized approach only considering the following step. Take the number of layers and quantization step $\begin{bmatrix} N_l & \Delta q \end{bmatrix}$ as two dimension states of the world, while regarding the discrete actions as the increment of these two scalars. There are four possible actions when the agent explores the surrounding world as shown in relation (3), where $\delta_q$ is the incremental quantity of quantization step and the variation of layer is 1 for each step.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} N_l & \Delta q + \delta_q \\ N_l & \Delta q - \delta_q \\ N_l + 1 & \Delta q \\ N_l - 1 & \Delta q \end{bmatrix} \tag{3}$$

Change of the layer number and quantization step is supposed to be alternated. Each discrete time step, the agent observes state $[N_l^t, \Delta q^t]$, take action $a_i^t \in A^t$ $(i = 1, \cdots, 4)$, observes new state $[N_l^{t+1}, \Delta q^{t+1}]$, and receives immediate reward $r^t$. Transitions are probabilistic, that is, $[N_l^{t+1}, \Delta q^{t+1}]$ and $r^t$ are drawn from stationary probability distributions. In our approach, we choose Peseudo-stochastic method to describe the probability distributions.

The reinforcement signal $r^t$ provides information in terms of reward or punishment. In our case, on one hand, the reinforcement information has to take into account the approximation quality of network. On the other hand, the required memory size needs to be minimized within the limitation. Taking these considerations, the reinforcement signal is designed as three cases:

- $E_{square}^{t+1} < E_{square}^{t}$, the choice of structural parameters is in accordance with the correct direction.
  - If $E_{square}^{t}$ and $N_C^t$ achieve the desirable value then $r^t = 1$
  - Else, $r^t$ is determined according to (4). In equation (4) factor 1000 and 10 are designed only to balance the order of magnitude for memory size and approximation quality. $\alpha$ indicates the weight of these two structural parameters.

$$r^t = \frac{1}{\alpha N_C^t / 1000 + 10(1-\alpha) E_{square}^t} \tag{4}$$

- $E_{square}^{t+1} > E_{square}^{t}$, the trends of the chosen action is not appropriate: $r^t = -1$
- $E_{square}^{t+1} = E_{square}^{t}$, appropriateness of the trends of the chosen action is not clear: $r^t = 0$.

The Q matrix updates its evaluation of the value of the action while taking in account, the immediate reinforcement $r^t$ and the estimated value of the new state $V^t(N_l^{t+1}, \Delta q^{t+1})$, that is defined by (5), where $b$ is the action chosen within $A^t$.

$$V^t(N_l^{t+1}, \Delta q^{t+1}) = \max_{b \in A^{t+1}} Q(N_l^{t+1}, \Delta q^{t+1}, b) \tag{5}$$

If there is enough learning, the update equation could be written as (6), where $\gamma$ is discount factor and $\beta$ is the learning rate.

$$Q'(N_l^t, \Delta q^t, a^t) = (1-\beta) Q(N_l^t, \Delta q^t, a^t) + \beta[r^t + \gamma V^t(N_l^{t+1}, \Delta q^{t+1})] \tag{6}$$

The update corresponds to the barycenter of the old and the new rewards, weighted by $\beta$. If there comes up at the end of a period, then there is not appropriate state and the agent restarts a new sequence of training. The updating process is performed according to the equation (7).

$$Q'(N_l^t, \Delta q^t, a^t) = (1-\beta) Q(N_l^t, \Delta q^t, a^t) + \beta r^t \tag{7}$$

When the mean squared error satisfies the desirable approximation (refer to equation (8)), and the memory size is within the allocated rang as well (presented in equation (9)), the goal state is achieved.

$$\left| E_{square}^t \right| < \left| E_{square}^d \right| \tag{8}$$

$$N_C^t < N_C^d \tag{9}$$

### 4.2 Simulation Results and Convergence Analysis

Let us consider again the FSIN function approximation as an example. Suppose that the finite number of usable memory size is 1500, and the approximation error less

than 6.00% is favorable in order to maintain the approximation quality. In this case, we choose $\left|E^t_{square}\right| < 6.00\%$ and $N^t_C < 1500$ as the goal state in the training phase. The initial state of number of layer $N^0_l$ is set to be 20 and the quantization step can be chosen randomly within $[0.0000 \quad 0.0100]$, every 0.0002 as the incremental quantity $\delta_q$.

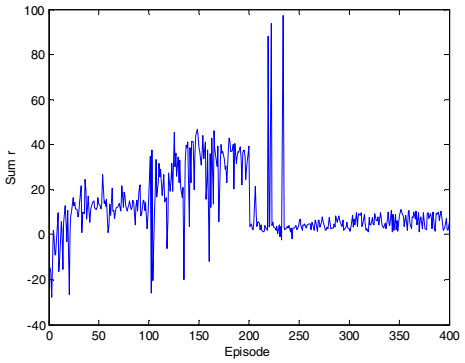



**Fig. 3.** Sum of $\Delta Q(t)$ for each episode

**Table 3.** Comparison between optimized and not optimized CMAC structure for FSIN function

| Structure Optimization | $\Delta q$ | $N_l$ | $E_{square}$ | $S_C$ | $N_C$ |
|---|---|---|---|---|---|
| NO | 0.0025 | 41 | 5.81% | 0.1225 | 4940 |
| YES | 0.0084 | 12 | 5.94% | 0.1008 | 1431 |

Figure 3 shows the sum of the computing value $\Delta Q(t)$ for each episode according to the number of episode. This updating value, which depends directly on the reinforcement signal, converges toward 3 within 250 episodes. The stability of our CMAC structure learning approach is theoretically guaranteed by the proof of the Q-learning convergence. When the learning stage is finished, then it is possible to obtain the maximum q-values in the matrix Q, which will lead to the optimized structural parameters of CMAC. A comparison of the optimized CMAC structure with the not optimized CMAC structure is given in Table 3. After the training phase, the CMAC with optimized structure ( $N_l = 12$ , $\Delta q = 0.0084$ ) can guarantee both of the desirable approximation quality and limitation required memory size.

# 5 Application of CMAC Optimization

## 5.1 Biped Robot's Gait Generation

In order to increase the robustness of control strategy for robot, CMAC neural network has been applied to learn a set of articular trajectories with popularity. However, the CPU of the robot has to do many intricacies tasks at the same time,

therefore the useable memory size is often allocated with restriction or fixed number and the precise control output is favorable. In this case, the structural parameters optimization problem is needed to be considered if we do not increase the complexity of the CMAC NN, since the simplicity of structure for network is always desirable. On the basis of our previous work which is on the gait pattern planning strategy of biped robot [9], The CMAC structural parameters optimization with modified Q-learning approach is applied to learn the joint angle trajectories of biped robot.

Usually, after footstep planning strategy, the position of the two stance feet can be calculated. Therefore, it is not difficult to derive the trajectory of the joint angle by inverse kinematics or bio-inspired approach. As this is not the emphasis of our statement, we use the inverse kinematics to generate the desirable joint angle trajectories to simplify the problem. Supposing in the stepping phase of robot, the stance leg does not bend. The geometrical relationship between stance leg and swing leg of biped robot is described in Figure 4.

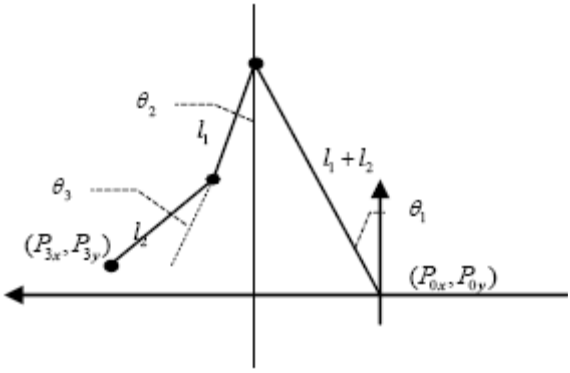



**Fig. 4.** Geometrical relationship between stance leg and swing leg

The angle between stance foot and vertical line $\theta_1$ and its position $(P_{0x}, P_{0y})$ are recorded as the initial condition. With inverse dynamics, the hip angle $\theta_2$ and knee angle $\theta_3$ of swing leg can be expressed as (10).

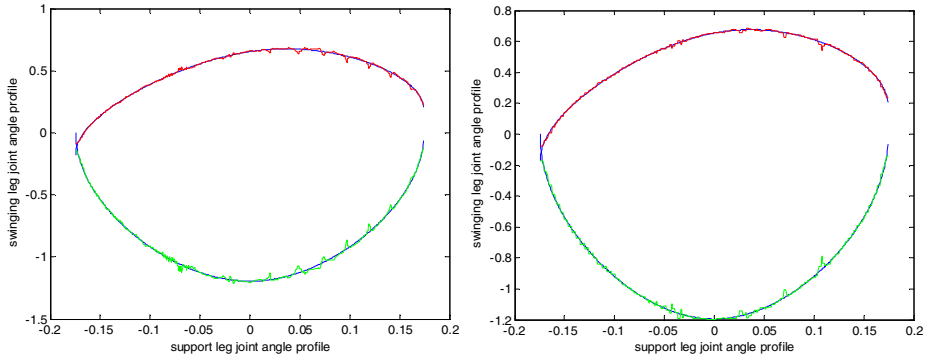
$$\theta_2 = atg(\frac{A}{B}) + atg(\frac{\sqrt{A^2+B^2-C^2}}{C})$$

With:
$$\begin{aligned} A &= 2l_1 \cdot [(l_1+l_2)\sin\theta_1 - P_{3x}] \\ B &= 2l_1 \cdot [P_{3y} - (l_1+l_2)\cos\theta_1] \\ C &= (l_1+l_2)\cdot[2P_{3x}\cdot\sin\theta_1 + 2P_{3y}\cdot\cos\theta_1 - (l_1+l_2)] + (l_2^2 - l_1^2 - P_{3x}^2 - P_{3y}^2) \\ F &= l_2\cos\theta_2 \\ G &= l_2\sin\theta_2 \\ H &= P_{3x} - (l_1+l_2)\sin\theta_1 - l_1\sin\theta_2 \end{aligned} \tag{10}$$

Based on our control strategy, each reference gait is characterized by both of the step length and step height which are associated with the position of stance and swinging

foot. Joint trajectory associated to one gait is memorized into one CMAC neural network. The biped robot is walking with a weighted average of several reference trajectories. Regarding the coordinate of swinging foot $(P_{3x}, P_{3y})$ as the two inputs, two CMAC neural networks are utilized for training hip joint angle $\theta_2$ and knee joint angle $\theta_3$ separately for each gait pattern. The weights of CMAC are updated based on the difference between the output of CMAC and reference hip joint angle (or knee joint angle) of swinging leg.

Figure 5 shows the results of swinging leg joint angle approximation with CMAC, in which blue curve stands for the reference joint angle profile, red one is the hip joint angle approximation, and green curve represents the output of CMAC approximating knee joint angle. In the first two simulation we do not know if the chosen structural parameters are appropriate. In the third experiment, the CMAC structural parameters are learning based on the developed Q-learning approach. we hope that the approximation error of reference gait less than $1.10\%$ is better and the pre-assigned memory size for each CMAC NN is 1000. According to equation (8) and (9), $\left|E^d_{square}\right| = 1.10\%$ and $N^d_C = 1000$ are set to be the goal state. After the learning phase, the optimized parameters are $N_l = 20$ , $\Delta q_1 = \Delta q_2 = 0.0051$ (refers to Fig.5(b)). The required memory size and approximation error are listed in Table 4 for these three experiments. In the first experiment, the calculated mean squared error ( $E^{\theta_2}_{square} = 1.16\%$ , $E^{\theta_3}_{square} = 1.19\%$ ) is very near to the desirable value, but the utilized memory size $N_C = 3589$ is 3.5 times bigger than the structure optimized example, since in this biped robot application case, several reference pattern gaits have to be stored, the total number of memory size become quickly very large. In second example, the memory size is desirable, however, the approximation quality ( $E^{\theta_2}_{square} = 1.52\%$ , $E^{\theta_3}_{square} = 1.56\%$ ) is much worse than the structure optimization case ( $E^{\theta_2}_{square} = 1.07\%$ , $E^{\theta_3}_{square} = 1.03\%$ ). The precise gait tracking is very important in the case of biped robot stepping over the obstacle.




(a $N_l = 30$ , $\Delta q_1 = \Delta q_2 = 0.0021$ (b): $N_l = 20$ , $\Delta q_1 = \Delta q_2 = 0.0051$

**Fig. 5.** Joint trajectory tracking with CMAC Neural Network: (a) CMAC structure without optimization process and (b) CMAC structure after Q-Learning based optimization

**Table 4.** Memory size and mean square error with randomly chosen and after learning CMAC structural parameters

| Structural parameters | $N_l$ | $\Delta q_1 = \Delta q_2$ | $E^{\theta_2}_{square}$ | $E^{\theta_3}_{square}$ | $N_C$ |
|---|---|---|---|---|---|
| Randomly chosen | 30 | 0.0021 | 1.16% | 1.19% | 3598 |
| Randomly chosen | 10 | 0.0080 | 1.56% | 1.52% | 795 |
| After learning | 20 | 0.0051 | 1.07% | 1.03% | 993 |

## 5.2 Robot's Arm Control for Work Piece Location

As the increasing of fixtureless assembly using robot arm, to precisely determine the location of the workpiece in order to design the intelligent grasping of robot is always desirable. The vision system is a good choice to estimate the target orientation and position automatically. For example, with the same side face up, a human worker place the workpiece randomly on the worktable, the X-Y position and orientation around Z axis are the key parameters which decide the trajectory and posture of robot grasping movement.

However, the camera position, workpiece models, and its relative position to the base or worktable are required to be modeled each time the target placing on the worktable. This will cause the redundant computing load to the system, which will affect the real time working or precisely assembly of the robot arm. According to previous study, CMAC neural network is desirable approach which is introduced to the machine vision system to generate the orientation of the workpiece [10, 11]. The proposed architecture constitutes two parts: feature detection and CMAC NN learning. (Refers to figure 6)

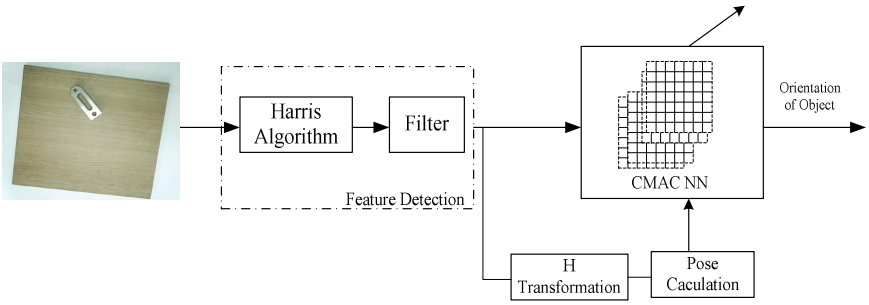



**Fig. 6.** Scheme of work piece location estimation based on feature detection and CMAC neural network

Harris algorithm is selected to detect the features of the image. After the filter of the background feature points, the selected features are taken as the inputs of CMAC NN. The purpose of we take the H-transformation before calculation of the work piece orientation is that we try to define the orientation based on the worktable whose axis is fixed to the robot coordinate. The work piece is placed on the worktable according to Fig.7(a). The detected features with Harris are shown in Figure 7(b). After H transformation and pose calculation, the image registration and affine transformation of work piece are shown in Figure 7(c) and (d) respectively. Regarding the X and Y pixels of detected features as two inputs of CMAC, the neural network is

trained with the pre-determined orientation of work piece which is calculated with feature detection approach.

In our tested example, the area of workplace is $38cm \times 34cm$, which the neural network training is done for the work piece placing spanning all orientations over. The image taken by the vision system is $640 \times 480$ pixels. Take the X pixel and Y pixel of the image as two inputs of CMAC separately, therefore, instead of the same quantification step for the two inputs of CMAC NN, the quantification step of the first input is set to be 4/3 bigger than the second. Also using the uniform CMAC quantization, in the first experiment, the number of layers is chosen to be 16 randomly, and the quantification step is $\Delta q_1 = 4$ for first input and $\Delta q_2 = 3$ for the second input, which means each receptive field of the first input contains 4 pixels while the second includes 3 pixels. In the second experiment, the structure parameters of CMAC are optimized with the approach based on Q-learning according to section 4. In the case of accurate robot assembly, the precise estimation of work piece orientation with CMAC NN is thought to be more important. Thus $\left|E^d_{square}\right| = 2.00\%$ and $N^d_C = 800$ are set to be the goal state. The training sets include 500 images with the work piece randomly placed within the workplace. After the training phase, we tested 20 images within which the pose of the work piece is not included in the tanning samples before. The estimation orientation of work piece for the two experiments are compared and listed in Table 5.

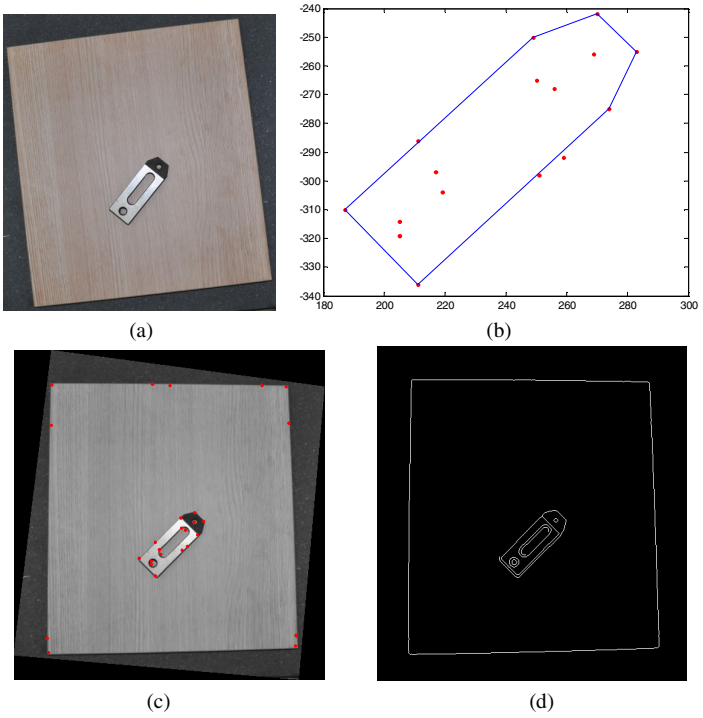



**Fig. 7.** Feature detection for the work piece. (a) original image, (b) features detection with Harris filter, (c) detected object and (d) affine transformation

**Table 5.** CMAC structure comparison with or without structure optimization for work piece orientation estimation

| Structure Optimization | $\Delta q_1$ | $\Delta q_2$ | $N_l$ | $E_{square}$ | $N_C$ |
|---|---|---|---|---|---|
| NO | 4 | 3 | 16 | 2.81% | 584 |
| YES | 8 | 6 | 20 | 1.54% | 491 |

Because of the good generalization ability of CMAC neural network, the attractive attribute of the presented approach is being able to generalize the efficient estimates of the object orientation after the supervised learning process. And it makes possible to learn the poses of the object without explicit modeling and the human intervention. Also, with the same scheme the location of the object could be generated after CMAC NN learning of the coordinates of the object from a series images. Furthermore, if the specific features of object, such as corners, arch etc., could be sufficiently learned, the pose of any work piece composed of the specific features can be generated.

## 6 Conclusions

Besides the appealing advantages of CMAC neural network, such as simple and effective training properties and fast learning convergence, a crucial problem to design CMAC is related to the choice of the neural network's structural parameters. In this paper, how both the number of layers and step quantization parameters influence the approximation qualities of CMAC neural networks is presented. The simulation results show that the minimal modeling error could be achieved by optimizing the structure parameters. Consequently, a CMAC structure optimization approach which is based on Q-learning is proposed. The stability of our proposed approach is theoretically guaranteed by the proof of the learning convergence of Q-learning. This Q-learning based structure optimization method is applied on the walk pattern generating for biped robot and work piece orientation estimation for robot arm assembly respectively. Simulation results show that the choice of CMAC's structure parameters after optimization allows decreasing the memory size and achieving the requirement of approximation quality at the same time.

**Acknowledgements.** This research is supported by the fundamental research fund of Northwestern Polytechnical University (JC20100211), P.R. China.

## References

1. Albus, J.S.: Data storage in the cerebellar model articulation controller (CMAC). Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control, 228–233 (1975)
2. Lu, H.-C., Yeh, M.-F., Chang, J.-C.: CMAC Study with adaptive Quantization. In: IEEE Int. Conf. on Systems, Man, and Cybernetics, Taipei, Taiwan, pp. 2596–2601 (2006)
3. Teddy, S.D., Lai, E.M.-K., Quek, C.: Hierarchically clustered adaptive quantization CMAC and its learning convergence. IEEE Trans. on Neural Networks 18(6), 1658–1682 (2007)

4. Menozzi, A., Chow, M.: On the training of a multi-resolution CMAC neural network. In: Proc. Int. Conf. Ind. Electron. Control Instrum., vol. 3, pp. 1130–1135 (1997)
5. Lin, C.-M., Chen, T.-Y.: Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems. IEEE Trans. on Neural Networks 20(9), 1377–1384 (2009)
6. Nguyen, M.N., Shi, D., Quek, C.: Self-organizing Gaussian fuzzy CMAC with truth value restriction. In: Proc. IEEE ICITA, Sydney, Australia, pp. 185–190 (2005)
7. Shi, D., Nguyen, M.N., Zhou, S., Yin, G.: Fuzzy CMAC with incremental Bayesian Ying-Yang learning and dynamic rule construction. IEEE Trans. on Systems, Man and Cybernetics 40(2), 548–552 (2010)
8. Watkins, C., Dayan, P.: Q-learning. Machine Learning, 279–292 (1992)
9. Yu, W., Sabourin, C., Madani, K., Yan, J.: Design of footstep planning controller for humanoid robot in dynamic environment. In: IEEE Int. Symp. on Knowledge Acquisition and Modeling, China, Wuhan (2008)
10. Carusone, J., D'Eleuterio, G.M.T.: The "FeatureCMAC": A Neural-Network-Based Vision System for Robotic Control. In: Proc.of the 1998 IEEE Int. Conf. on Robotics & Automation, Leuven, Belgium (1998)
11. Langley, C.S., D'Eleuterio, G.M.T.: Pose Estimation for Fixtureless Assembly Using a Feature CMAC Neural Network. In: 31st Int. Symp. on Robotics, Montreal, Canada (2000)

# Author Index