ADVANCES IN *SOFT COMPUTING* 41

Patricia Melin
Oscar Castillo
Eduardo Gómez Ramírez
Janusz Kacprzyk
Witold Pedrycz (Eds.)

Analysis and Design of Intelligent Systems Using Soft Computing Techniques

Springer

Advances in Soft Computing 41

Editor-in-Chief: J. Kacprzyk

Advances in Soft Computing

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

Further volumes of this series can be found on our homepage: springer.com

Abraham Ajith, Yasuhiko Dote,
Takeshi Furuhashi, Mario Köppen,
Azuma Ohuchi, Yukio Ohsawa (Eds.)
Soft Computing as Transdisciplinary Science and Technology, 2005
ISBN 978-3-540-25055-5

Barbara Dunin-Keplicz, Andrzej
Jankowski, Andrzej Skowron,
Marcin Szczuka (Eds.)
Monitoring, Security, and Rescue Techniques in Multiagent Systems, 2005
ISBN 978-3-540-23245-2

Frank Hoffmann, Mario Köppen,
Frank Klawonn, Rajkumar Roy (Eds.)
Soft Computing Methodologies and Applications, 2005
ISBN 978-3-540-25726-4

Mieczyslaw A. Klopotek, Slawomir T.
Wierzchon, Kryzysztof Trojanowski
(Eds.)
Intelligent Information Processing and Web Mining, 2005
ISBN 978-3-540-25056-2

Abraham Ajith, Bernard de Bacts,
Mario Köppen, Bertram Nickolay (Eds.)
Applied Soft Computing Technologies: The Challenge of Complexity, 2006
ISBN 978-3-540-31649-7

Mieczyslaw A. Klopotek, Slawomir T.
Wierzchon, Kryzysztof Trojanowski
(Eds.)
Intelligent Information Processing and Web Mining, 2006
ISBN 978-3-540-33520-7

Ashutosh Tiwari, Joshua Knowles,
Erel Avineri, Keshav Dahal,
Rajkumar Roy (Eds.)
Applications and Soft Computing, 2006
ISBN 978-3-540-29123-7

Bernd Reusch, (Ed.)
Computational Intelligence, Theory and Applications, 2006
ISBN 978-3-540-34780-4

Miguel López-Díaz, María ç. Gil,
Przemysław Grzegorzewski, Olgierd
Hryniewicz, Jonathan Lawry
Soft Methodology and Random Information Systems, 2006
ISBN 978-3-540-34776-7

Ashraf Saad, Erel Avineri, Keshav Dahal,
Muhammad Sarfraz, Rajkumar Roy (Eds.)
Soft Computing in Industrial Applications, 2007
ISBN 978-3-540-70704-2

Bing-Yuan Cao (Ed.)
Fuzzy Information and Engineering, 2007
ISBN 978-3-540-71440-8

Patricia Melin, Oscar Castillo,
Eduardo Gómez Ramírez, Janusz Kacprzyk,
Witold Pedrycz (Eds.)
Analysis and Design of Intelligent Systems Using Soft Computing Techniques, 2007
ISBN 978-3-540-72431-5

Patricia Melin, Oscar Castillo,
Eduardo Gómez Ramírez, Janusz Kacprzyk,
Witold Pedrycz (Eds.)

Analysis and Design of Intelligent Systems Using Soft Computing Techniques

Springer

Editors

Prof. Dr. Patricia Melin
Dept. of Computer Science,
Tijuana Institute of Technology,
Tijuana, Mexico
P.O. Box 4207
Chula Vista CA 91909, USA
E-mail: pmelin@tectijuana.mx

Prof. Dr. Oscar Castillo
Dept. of Computer Science,
Tijuana Institute of Technology,
Tijuana, Mexico
P.O. Box 4207
Chula Vista CA 91909, USA
E-mail: ocastillo@tectijuana.mx

Eduardo Gómez Ramírez
Laboratorio de Investigación y
Desarrollo de Tecnología
Avanzada Lidetea,
Coordinación General De
Investigacion, Universidad La Salle,
Benjamin Franklin 47 Col. Condesa, CP
06140 México, D.F., Mexico
E-mail: egr@ulsa.edu.mx

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

Prof. Witold Pedrycz
Department of Electrical Engineering,
University of Manitoba,
Engineering Building,
15 Gillson St., Winnipeg,
Manitoba, Canada, R3T 5V6
E-mail: pedrycz@ece.ualberta.ca

Library of Congress Control Number: 2007926020

ISSN print edition: 1615-3871
ISSN electronic edition: 1860-0794
ISBN-10 3-540-72431-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72431-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: by the authors and SPS using a Springer LaTeX macro package

Printed on acid-free paper SPIN: 12060320 89/SPS 5 4 3 2 1 0

Preface

This book comprises a selection of papers from IFSA 2007 on new methods for analysis and design of hybrid intelligent systems using soft computing techniques. Soft Computing (SC) consists of several computing paradigms, including fuzzy logic, neural networks, and genetic algorithms, which can be used to produce powerful hybrid intelligent systems for solving problems in pattern recognition, time series prediction, intelligent control, robotics and automation. Hybrid intelligent systems that combine several SC techniques are needed due to the complexity and high dimensionality of real-world problems. Hybrid intelligent systems can have different architectures, which have an impact on the efficiency and accuracy of these systems, for this reason it is very important to optimize architecture design. The architectures can combine, in different ways, neural networks, fuzzy logic and genetic algorithms, to achieve the ultimate goal of pattern recognition, time series prediction, intelligent control, or other application areas.

This book is intended to be a major reference for scientists and engineers interested in applying new computational and mathematical tools to design hybrid intelligent systems. This book can also be used as a reference for graduate courses like the following: soft computing, intelligent pattern recognition, computer vision, applied artificial intelligence, and similar ones. The book is divided in to twelve main parts. Each part contains a set of papers on a common subject, so that the reader can find similar papers grouped together. Some of these parts are comprised from papers of organized sessions of IFSA 2007 and we thank the session's organizers for their incredible job on forming these sessions with invited and regular papers. We begin the book with an introductory paper by the father of Fuzzy Logic, Prof. Lotfi Zadeh, which describes future lines of research for the area.

In Part I, we begin with a set of papers on "Type-2 Fuzzy Logic: theory and applications" that describe different contributions to the theory of type-2 fuzzy logic and its applications to real-world problems. The papers on interval type-2 fuzzy logic theory provide new concepts and tools that can be used for future applications, while the papers on applications show results of using interval type-2 fuzzy logic on problems of control, time series prediction and pattern recognition.

In Part II, we have a set of papers on "Fuzzy Clustering: theory and applications" that describe different contributions to the theory of fuzzy clustering and its applications to real-world problems. The theoretical papers on fuzzy clustering show advances to the methods of clustering, while the papers on applications show real-world problems solved using different fuzzy clustering algorithms.

In Part III, we have a set of papers on "Intelligent Identification and Control" that describe different contributions to the theory of system identification and intelligent control with fuzzy logic and its applications to real-world problems. The theoretical papers on intelligent identification and control show advances to the methods of obtaining models and achieving intelligent control of non-linear plants, while the papers on applications show real-world problems solved using different fuzzy logic or hybridmethods.

In Part IV, we have a set of papers on "Time Series Prediction" that describe different contributions to the theory and methodologies for time series prediction and its applications to real-world problems. The theoretical papers on times series prediction show advances to the methods of prediction and forecasting, while the papers on applications show real-world problems solved using different fuzzy logic methods.

In Part V, we have a set of papers on "Pattern Recognition" that describe different contributions to the theory and methodologies for pattern recognition and its applications to real-world problems. The theoretical papers on pattern recognition show advances to the methods for image recognition, while the papers on applications show real-world problems solved using different fuzzy logic and hybrid models. In particular, there are papers on face, fingerprint and voice recognition.

In Part VI, we have a set of papers on "Evolutionary Computation" that describe different contributions to the theory and methodologies of evolutionary computing and its applications to real-world problems. The theoretical papers on evolutionary computing show advances to the methods of optimization with search techniques, while the papers on applications show real-world problems solved using different evolutionary algorithms.

In Part VII, we have a set of papers on "Fuzzy Modeling" that describe different contributions to the theory and methodologies of fuzzy logic modeling and its applications to real-world problems. The theoretical papers on fuzzy modeling show advances to the methods of modeling problems using fuzzy systems, while the papers on applications show real-world problems solved using different fuzzy logic methods.

In Part VIII, we have a set of papers on "Intelligent Manufacturing and Scheduling" that describe different contributions to the theory and methodologies for intelligent manufacturing and scheduling and its applications to real-world problems. The theoretical papers on intelligent manufacturing show advances to the methods of scheduling, quality control and production planning, while the papers on applications show real-world problems solved using different fuzzy logic and hybrid methods.

In Part IX, we have a set of papers on "Intelligent Agents" that describe different contributions to the theory and methodologies of building and designing intelligent agents and its applications to real-world problems. The theoretical papers on intelligent agents show advances to the methods of agent's creation and learning, while the papers on applications show real-world problems solved using different architectures of intelligent agents.

In Part X, we have a set of papers on "Neural Networks Theory" that describe different contributions to the theory and methodologies for designing and building neural network models and its applications to real-world problems. The theoretical papers on neural networks show advances to the methods of learning and design of neural models, while the papers on applications show real-world problems solved using different neural network models.

In Part XI, we have a set of papers on "Robotics" that describe different contributions to the theory and methodologies for robot control and identification and its applications to real-world problems. The theoretical papers on robotics show advances to the methods of intelligent control and planning of autonomous robots and manipulators, while the papers on applications show real-world problems solved using different fuzzy logic methods.

In Part XII, we have a set of papers on “Fuzzy Logic Applications”, that describe different contributions to the theory and applications to real-world problems of fuzzy models. The theoretical papers on fuzzy logic show advances to the methods of fuzzy modeling, while the papers on applications show real-world problems solved using different fuzzy logic methods.

We end this preface of the book by giving thanks to all the people who have help or encourage us during the making of this book. We would like to thank our colleagues working in Soft Computing, which are too many to mention each by their name. Of course, we need to thank our supporting agencies in our countries for their help during this project. We have to thank our institutions for always supporting our projects.

Prof. Dr. Oscar Castillo
Tijuana Institute of Technology,
Mexico

Prof. Dr. Patricia Melin
Tijuana Institute of Technology,
Mexico

Prof. Dr. Eduardo Gomez-Ramirez
La Salle University,
Mexico

Prof. Dr. Janusz Kacprzyk
Polish Academy of Sciences,
Poland

Prof. Dr. Witold Pedrycz
University of Alberta, Canada

Organization

IFSA 2007 is organized by the International Fuzzy Systems Association (IFSA), the Hispanic American Fuzzy Systems Association (HAFSA), and the Division of Graduate Studies of Tijuana Institute of Technology, with support of DGEST and CONACYT of Mexico.

Executive Committee

Honorary Chair

Lotfi Zadeh
University of California at Berkeley

General Chair

Oscar Castillo
Tijuana Institute of Technology

Program Chair

Patricia Melin
Tijuana Institute of Technology

Advisory Committee

Janusz Kacprzyk
Polish Academy of Sciences

Witold Pedrycz
University of Alberta

Ronald R. Yager
Iona College, USA

Panels Chair

Masoud Nikravesh
University of California at Berkeley

Workshop Chair

Eduardo Gomez-Ramirez
La Salle University, Mexico

International Program Committee

Ildar Batyrshin, Mexican Petroleum Institute, Mexico
Z. Zenn Bien, KAIST, Korea
Ulrich Bodenhofer, Johannes Kepler University, Austria
Piero Bonissone, GE Global Research, USA
Patrick Bosc, IRISA/ENSSAT, France
Bernadette Bouchon-Meunier, University Pierre Marie Curie, France
Christer Carlsson, IAMSR/Abo Akademi University, Finland
Joao P.B. Carvalho, INESC, TU Lisbon, Portugal
Oscar Castillo, Tijuana Institute of Technology, Mexico
Carlos Coello, CINVESTAV, Mexico
Oscar Cordon, European Centre for Soft-Computing, Spain
Guoqing Chen, Tsinghua University, China
Bernard De Baets, Ghent University, Belgium
Hai-Bin Duan, Beihang University, China
Pablo Estevez, University of Chile, Chile
Dimitar Filev, Ford Research Center, USA
Takeshi Furuhashi, Nagoya University, Japan
Eduardo Gomez-Ramirez, La Salle University, Mexico
Fernando Gomide, State University of Campinas, Brazil
Madan M. Gupta, University of Saskatchewan, Canada
Mustafa Gunes, King Abdulaziz University, Saudi Arabia
Janos Grantner, Western Michigan University, USA
Kaouru Hirota, Tokyo Institute of Technology, Japan
Hani Hagras, University of Essex, UK
Robert John, De Montfort University, UK
Cengiz Kahraman, Istanbul Technical University, Turkey
Janusz Kacprzyk, Polish Academy of Sciences, Poland
Jim Keller, University of Missouri, USA
Etienne Kerre, Ghent University, Belgium
George Klir, Binghamton University-SUNY, USA
Lazlo Koczy, Budapest University of Technology and Economics, Hungary
Rudolf Kruse, Universitaet Magdeburg, Germany
Vladik Kreinovich, University of Texas, El Paso, USA
Reza Langari, Texas A & M University, USA
Vincenzo Loia, Universita di Salerno, Italy
Luis Magdalena, European Centre for Soft Computing, Spain
E. Mamdani, University of London, UK
Patricia Melin, Tijuana Institute of Technology, Mexico
Jerry Mendel, University of Southern California, USA
Evangelia Micheli-Tzanakou, Rutgers University, USA
K.C. Min, Korea
Ying Mingsheng, Tsinghua University, China
Masoud Nikravesh, University of California, Berkeley, USA
V. Novak, University of Ostrava, Czech Republic
Jose A. Olivas-Varela, Universidad de Castilla La Mancha, Spain

Gabriella Pasi, Universita degli Studi di Milano, Italy
Witold Pedrycz, University of Alberta, Canada
Frank Chung-Hoon Rhee, Hanyang University, Korea
Rajkumar Roy, Cranfield University, UK
Edgar N. Sanchez, CINVESTAV-Guadalajara, Mexico
Pilar Sobrevilla, Universidad Politecnica de Cataluna, Spain
Joao M.C. Sousa, IDMEC, TU Lisbon, Portugal
Thomas Sudkamp, Wright State University, USA
Michio Sugeno, Doshisha University, Japan
Ricardo Tanscheit, Catholic University of Rio de Janeiro, Brazil
Tsu-Tian Lee, National Taipei University of Technology, Taiwan
Enric Trillas, European Centre for Soft Computing, Spain
I. Burhan Turksen, Economics and Technology University, Turkey, University of Toronto, Canada
Jose L. Verdegay, University of Granada, Spain
Michael Wagenknecht, Univ. of Applied Sciences Zittau, Germany
Ronald R. Yager, Iona College, USA
Liu Yingming, Sichuan University, China
Hans Zimmerman, Aachen University of Technology, Germany
Jacek Zurada, University of Louisville, USA

Local Organizing Committee

Luis Tupak Aguilar Bustos,
CITEDI-IPN, Tijuana
Arnulfo Alanis,
Tijuana Institute of Technology, Tijuana
Oscar Castillo
Tijuana Institute of Technology, Tijuana
Carlos A. Coello Coello,
CINVESTAV, Mexico
Mario Garcia V.
Tijuana Institute of Technology, Tijuana
Pilar Gómez Gil,
Universidad de las Americas Puebla, Puebla
Eduardo Gómez Ramírez,
Universidad de La Salle, México
Miguel Angel Lopez
Tijuana Institute of Technology, Tijuana
Gerardo Maximiliano Méndez,
Instituto Tecnológico de Nuevo León, Monterrey
Alejandra Mancilla
Tijuana Institute of Technology, Tijuana
Patricia Melin
Tijuana Institute of Technology, Tijuana
Oscar Montiel Ross
CITEDI-IPN, Tijuana

Antonio Rodríguez Díaz,
Univ. Autónoma de Baja California, Tijuana
Edgar N. Sánchez,
CINVESTAV, Guadalajara
Roberto Sepúlveda Cruz
CITEDI-IPN, Tijuana

List of Additional Reviewers

Troels Andreassen, Denmark
Gloria Bordogna, Italy
Jesús Campana, Spain
Eliseo Clementini, Italy
Oscar Cordon, Spain
Fabio Crestani, Switzerland
Guy De Tre, Belgium
Marcin Detyniecki, France
Axel Hallez, Belgium
Enrique Herrera-Viedma, Spain
Janusz Kacprzyk, Poland
Donald Kraft, USA
Zongmin Ma, China
Luis Mancilla, Mexico
Milan Mares, Czech Republic
Nicolas Marin, Spain
Christophe Marsala, France
Maria-Jose Martin-Bautista, Spain
Tom Matthe, Belgium
Oscar Montiel, Mexico
Adam Niewiadomski, Poland
Gabriella Pasi, Italy
Frederick E. Petry, USA
Olivier Pivert, France
Olga Pons, Spain
Ellie Sanchez, France
Roberto Sepúlveda, Mexico
Eulalia Szmidt, Poland
Joerg Verstraete, Belgium
Manolis Wallace, Greece
Slawomir Zadrozny, Poland

Contents

Part I: Type-2 Fuzzy Logic: Theory and Applications

Part II: Fuzzy Clustering: Theory and Applications

Part III: Intelligent Identification and Control

Part IV: Time Series Prediction

Part V: Pattern Recognition

Part VI: Evolutionary Computation

Part VII: Fuzzy Modeling

Part VIII: Intelligent Manufacturing and Scheduling

Part IX: Intelligent Agents

Part X: Neural Networks Theory

Part XI: Robotics

Part XII: Fuzzy Logic Applications

Fuzzy Logic as the Logic of Natural Languages*

Lotfi A. Zadeh

Department of EECS, University of California, Berkeley, CA 94720-1776;
Telephone: 510-642-4959; Fax: 510-642-1712
zadeh@eecs.berkeley.edu

Abstract. As we move further into the age of machine intelligence and automated decision-making, a long-standing problem becomes more pressing and more complex. The problem is: How to reason and compute with information described in natural language. The basic importance of this problem derives from the fact that much of human knowledge—and especially world knowledge—is expressed in natural language.

Existing techniques for dealing with the problem are based, for the most part, on bivalent logic. Bivalent-logic-based theories of natural language have achieved a measure of success, but a basic question which arises is: Can bivalent-logic-based theories lead to the solution of many complex problems which are encountered in the realm of natural language understanding? In my view, the answer is: No. The reason is rooted in a fundamental mismatch between the precision of bivalent-logic-based methods and the imprecision of natural languages—especially in the realm of semantics.

A natural language is basically a system for describing perceptions. Perceptions are intrinsically imprecise, reflecting the bounded ability of human sensory organs, and ultimately the brain, to resolve detail and store information. Imprecision of perceptions is passed on to natural languages. More specifically, in natural language almost everything is a matter of degree that is, fuzzy. By contrast, in bivalent logic every proposition is either true or false, with no shades of truth allowed.

A radical view which is articulated in my lecture is that the foundation for theories of natural language should be shifted from bivalent logic to fuzzy logic—a logic in which, as in natural language, everything is a matter of degree. Furthermore, in fuzzy logic everything is or is allowed to be granulated, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality. Granulation is what humans employ in coping with the imprecision of perceptions. It should be noted that granulation is the basis for the concept of a linguistic variable—a concept which plays a pivotal role in almost all applications of fuzzy logic.

Giving the entrenchment of bivalent logic in natural language theories, a shift from bivalent logic to fuzzy logic will certainly be hard to accept. Still, acceptance in my view, is only a matter of time.

A shift from bivalent logic to fuzzy logic has many ramifications. In my lecture, attention is focused on computation with information described in natural language, or NL-Computation for short. NL-Computation is closely related to Computing with Words.

In conventional modes of computation, the objects of computation are values of variables. In NL-Computation, the objects of computation are not values of variables, but states of information about the values of variables, with the added assumption that information is described in natural language. A simple example: *f* is a function from reals to reals, with *Y*=*f*(*X*). Described

[*] Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Omron Grant, Tekes Grant, Chevron Texaco Grant and the BISC Program of UC Berkeley.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 1–2, 2007.
springerlink.com

in a natural language, the information about X is: if X is small then Y is small; if X is medium then Y is large; if X is large then Y is medium. Similarly, the information about X is: usually X is medium. What is the valued of Y? Another simple example: (a) overeating causes obesity; and (b) obesity causes high blood pressure. I overeat and am obese. What is the probability that I will develop high blood pressure?

More generally, the point of departure in NL-Computation is (a) an input dataset, p, which consists of a system of propositions expressed in natural language; and (b) a question, q, likewise expressed in natural language. The output dataset is an answer to q, ans $(q|p)$. The input dataset is closed (circumscribed) if no propositions are allowed to be added to the input dataset; and it is open if propositions can be added from an external knowledge base, especially world knowledge. The default assumption is that the input dataset is closed.

In general, propositions in the input dataset are not well-posed for computation. In NL-Computation, the first step is precisiation of meaning of p and q. Precisiation is carried out through translation of p and q into what is termed a precisiation language, PL. Conventional meaning representation languages are based on bivalent logic and are not expressive enough to serve the function of precisiation. In NL-Computation, there are two tracks: (a) precisiation through translation into PRUF (Possibilistic Relational Universal Fuzzy) through the use of test-score semantics (Zadeh 1978, 1982); and (b) translation into GCL (Generalized Constraint Language) (Zadeh 2006). In my lecture, attention is focused on translation into GCL—a precisiation language which is maximally expressive.

The concept of a generalized constraint plays a pivotal role in NL-Computation. Briefly, a generalized constraint is expressed as X isr R, where X is the constrained variable, R is a constraining relation and r is an indexical variable which defines the way in which R constrains X. The principal constraints are possibilistic, veristic, probabilistic, usuality, random set, fuzzy graph and group. Generalized constraints may be combined, qualified, propagated, and counterpropagated, generating what is called the Generalized Constraint Language, GCL. The key underlying idea is that information conveyed by a proposition may be represented as a generalized constraint, that is, as an element of GCL.

In our approach, NL-Computation involves three modules: (a) Precisiation module; (b) Protoform module; and (c) Computation module. An object of precisiation, p, is referred to as precisiend, and the result of precisiation, p^*, is called a precisiand. Usually, a precisiend is a proposition, a system of propositions or a concept. A precisiend may have many precisiands. Definition is a form of precisiation. A precisiand may be viewed as a model of meaning. The degree to which the intension (attribute-based meaning) of p^* approximates to that of p is referred to as cointension. A precisiand, p^*, is cointensive if its cointension with p is high, that is, if p^* is a good model of meaning of p.

The Protoform module serves as an interface between Precisiation and Computation modules. In essence, its function is that of summarization, abstraction and generalization.

The Computation module is a system of rules of deduction. Basically, the rules of deduction are the rules which govern propagation and counterpropagation of generalized constraints. The principal rule of deduction is the extension principle (Zadeh 1965, 1975). Basically, the extension principle relates the generalized constraint on the output dataset to that on the input dataset. In this way, an answer to the question, q, is related to a generalized constraint on the input dataset, p.

The generalized-constraint-based computational approach to NL-Computation opens the door to a wide-ranging enlargement of the role of natural languages in scientific theories. Particularly important application areas are decision-making with information described in natural language, economics, systems analysis, risk assessment, search, question-answering and theories of evidence.

Part I

Type-2 Fuzzy Logic: Theory and Applications

A Method for Response Integration in Modular Neural Networks with Type-2 Fuzzy Logic for Biometric Systems

Jérica Urías, Denisse Hidalgo, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology, Tijuana, Mexico

Abstract. We describe in this paper a new method for response integration in modular neural networks using type-2 fuzzy logic. The modular neural networks were used in human person recognition. Biometric authentication is used to achieve person recognition. Three biometric characteristics of the person are used: face, fingerprint, and voice. A modular neural network of three modules is used. Each module is a local expert on person recognition based on each of the biometric measures. The response integration method of the modular neural network has the goal of combining the responses of the modules to improve the recognition rate of the individual modules. We show in this paper results of a type-2 fuzzy approach for response integration that improves performance over type-1 fuzzy logic approaches.

1 Introduction

Today, a variety of methods and techniques are available to determine unique identity, the most common being fingerprint, voice, face, and iris recognition [10]. Of these, fingerprint and iris offer a very high level of certainty as to a person's identity, while the others are less exact. A large number of other techniques are currently being examined for suitability as identity determinants. These include (but are not limited to) retina, gait (walking style), typing style, body odour, signature, hand geometry, and DNA. Some wildly esoteric methods are also under development, such as ear structure, thermal imaging of the face and other parts of the body, subcutaneous vein patterns, blood chemistry, anti-body signatures, and heart rhythm, to name a few [14].

The four primary methods of biometric authentication in widespread use today are face, voice, fingerprint, and iris recognition. All of these are supported in our approach, some more abundantly than others. Generally, face and voice are considered to be a lower level of security than fingerprint and iris, but on the other hand, they have a lower cost of entry. We describe briefly in this section some of these biometric methods.

Face Recognition. Facial recognition has advanced considerably in the last 10 to 15 years. Early systems, based entirely on simple geometry of key facial reference points, have given way to more advanced mathematically-based analyses such as Local Feature Analysis and Eigenface evaluation. These have been extended though the addition of "learning" systems, particularly neural networks.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 5–15, 2007.
springerlink.com

Face recognition systems are particularly susceptible to changes in lighting systems. For example, strong illumination from the side will present a vastly different image to a camera than neutral, evenly-positioned fluorescent lighting. Beyond this, however, these systems are relatively immune to changes such as weight gain, spectacles, beards and moustaches, and so on. Most manufacturers of face recognition systems claim false accept and false reject rates of 1% or better.

Voice Recognition. Software systems are rapidly becoming adept at recognising and converting free-flowing speech to its written form. The underlying difficulty in doing this is to flatten out any differences between speakers and understand everyone universally. Alternatively, when the goal is to specifically identify one person in a large group by their voice alone, these very same differences need to be identified and enhanced.

As a means of authentication, voice recognition usually takes the form of speaking a previously-enrolled phrase into a computer microphone and allowing the computer to analyse and compare the two sound samples. Methods of performing this analysis vary widely between vendors. None is willing to offer more than cursory descriptions of their algorithms--principally because, apart from LAN authentication, the largest market for speaker authentication is in verification of persons over the telephone.

Fingerprint Recognition. The process of authenticating people based on their fingerprints can be divided into three distinct tasks. First, you must collect an image of a fingerprint; second, you must determine the key elements of the fingerprint for confirmation of identity; and third, the set of identified features must be compared with a previously-enrolled set for authentication. The system should never expect to see a complete 1:1 match between these two sets of data. In general, you could expect to couple any collection device with any algorithm, although in practice most vendors offer proprietary, linked solutions.

A number of fingerprint image collection techniques have been developed. The earliest method developed was optical: using a camera-like device to collect a high-resolution image of a fingerprint. Later developments turned to silicon-based sensors to collect an impression by a number of methods, including surface capacitance, thermal imaging, pseudo-optical on silicon, and electronic field imaging.

As discussed, a variety of fingerprint detection and analysis methods exist, each with their own strengths and weaknesses. Consequently, researchers vary widely on their claimed (and achieved) false accept and false reject rates. The poorest systems offer a false accept rate of around 1:1,000, while the best are approaching 1:1,000,000. False reject rates for the same vendors are around 1:100 to 1:1000.

2 Proposed Approach for Recognition

Our proposed approach for human recognition consists in integrating the information of the three main biometric parts of the person: the voice, the face, and the fingerprint [14]. Basically, we have an independent system for recognizing a person from each of its biometric information (voice, face, and fingerprint), and at the end we have an integration unit to make a final decision based on the results from each of the modules. In Figure 1 we show the general architecture of our approach in which it is clearly seen that we have one module for voice, one module for face recognition, and one module for fingerprint recognition. At the top, we have the decision unit integrating the results from the three modules. In this paper the decision unit is implemented with a type-2 fuzzy system.

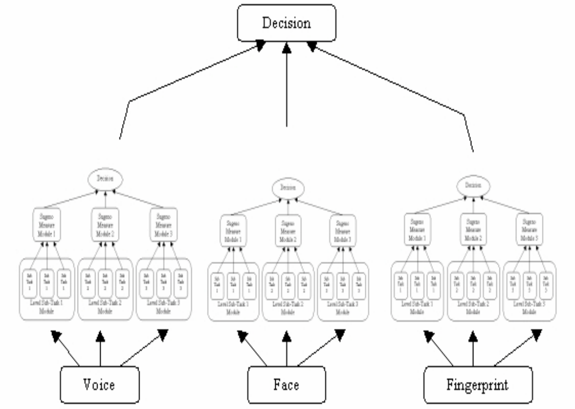


Fig. 1. Architecture of the proposed modular approach

3 Modular Neural Networks

This section describes a particular class of "modular neural networks", which have a hierarchical organization comprising multiple neural networks; the architecture basically consists of two principal components: local experts and an integration unit, as illustrated in Figure 2. In general, the basic concept resides in the idea that combined (or averaged) estimators may be able to exceed the limitation of a single estimator [3]. The idea also shares conceptual links with the "divide and conquer" methodology. Divide and conquer algorithms attack a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the complex problem [1] [6] [13]. When using a modular network, a given task is split up among several local experts NNs [4]. The average load on each NN is reduced in comparison with a single NN that must learn the entire original task, and thus the combined model may be able to surpass the limitation of a single NN. The outputs of a certain number of local experts (O_i) are mediated by an integration unit. The integrating unit puts the outputs together using estimated combination weights (g_i). The overall output Y is

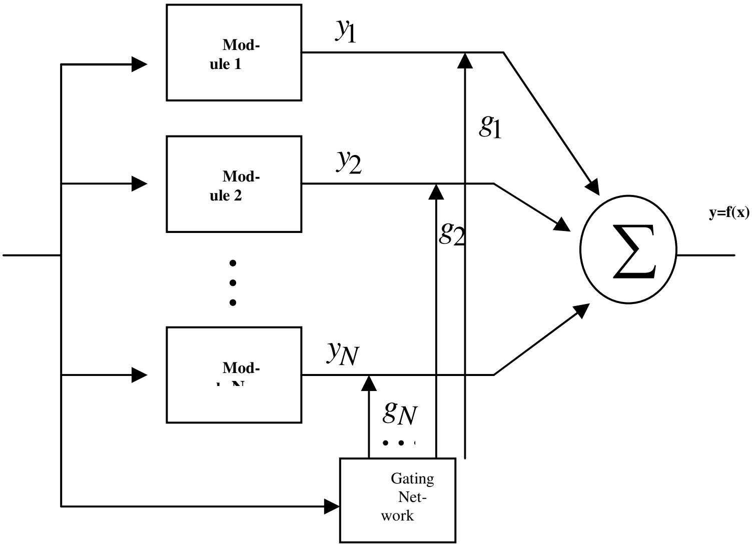


Fig. 2. Architecture of a modular neural network

$$Y_i = \Sigma g_i \, O_I \qquad (1)$$

Nowlan, Jacobs, Hinton, and Jordan [5] described modular networks from a competitive mixture perspective. That is, in the gating network, they used the "softmax" function, which was introduced by McCullagh and Nelder [7]. More precisely, the gating network uses a softmax activation g_i of ith output unit given by

$$G_i = \exp\,(ku_i)/\ \Sigma_j \exp\,(ku_j) \qquad (2)$$

Where u_i is the weighted sum of the inputs flowing to the ith output neuron of the gating network. Use of the softmax activation function in modular networks provides a sort of "competitive" mixing perspective because the ith local expert's output O_i with a minor activation u_i does not have a great impact on the overall output Y_i.

4 Integration of Results for Person Recognition Using Fuzzy Logic

On the past decade, fuzzy systems have displaced conventional technology in different scientific and system engineering applications, especially in pattern recognition and control systems. The same fuzzy technology, in approximation reasoning form, is resurging also in the information technology, where it is now giving support to decision making and expert systems with powerful reasoning capacity and a limited quantity of rules [17]. For the case of modular neural networks, a fuzzy system can be used as an integrator or results [10].

The fuzzy sets were presented by L. A. Zadeh in 1965 to process / manipulate data and information affected by unprobabilistic uncertainty / imprecision [15]. These were designed to mathematically represent the vagueness and uncertainty of linguistic problems; thereby obtaining formal tools to work with intrinsic imprecision in different type of problems; it is considered a generalization of the classic set theory.

Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way. These type-2 fuzzy sets were originally presented by Zadeh in 1975 and are essentially "fuzzy fuzzy" sets where the fuzzy degree of membership is a type-1 fuzzy set [16]. The new concepts were introduced by Mendel [11] [12] allowing the characterization of a type-2 fuzzy set with a superior membership function and an inferior membership function; these two functions can be represented each one by a type-1 fuzzy set membership function. The interval between these two functions represent the footprint of uncertainty (FOU), which is used to characterize a type-2 fuzzy set. The uncertainty is the imperfection of knowledge about the natural process or natural state. The statistical uncertainty is the randomness or error that comes from different sources as we use it in a statistical methodology [2].

5 Modular Neural Networks with Type-2 Fuzzy Logic as a Method for Response Integration

As was mentioned previously, type-2 fuzzy logic was used to integrate the responses of the three modules of the modular network. Each module was trained with the corresponding data, i.e. face, fingerprint and voice. Also, a set of modular neural networks was built to test the type-2 fuzzy logic approach of response integration. The architecture of the modular neural network is shown in Figure 3. From this figure we can appreciate that each module is also divided in three parts with the idea of also dividing each of the recognition problems in three parts.

Experiments were performed with sets of 20 and 30 persons. The trainings were done with different architectures, i.e. different number of modules, layers and nodes.

As can be appreciated from Figure 3, the first module was used for training with voice data. In this case, three different words were used for each person. The words used were: access, presentation, and hello.

The second module was used for traing with person face data. In this case, two different photos were taken from each person, one in a normal position and the other with noise. The idea is that training with noise will make the recognition more robust to changes in the real world. We show in Figure 4 the photos of two persons in a normal situation and in a noisy situation.

The third module was used with fingerprint data of the group of persons. The fingerprint information was taken with a scanner. Noise was added for training the neural networks.

In all cases, each module is subdivided in three submodules, in this way making easier the respective recognition problem.

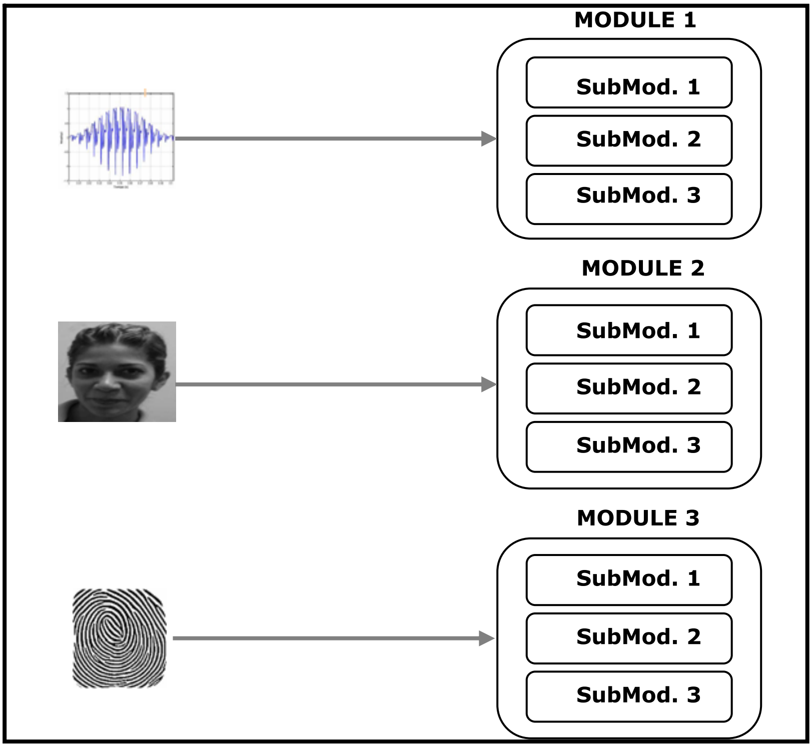


Fig. 3. Architecture of the Modular Network used for the recognition problem

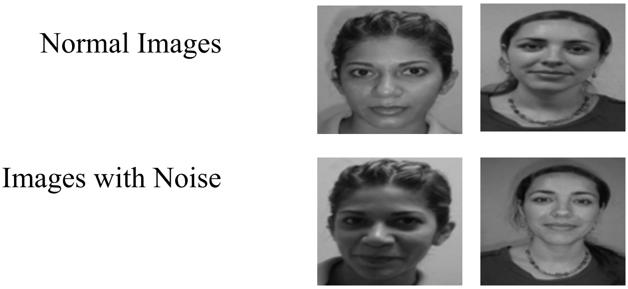
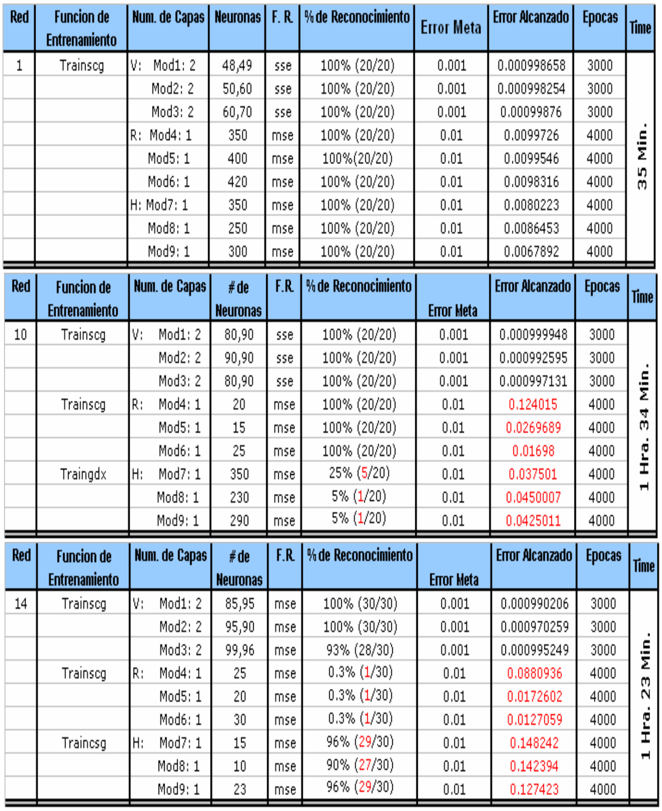


Fig. 4. Sample Photos of Faces in a Normal and Noisy Situation

6 Simulation Results

A set of different trainings for the modular neural networks was performed to test the proposed type-2 fuzzy logic approach for response integration in modular neural networks. We show in Table 1 some of these trainings with different numbers of

Table 1. Sample Trainings of the Modular Neural Network



Red	Funcion de Entrenamiento	Num. de Capas	Neuronas	F. R.	% de Reconocimiento	Error Meta	Error Alcanzado	Epocas	Time
1	Trainscg	V: Mod1: 2	48,49	sse	100% (20/20)	0.001	0.000998658	3000	35 Min.
		Mod2: 2	50,60	sse	100% (20/20)	0.001	0.000998254	3000	
		Mod3: 2	60,70	sse	100% (20/20)	0.001	0.00099876	3000	
		R: Mod4: 1	350	mse	100% (20/20)	0.01	0.0099726	4000	
		Mod5: 1	400	mse	100%(20/20)	0.01	0.0099546	4000	
		Mod6: 1	420	mse	100% (20/20)	0.01	0.0098316	4000	
		H: Mod7: 1	350	mse	100% (20/20)	0.01	0.0080223	4000	
		Mod8: 1	250	mse	100% (20/20)	0.01	0.0086453	4000	
		Mod9: 1	300	mse	100% (20/20)	0.01	0.0067892	4000	

Red	Funcion de Entrenamiento	Num. de Capas	# de Neuronas	F.R.	% de Reconocimiento	Error Meta	Error Alcanzado	Epocas	Time
10	Trainscg	V: Mod1: 2	80,90	sse	100% (20/20)	0.001	0.000999948	3000	1 Hra. 34 Min.
		Mod2: 2	90,90	sse	100% (20/20)	0.001	0.000992595	3000	
		Mod3: 2	80,90	sse	100% (20/20)	0.001	0.000997131	3000	
	Trainscg	R: Mod4: 1	20	mse	100% (20/20)	0.01	0.124015	4000	
		Mod5: 1	15	mse	100% (20/20)	0.01	0.0269689	4000	
		Mod6: 1	25	mse	100% (20/20)	0.01	0.01698	4000	
	Traingdx	H: Mod7: 1	350	mse	25% (5/20)	0.01	0.037501	4000	
		Mod8: 1	230	mse	5% (1/20)	0.01	0.0450007	4000	
		Mod9: 1	290	mse	5% (1/20)	0.01	0.0425011	4000	

Red	Funcion de Entrenamiento	Num. de Capas	# de Neuronas	F.R.	% de Reconocimiento	Error Meta	Error Alcanzado	Epocas	Time
14	Trainscg	V: Mod1: 2	85,95	mse	100% (30/30)	0.001	0.000990206	3000	1 Hra. 23 Min.
		Mod2: 2	95,90	mse	100% (30/30)	0.001	0.000970259	3000	
		Mod3: 2	99,96	mse	93% (28/30)	0.001	0.000995249	3000	
	Trainscg	R: Mod4: 1	25	mse	0.3% (1/30)	0.01	0.0880936	4000	
		Mod5: 1	20	mse	0.3% (1/30)	0.01	0.0172602	4000	
		Mod6: 1	30	mse	0.3% (1/30)	0.01	0.0127059	4000	
	Traincsg	H: Mod7: 1	15	mse	96% (29/30)	0.01	0.148242	4000	
		Mod8: 1	10	mse	90% (27/30)	0.01	0.142394	4000	
		Mod9: 1	23	mse	96% (29/30)	0.01	0.127423	4000	

modules, layers and nodes. The training times are also shown in this table to illustrate the performance with different training algorithms and conditions.

Once the necessary trainings were done, a set of tests were performed with different type-2 fuzzy systems. The fuzzy systems were used as response integrators for the three modules of the modular network. In the type-2 fuzzy systems, different types of membership finctions were considered with goal of comparing the results and deice on the best choice for the recognition problem.

The best type-2 fuzzy system, in the sense that it produced the best recognition results, was the one with triangular membership functions. This fuzzy system has 3 input variables and one output variable, with three membership functions per variable.We show in Figures 11 and 12 the membership functions of the type-2 fuzzy sytem.

The recognition results of this type-2 fuzzy system, for each training of the modular neural, are shown in Table 2.

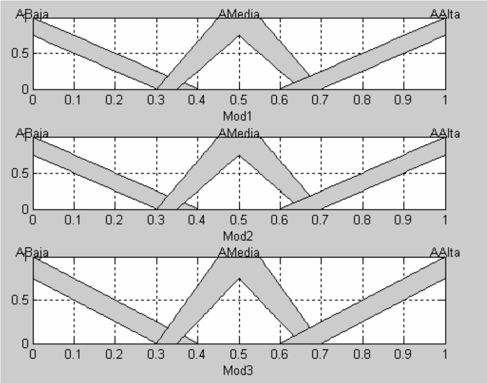


Fig. 5. Input variables of the type-2 fuzzy system

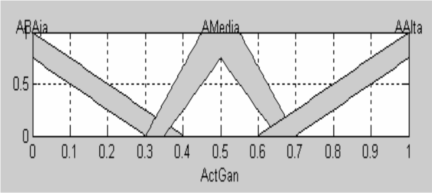
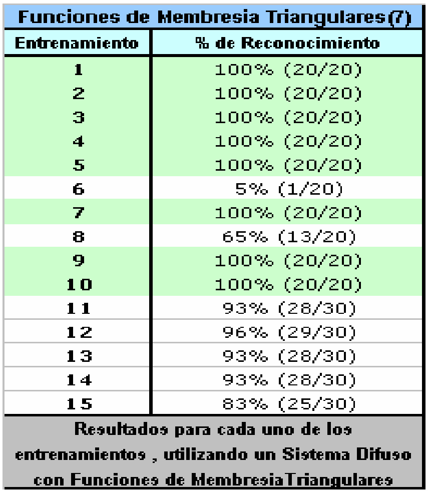


Fig. 6. Output variables of the type-2 fuzzy system

Table 2. Results of the Type-2 Fuzzy System with Triangular Membership Functions



Funciones de Membresia Triangulares(7)	
Entrenamiento	**% de Reconocimiento**
1	100% (20/20)
2	100% (20/20)
3	100% (20/20)
4	100% (20/20)
5	100% (20/20)
6	5% (1/20)
7	100% (20/20)
8	65% (13/20)
9	100% (20/20)
10	100% (20/20)
11	93% (28/30)
12	96% (29/30)
13	93% (28/30)
14	93% (28/30)
15	83% (25/30)
Resultados para cada uno de los entrenamientos , utilizando un Sistema Difuso con Funciones de MembresiaTriangulares	

In Table 2 we show the results for 15 trainings of the modular neural network. In each row of this table we can appreciate the recognition rate with the type-2 fuzzy sytem. We can appreciate that in 8 out of 15 cases, a 100% recognition rate was achieved.

The fuzzy systems with worst results for the modular neural network were the ones with Gaussian and Trapezoidal membership functions. We use 3 input variables and one ouput vraiable, as in the previous fuzzy system. We show in Figures 7 and 8 the Gaussian membership functions of this system.

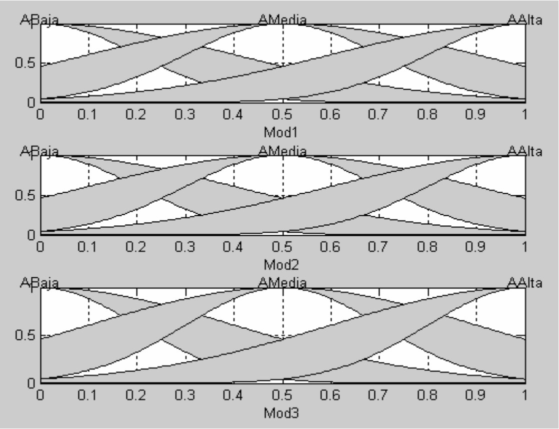


Fig. 7. Input variables for type-2 fuzzy system with Gaussian membership functions

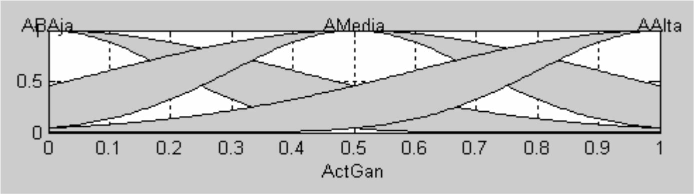


Fig. 8. Output variable for type-2 fuzzy system with Gaussian membership functions

We show in Figures 9 and 10 the Trapezoidal membership functions of another type-2 fuzzy system.

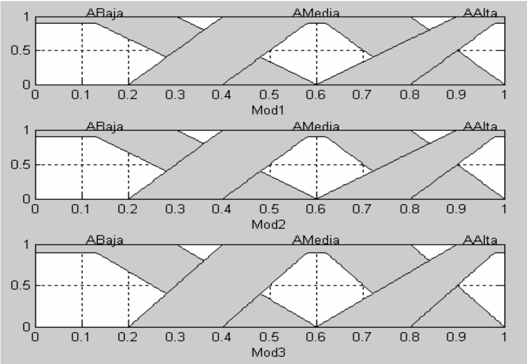


Fig. 9. Input variables for the Type-2 Fuzzy System with Trapezoidal Functions

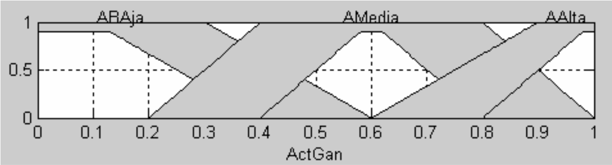
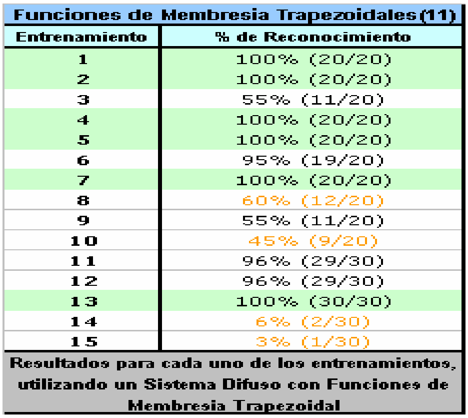


Fig. 10. Output variable for type-2 fuzzy system with Trapezoidal functions

The results that were obtained with Gaussian and Trapezoidal membership functions are similar. We show in Table 3 the recognition results obtained with the type-2 fuzzy system with Trapezoidal membership functions. We can appreciate from Table 3 that only in 6 out of the 15 cases a 100% recognition rate is obtained. Also, there are 4 cases with low recognition rates.

Table 3. Recognition rates with the Type-2 System and Trapezoidal Functions



Funciones de Membresia Trapezoidales(11)

Entrenamiento	% de Reconocimiento
1	100% (20/20)
2	100% (20/20)
3	55% (11/20)
4	100% (20/20)
5	100% (20/20)
6	95% (19/20)
7	100% (20/20)
8	60% (12/20)
9	55% (11/20)
10	45% (9/20)
11	96% (29/30)
12	96% (29/30)
13	100% (30/30)
14	6% (2/30)
15	3% (1/30)

Resultados para cada uno de los entrenamientos, utilizando un Sistema Difuso con Funciones de Membresia Trapezoidal

We have to mention that results with a type-1 fuzzy integration of responses were performed in previous paper, in which the recognition rates were consistently lower by an average of 5%. We can state in conclusion that the type-2 fuzzy system for response integration is improving the recognition rate in the case of persons based on face, fingerprint and voice.

7 Conclusions

We described in this paper a new method for response integration in modular neural networks that uses type-2 fuzzy logic to model uncertainty in the decision process. We showed different trainings of the modular neural networks, and tested different type-2 fuzzy systems for response integration. Based on the obtained recognition rates, the best results were achieved with a type-2 fuzzy system with triangular membership functions. The results obtained with this type-2 fuzzy system are better than the previously obtained by a similar type-1 approach [11].

Acknowledgements

The authors would like to thank CONACYT and DGEST for the financial support given to this research project. The students (Jerica Urias and Denisse Hidalgo) were supported by a scholarship from CONACYT.

References

[1] O. Castillo and P. Melin,"Hybrid Intelligent Systems for Time Series Prediction using Neural Networks, Fuzzy Logic and Fractal Theory", IEEE Transactions on Neural Networks, Vol. 13, no. 6, pp. 1395-1408, 2002.

[2] O. Castillo, L. Aguilar, N. Cazarez, D. Rico, "Intelligent Control of Dynamical Systems with Type-2 Fuzzy Logic and stability Study", Proceedings of the Conference on Artificial Intelligence (IC' AI' 2005), pp. 400-405, June 2005.

[3] F. Fogelman-Soulie, "Multi-modular neural network-hybrid architectures: a review", Proceedings of 1993 International Joint Conference on Neural Networks, 1993.

[4] B. Happel and J. Murre, "Design and evolution of modular neural network architectures", Neural Networks, vol. 7, pp. 985-1004, 1994.

[5] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts", Neural Computation, vol. 3, pp. 79-87, 1991.

[6] R. Jenkins and B. Yuhas, "A simplified neural network solution through problem decomposition: The case of the truck backer-upper", IEEE Transactions on Neural Networks, vol. 4, no. 4, pp. 718-722, 1993.

[7] M. I. Jordan and R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm", Neural Computation, vol. 6, pp. 181-214, 1994.

[8] N. N. Karnik, J. M. Mendel, "Operations on type-2 fuzzy sets", Signal and Image Processing Institute, Department of Electrical Engineering-Systems, University of Southern California, May 2000.

[9] N. N. Karnik, J. M. Mendel, *IEEE*, y Qilian Liang, "Type-2 Fuzzy Logic Systems", IEEE Transactions on Fuzzy Systems, Vol. 7, No. 6, December 1999.

[10] P. Melin and O. Castillo, "Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing", Springer, 2005.

[11] J. M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems, Introduction and New Directions", Prentice Hall, 2001.

[12] J. M. Mendel and R. I. John, "Type-2 Fuzzy Sets Made Simple", IEEE Transactions on Fuzzy Systems, vol. 10, No. 2, April 2002.

[13] C. Monrocq, "A probabilistic approach which provides and adaptive neural network architecture for discrimination", Proceedings of the International Conference on Artificial Neural Networks, vol. 372, pp. 252-256, 1993.

[14] J. Urias, D. Solano, M. Soto, M. Lopez, and P. Melin, "Type-2 Fuzzy Logic as a Method of Response Integration in Modular Neural Networks", Proceedings of the 2006 International Conference on Artificial Intelligence, vol. 2 pp. 584.

[15] Zadeh, L.A.,"Fuzzy Sets", Information and Control, vol. 8, 1975.

[16] Zadeh L.A., "Fuzzy Logic = Computing with Words", IEEE Transactions on Fuzzy Systems, vol. 4, No. 2, May 1996.

[17] Zadeh, L.A., "Fuzzy Logic", Computer, vol. 1, No. 4, pp. 83-93, 1998.

Evolving Type-2 Fuzzy Logic Controllers for Autonomous Mobile Robots

Christian Wagner and Hani Hagras

Department of Computer Science, University of Essex, Wivenhoe Park,
Colchester, United Kingdom

Abstract. Autonomous mobile robots navigating in changing and dynamic unstructured environments like the outdoor environments need to cope with large amounts of uncertainties that are inherent in natural environments. The traditional type-1 Fuzzy Logic Controller (FLC) using precise type-1 fuzzy sets cannot fully handle such uncertainties. A type-2 FLC using type-2 fuzzy sets can handle such uncertainties to produce a better performance. However, manually designing the type-2 Membership Functions (MFs) for an interval type-2 FLC is a difficult task. This paper will present a Genetic Algorithm (GA) based architecture to evolve the type-2 MFs of interval type-2 FLCs used for mobile robots. The GA based system converges after a small number of iterations to type-2 MFs which gave very good performance. We have performed various experiments in which the evolved type-2 FLC dealt with the uncertainties and resulted in a very good performance that has outperformed its type-1 counterpart as well as the manually designed type-2 FLC.

Keywords: Type-2 Fuzzy Logic Control, Autonomous Mobile Robots, Genetic Algorithms.

1 Introduction

Autonomous mobile robots navigating in real-world unstructured environments (i.e. environments that have not been specifically engineered for the robot) must be able to operate under the conditions of imprecision and uncertainty present in such environments. Thus the choice of adequate methods to model and handle such uncertainties is crucial for mobile robot controllers.

The Fuzzy Logic Controller (FLC) is credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance in applications where the inherent uncertainty makes it difficult to achieve good results using traditional methods [14]. As a result the FLC has become a popular approach to mobile robot control in recent years [18], [19]. There are many sources of uncertainty facing the FLC for a mobile robot navigating in changing and dynamic unstructured environments; we list some of them as follows:

- Uncertainties in inputs to the FLC which translate to uncertainties in the antecedent Membership Functions (MFs) as the sensor measurements are typically noisy and are affected by the conditions of observation (i.e. their characteristics are changed by the environmental conditions such as wind, sunshine, humidity, rain, etc.).

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 16–25, 2007.
springerlink.com

- Uncertainties in control outputs which translate to uncertainties in the consequent MFs of the FLC. Such uncertainties can result from the change of the actuators characteristics which can be due to wear, tear, environmental changes, etc.
- Linguistic uncertainties as the meaning of words that are used in the antecedent and consequent linguistic labels can be uncertain - words mean different things to different people [16], [17]. In addition, experts do not always agree and they often provide different consequents for the same antecedents. A survey of experts will usually lead to a histogram of possibilities for the consequent of a rule; this histogram represents the uncertainty about the consequent of a rule [15].
- Uncertainties associated with the use of noisy training data that could be used to learn, tune, or optimise the FLC.

While traditionally, type 1 FLCs have been employed widely in robot control, it has become apparent in recent years that the type-1 FLC cannot fully handle high levels of uncertainties as its MFs are in fact completely crisp [10], [16]. The linguistic and numerical uncertainties associated with dynamic unstructured environments cause problems in determining the exact and precise MFs during the robot FLC design. Moreover, the designed type-1 fuzzy sets can be sub-optimal under specific environmental and operational conditions. The environmental changes and the associated uncertainties might require the continuous tuning of the type-1 MFs as otherwise the type-1 FLC performance might deteriorate [5]. As a consequence, research has started to focus on the possibilities of higher order FLCs, such as type-2 FLCs that use type-2 fuzzy sets.

A type-2 fuzzy set is characterised by a fuzzy MF, i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1] [15]. The MF of a type-2 fuzzy set is three dimensional and includes a footprint of uncertainty. It is the third-dimension of the type-2 fuzzy sets and the footprint of uncertainty that provide additional degrees of freedom making it possible to better model and handle uncertainties when compared to type-1 fuzzy sets. It has been shown that interval type-2 FLCs (that use interval type-2 fuzzy sets) can handle the uncertainties and outperform their type-1 counterparts in applications with high uncertainty levels such as mobile robot control [4], [5]. However, manually designing and tuning a type-2 FLC to give a good response is a difficult task, particularly as the number of MF parameters increases.

In the type-1 FLC domain, several researchers have explored the possibilities of using different learning and evolutionary techniques to specify FLC parameters [1], [2], [6], [7], [8]. However the number of parameters and the complexity increase when dealing with type-2 FLCs. It has been shown that the selection of an optimal footprint of uncertainty for the individual type-2 MFs is highly complex and thus an automatic optimisation process is highly desirable [12], [15].

Recent work had proposed the use of neural based systems to learn the type-2 FLC parameters [12], [13], [20]. However, these approaches require existing data to optimise the type-2 FLC. Thus, they are not suitable for applications where there is no or not sufficient data available to represent the various situations faced by the controller (as in the mobile robots domain).

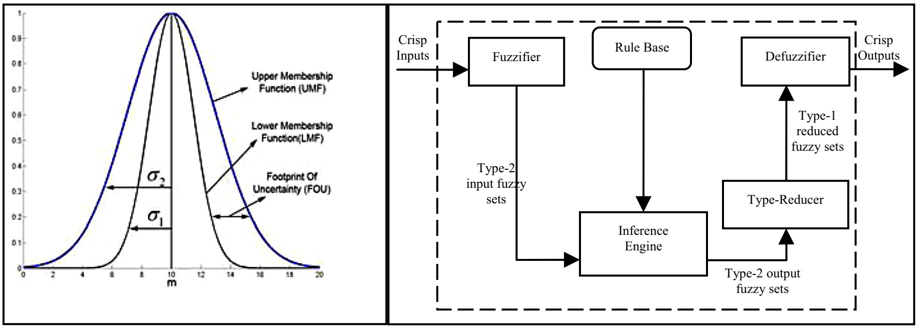
Genetic Algorithms (GAs) do not require a priori knowledge such as a model or data but perform a search through the solution space based on natural selection, using a specified fitness function. GAs have been used for type-2 FLCs where in [22], two GAs were used first to optimise a type-1 FLC, then to blur the MFs of this type-1 FLC to create a footprint of uncertainty for the type-2 FLC. In this paper, we will take a different approach by directly optimising the type-2 MFs rather than developing the type-1 MFs and then blurring them. We did not evolve the type-2 FLC rule base as it will remain the same as the type-1 FLC rule base [15]. However, the FLC antecedents and consequents will be represented by interval type-2 MFs rather than type-1 MFs.

The next section will present the interval type-2 FLC and we will highlight its benefits. Section 3 will detail the GA based evolutionary system. Section 4 will present the experiments and results. Finally, the conclusions will be presented in Section 5.

2 Interval Type-2 Fuzzy Logic Controllers

The interval type-2 FLC uses interval type-2 fuzzy sets (such as those shown in Fig. 1(a) to represent the inputs and/or outputs of the FLC. In the interval type-2 fuzzy sets all the third dimension values equal to one. The use of interval type-2 FLC helps to simplify the computation (as opposed to the general type-2 FLC which is computationally intensive) which will enable the design of a robot FLC that operates in real time [5].

The structure of an interval type-2 FLC is depicted in Fig. 1(b), it consists of a Fuzzifier, Inference Engine, Rule Base, Type-Reducer and a Defuzzifier.



(a) (b)

Fig. 1. (a) An interval type-2 fuzzy set. (b) Structure of the type-2 FLC.

The interval type-2 FLC works as follows: the crisp inputs from the input sensors are first fuzzified into input type-2 fuzzy sets; singleton fuzzification is usually used in interval type-2 FLC applications due to its simplicity and suitability for embedded processors and real time applications. The input type-2 fuzzy sets then activate the inference engine and the rule base to produce output type-2 fuzzy sets. The type-2 FLC rules will remain the same as in a type-1 FLC but the antecedents and/or the consequents will be represented by interval type-2 fuzzy sets. The inference engine combines the fired rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. The type-2 fuzzy outputs of the inference engine are then processed by the type-reducer which combines the output sets and performs a centroid calculation which leads to type-1 fuzzy sets called the type-reduced sets. There are different types of type-reduction methods. In this paper we will be using the Center of Sets type-reduction as it has reasonable computational complexity that lies between the computationally expensive centroid type-reduction and the simple height and modified height type-reductions which have problems when only one rule fires [15]. After the type-reduction process, the type-reduced sets are defuzzified (by taking the average of the type-reduced set) to obtain crisp outputs that are sent to the actuators. More information about the interval type-2 FLC can be found in [5], [15].

It has been argued that using interval type-2 fuzzy sets to represent the inputs and/or outputs of FLCs has many advantages when compared to type-1 fuzzy sets; we summarise some of these advantages as follows:

- As the type-2 fuzzy set membership functions are themselves fuzzy and contain a footprint of uncertainty, they can model and handle the linguistic and numerical uncertainties associated with the inputs and outputs of the FLC. Therefore, FLCs that are based on type-2 fuzzy sets will have the potential to produce a better performance than type-1 FLCs when dealing with uncertainties [5].
- Using type-2 fuzzy sets to represent the FLC inputs and outputs will result in the reduction of the FLC rule base when compared to using type-1 fuzzy sets as the uncertainty represented in the footprint of uncertainty in type-2 fuzzy sets lets us cover the same range as type-1 fuzzy sets with a smaller number of labels. The rule reduction will be greater as the number of the FLC inputs increases [15].
- Each input and output will be represented by a large number of type-1 fuzzy sets which are embedded in the type-2 fuzzy sets [15], [16]. The use of such a large number of type-1 fuzzy sets to describe the input and output variables allows for a detailed description of the analytical control surface as the addition of the extra levels of classification gives a much smoother control surface and response. According to Karnik and Mendel [9], the type-2 FLC can be thought of as a collection of many different embedded type-1 FLCs.
- It has been shown in [23] that the extra degrees of freedom provided by the footprint of uncertainty enables a type-2 FLC to produce outputs that cannot be achieved by type-1 FLCs with the same number of membership functions. It has also been shown that a type-2 fuzzy set may give rise to an equivalent type-1 membership grade that is negative or larger than unity. Thus a type-2 FLC is able to model more complex input-output relationships than its type-1 counterpart and thus can give a better control response.

3 The GA Based Evolutionary System

The GA chromosome includes the interval type-2 MFs parameters for both the inputs and outputs of the robot type-2 FLC. We have used the interval type-2 fuzzy set which is described by a Gaussian primary MF with uncertain standard deviation as shown in Fig. 1(a).

The GA based system uses real value encoding to encode each gene in the chromosome. Each GA population consists of 30 chromosomes. The GA uses an elitist selection strategy. The GA based system procedure can be summarised as follows:

Step 1: 30 chromosomes are generated randomly while taking into account the grammatical correctness of the chromosome (for example the inner standard deviation (σ_1) is less than the outer standard deviation (σ_2)). The "Chromosome Counter" is set to 1 – the first chromosome. The "Generation Counter" is set to 1 – the first generation.

Step 2: Type-2 FLC is constructed using the chromosome "Chromosome Counter" and is executed on the robot for 400 control steps to provide a fitness for the chromosome. During fitness evaluation, a safety behaviour is in place to avoid the robot colliding with the wall. A controller that would have caused a collision is automatically assigned a disastrous fitness, making sure it is excluded through the selection process. After a controller has been executed for 400 control steps, a fixed controller takes over control and returns the robot to a correct position in respect to the wall to enable the test of the next controller.

Step 3: If "Chromosome Counter" < 30, increment "Chromosome Counter" by 1 and go to Step 2, otherwise proceed to Step 4.

Step 4: The best individual-so-far chromosome is preserved separately.

Step 5: If "Generation Counter" = 1 then store current population, copy it to a new population *P* and proceed to Step 6. Else, select 30 best chromosomes from population "Generation Counter" and population "Generation Counter"-1 and create a new population *P*.

Step 6: Use roulette wheel selection on population *P* to populate the breeding pool.

Step 7: Crossover is applied to chromosomes in the breeding pool and "chromosome consistency" is checked. (*)

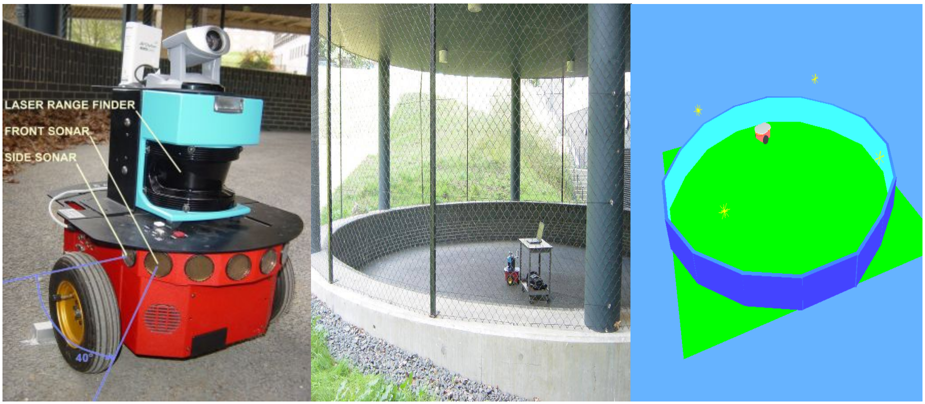
Step 8: "Generation Counter" is incremented. If "Generation Counter" < the number of maximum generations or if the desired performance is not achieved, reset "Chromosome Counter" to 1 and go to Step 2, else go to Step 9.

Step 9: Chromosome with best fitness is kept and solution has been achieved; END.

(*)The crossover operator employed computes the arithmetic average between two genes [3]. It is used with a probability of 100% to force the GA to explore the solution space in between the previously discovered, parental solutions [3]. With chromosome consistency we refer to the correctness of the chromosome's genes in relation to their function in the FLC, (for example the inner standard deviation (σ_1) is less than the outer standard deviation (σ_2)). While in [22], the chromosome's genes are re-arranged to achieve this, we eliminate a chromosome from the population if it violates this criteria. The violating chromosome is replaced by a new, randomised but consistent chromosome thus introducing new genetic material into the population at the same time.

4 Experiments and Results

We have performed many experiments to evaluate the proposed system, however, due to the space limitations, we are going to present representative experiments involving the edge following behaviour.



(a) (b) (c)

Fig. 2. (a) The robot, (b) The arena, (c) The simulated environment

We have used the Pioneer 2-wheeled robot shown in Fig. 2(a) which is equipped with 8 sonar sensors, a camera and a laser scanner. The edge following behaviour experiments used only the two side sonar sensors. The laser range finder was used to give an accurate reading of the robots actual distance from the wall. All the computation was done using the on-board computer running Linux and Java version 1.5 while the results were logged on a server using a wireless link.

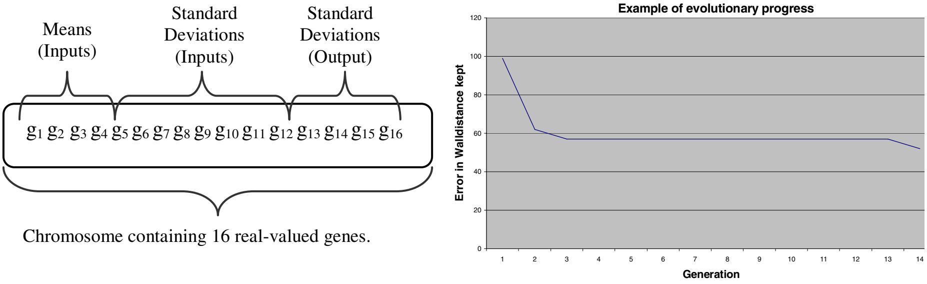
The robot operated in a real-world, outdoor, unstructured environment which consisted of an outside arena of circular shape as shown in Fig. 2(b). The floor of the arena is standard road tarmac, providing a relatively smooth, but uneven surface, while the surrounding wall is constructed of dark grey bricks (which due to their colour as well as their smooth and slightly glossy texture provided a challenge to both the robots sonar and laser sensors). This outdoor environment provided various sources of uncertainty not present in indoor environments, ranging from wind and significant differences in humidity to small debris such as leaves and small stones.

The robot simulation uses the Webots simulator version 5.1.7. The simulated environment consists of a circular arena modelled using VRML, shown in Fig. 2(c). The wall is modelled using 16 flat panels, thus creating a roughly circular shape which tries to simulate the uncertainty levels present in the real environment.

In these experiments, the type-2 FLC has two sonar sensors as inputs which are represented by two Gaussian type-2 MFs (Near and Far) which have certain means and uncertain standard deviations. Therefore each input type-2 MF is represented by three parameters (one certain mean and two standard deviations). Thus, 12 genes are used to represent the type-2 FLC inputs (The FLC has 2 input sensors, each represented by two type-2 MFs and each MF is represented by 3 parameters).

The type-2 FLC has 1 output governing the speed difference between the left and right wheel of the robot and thus, the turning angle. The FLC output is represented by 2 Gaussian type-2 MFs which have certain means and uncertain standard deviations. Only the standard deviations of the output type-2 MFs are evolved and the means are fixed to guarantee that the FLC outputs are within the allowed domain of the outputs.

Hence as shown in Fig. 3(a), the GA chromosome for this type-2 FLC comprises 12(inputs) + 4(outputs) = 16 genes.


Means
(Inputs)
Standard
Deviations
(Inputs)
Standard
Deviations
(Output)
g1 g2 g3 g4 g5 g6 g7 g8 g9 g10 g11 g12 g13 g14 g15 g16
Chromosome containing 16 real-valued genes.
Example of evolutionary progress
Error in Walldistance kept
0
20
40
60
80
100
120
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Generation

(a) (b)

Fig. 3. (a) Chromosome Structure, (b) Progress of best individual

The fitness of each chromosome is generated by monitoring how the generated type-2 FLC has succeeded in following the edge at the desired distance over 400 control steps.

During this fitness evaluation period, a safety behaviour is in place that keeps the robot from colliding with the wall. A controller that would have resulted in a collision is assigned a disastrous fitness and as such eliminated through the selection process.

After a controller has been executed for 400 control steps, a fixed controller takes over control and returns the robot to a correct position in respect to the wall to enable the test of the next controller.

Through various experiments, it was found that the GA based system evolves to good type-2 MFs after about only 14 generations. An example of the evolutionary progress is shown in Fig. 3(b) which shows the performance of the best individual found so far against the number of generations.

We have compared the performance of the evolved type-2 FLC against a manually designed type-2 FLC as well as a series of three type-1 FLCs with 4, 9 and 25 rules. For each of these FLCs a visualisation of their respective control surface was computed and is displayed in Fig. 4.

Each of the controllers was tested in the simulated arena to determine its actual performance. The robot paths resulting from the controllers employing 4 rules are shown in Fig. 5, together with the Root Mean Squared Error (RMSE) which was computed by sampling the error between the desired distance to the wall and the actual distance to the wall at every control step over the entire run.

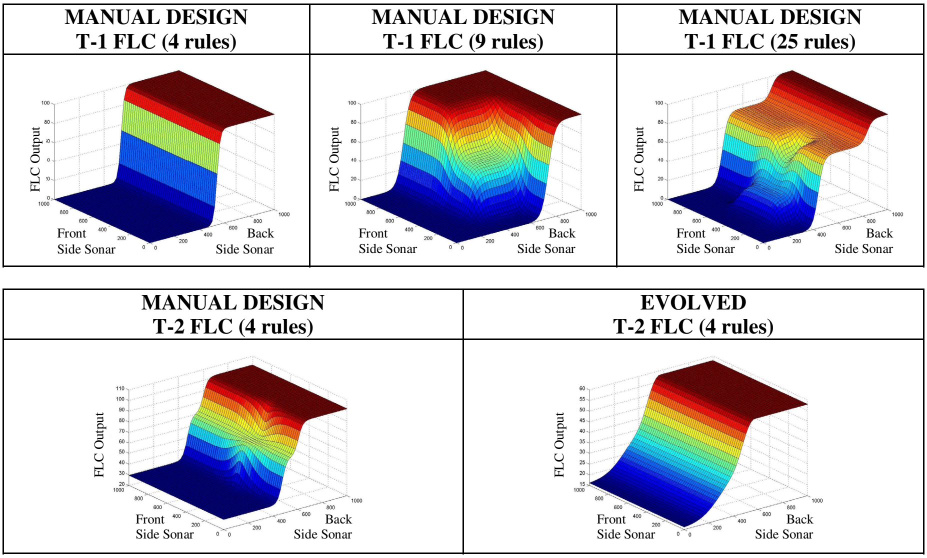


Fig. 4. Control Surfaces for all FLCs

The control surfaces in Fig. 4 show very crude control for the Type-1 FLC with 4 rules which leads to expectedly unsmooth motion control as shown in Fig. 5. As the number of rules increases to 9 and then 25 rules, the control surface becomes more and more detailed which is reflected in improved motion control resulting in a RMSE of 54.534 for the 9-rule and 32.319 for the 25-rule type-1 FLC. (The paths for the 9 and 25 rule type-1 FLC were not included due to space restrictions.)

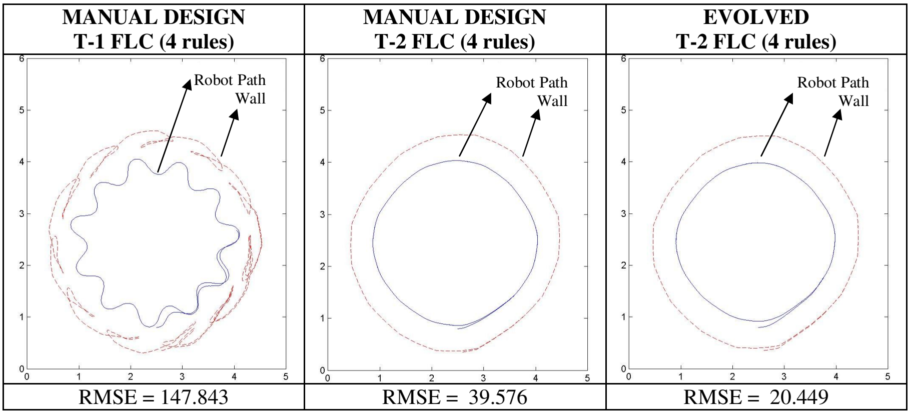


Fig. 5. Robot Paths and RMSEs for FLCs using 4 rules

The control surface for the manually designed type-2 FLC with 4 rules already looks much more smoother than its type-1 counterpart, showing the potential of type-2 FLCs to outperform type-1 FLCs with the same number of rules. The resulting motion control produced a very good RMSE of 39.576.

The evolved type-2 FLCs control surface finally shows a very smooth, control surface which results in a RMSE of 20.449 as can be seen in Fig. 5 which outperforms the manually designed type-2 FLC and all three type-1 FLCs. As can be seen in Fig.5, the more type-1 fuzzy sets are used in the type-1 FLC, the more its response approaches the smooth response of the type-2 FLC. This is because the type-2 fuzzy sets contain a large number of embedded type-1 fuzzy sets which allow for the detailed description of the analytical control surface as the addition of the extra levels of classification gives a much smoother control surface and response.

5 Conclusions

In this paper, we have presented the use of a GA based architecture to directly evolve the type-2 MFs of interval type-2 FLCs used for mobile robot control.

We have shown that the genetically evolved type-2 FLCs can lead to superior performance in comparison to type-1 FLCs and the manually designed type-2 FLCs. The results indicate that the genetic evolution of type-2 MFs can provide valid and high-performance type-2 FLCs without relying on any a priori knowledge such as logged data or a previously existing model, making it suitable for control problems where no such a priori data is available such as in the mobile robots domain.

This work will be a step towards overcoming the problem of manually specifying pseudo-optimal MFs for type-2 FLCs, which to date is one of the main obstacles when designing type-2 FLCs.

Our current and future work will focus on further analysis and development of the GA system with a view to adapt it to evolving general type-2 Fuzzy Controller parameters.

Furthermore, one of our main aims is to evolve the controllers online and in the real world, allowing for a wide range of practical applications.

References

1. E. Alba, C. Cotta and J. M. Troya; “Evolutionary Design of Fuzzy Logic Controllers Using Strongly-Typed GP”; Mathware & Soft Computing, 6(1), pp. 109-124, 1999.
2. O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena; Genetic Fuzzy System: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases; Singapore: World Scientific; 2001
3. L. Davis; Ed., Handbook of Genetic Algorithms; New York: Van Nostrand, Reinhold; 1990
4. J. Figueroa, J. Posada , J. Soriano, M. Melgarejo and S. Roj; A type-2 fuzzy logic controller for tracking mobile objects in the context of robotic soccer games; Proceeding of the 2005 IEEE International Conference on Fuzzy Systems, pp. 359-364, Reno, USA; May 2005.

5. H. Hagras; "A Hierarchical type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots"; IEEE Transactions On Fuzzy Systems, Vol. 12 No. 4, pp. 524-539, August 2004.
6. H. Hagras, M. Colley, V. Callaghan; " Learning and Adaptation of an Intelligent Mobile Robot Navigator Operating in Unstructured Environment Based on a Novel Online Fuzzy-Genetic System"; Journal of Fuzzy Sets and Systems, Volume 141, Number 1, pp. 107-160, January 2004
7. F. Hoffmann and G. Pfister; " Evolutionary Design of a fuzzy knowledge base for a mobile robot"; International Journal of Approximate Reasoning, Vol. 17, No. 4, pp. 447-469, 1997
8. A. Homaifar and E. McCormick; Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms; IEEE Trans. Fuzzy Systems, Vol. 3, No. 2; May 1995
9. N. Karnik and J. Mendel; An introduction to type-2 fuzzy logic systems; USC report, http://sipi.usc.edu/~mendel/report; October 1998.
10. G. Klir and T. Fogler; "Fuzzy Sets, Uncertainty and Information."; Prentice Hall, 1988
11. Q. Liang and J. Mendel; Interval type-2 Fuzzy Logic Systems: Theory and Design; IEEE Transactions on Fuzzy Systems, Vol. 8, No. 5, October 2000
12. C. Lynch, H. Hagras, and V. Callaghan; Using Uncertainty Bounds in the Design of an Embedded Real-Time type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines; Proceedings of the 2006 IEEE International Conference of Fuzzy Systems, Vancouver, Canada; July 2006
13. C. Lynch. H. Hagras and V. Callaghan; Embedded Interval type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines; Proceedings of the 2006 Information Processing and Management of Uncertainty in Knowledge-based Systems conference, pp. 1340-1347, Paris, France; July 2006.
14. E. Mamdani; Fuzzy control - a misconception of theory and application; IEEE Expert, Vol. 9, No. 4, pp. 27-28; August 1994.
15. J. Mendel; "Uncertain Rule-Based Fuzzy Logic Systems"; Prentice Hall 2001
16. J. Mendel and R. John; "Type-2 Fuzzy Sets Made Simple" ; IEEE Transactions on Fuzzy Systems, Vol. 10, pp.117-127, April 2002.
17. J. Mendel and H. Wu; "Uncertainty versus choice in rule-based fuzzy logic systems"; Proceedings of IEEE International Conference of Fuzzy Systems, pp.1336-1342, Honolulu, USA, 2002
18. A. Saffiotti; "The uses of fuzzy logic in autonomous robot navigation"; Journal of Soft Computing, Vol. 1, No. 4, pp.180-197,1997.
19. A. Saffiotti, E. H. Ruspini and K. Konolige; International Handbook of Fuzzy Sets and Possibility Theory, Vol. 5; 1997
20. T. W. Wan, D. H. Kamal; On-line Learning Rules For type-2 Fuzzy Controller; Proceedings of the 2006 IEEE International Conference of Fuzzy Systems, Vancouver, Canada; July 2006
21. H. Wu and J. Mendel; Uncertainty Bounds and Their Use in the Design of Interval type-2 Fuzzy Logic Systems; IEEE Transactions on Fuzzy Systems, Vol. 10, No. 5, October 2002
22. D. Wu and W.W. Tan; A type-2 Fuzzy logic Controller for the Liquid-level Process; Fuzz-IEEE 2004, IEEE International Conference on Fuzzy Systems, Budapest; July 2004
23. D. Wu and W.W. Tan; type-2 FLS modeling capability analysis.; Proceedings of the 2005 IEEE International Conference on Fuzzy Systems, pp. 242-247; Reno, USA; May 2005.

Adaptive Type-2 Fuzzy Logic for Intelligent Home Environment

Sunghoi Huh, Seung-Eun Yang, Kwang-Hyun Park, JunHyeong Do, Hyoyoung Jang, and Z. Zenn Bien

Human-friendly Welfare Robot System-Engineering Research Center, KAIST, 373-1
Guseong-Dong, Yusong-Gu, Daejun, 305-701, Korea
shhuh@ctrsys.kaist.ac.kr

Abstract. In this paper, an intelligent home environment with an adaptive type-2 fuzzy logic (AT2FL) method is developed. The considered system is mainly for elderly and disabled persons and catches the users' intention through recognizing their hand gestures instead of remote controller. However, during the recognition procedure, to distinguish between the command and ordinary gestures needs a very rigorous procedure, and to cope with that, the AT2FL approach is proposed in this paper. The presented algorithm is applied to control home appliances, and some experimental results will be shown to verify the feasibility and effectiveness.

Keywords: Adaptive type-2 fuzzy logic, human friendly robot, intelligent system, gesture recognition system.

1 Introduction

As the economic level of human society is getting better gradually, social welfare conditions for elderly and disabled persons have been noticed by degrees. Recently, more countries have put their eyes on the welfare issue [1], and with the help of technical developments, many kinds of convenient facilities with intelligent space have been being developed for evolving their living spaces[2-6]. Intelligent sweat home by our research center [3] is one of those efforts, and all of home appliances are controlled by users' hand gesture without remote controller, which is called as a soft remote control system (SMCS) in this paper. SMCS is an interface to control multiple home appliances based on various hand gestures. In our system, ten different hand gesture commands are used to control home appliances, and each command gesture is selected for the elderly and disabled user to use the system easily[2,3]. However, because those command gestures are primitive motion, command like primitive motion is also appeared in user's ordinary behavior. Therefore, the system may wrongly appreciate the user's ordinary behavior as command gesture. Recently, fuzzy logic systems (FLS) are widely used as a universal approximator in the area of nonlinear mapping and control problems [9-12], and among them, type-2 fuzzy logic system (T2FLS) can be noticed due to its drastic performance on handling of uncertainties [11-13]. T2FLS is an extensional concept of an ordinary FLS (henceforth to be called type-1 fuzzy set) and was introduced by Zadeh. [11] Even though T2FL shows a

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 26–35, 2007.
springerlink.com

computational burden compared to the type-1, its drastic performance for handling uncertain rule bases make it widely used[12]. However, it is not easy to determine exact fuzzy variables, and this error may deteriorate the system performance. To compensate for the lower efficiency, until now, many efforts for the adaptive FLS have been developed [13-15]. This paper deals with the SMCS that controls home appliances by recognizing the user's hand and pointing gesture. To remedy unknown uncertainties in the fuzzy rule bases, T2FLS is introduced and adaptive algorithm is derived to optimize the fuzzy variables. Detailed description for the intelligent space and control policies are described, and to show the effectiveness of the proposed system, some experimental results are to be presented.

2 Intelligent Sweat Home (ISH)

Intelligent sweat home (ISH) is a service robot-integrated intelligent house platform with sensor-based home network system, which enables to provide various kinds of daily living assistive tasks for the inhabitant (e.g. serving a meal/drink, assisting walk, delivering a newspaper, etc.). Particularly, the systems will be very helpful to assist such people like elderly persons or the handicapped who have serious problems in doing a certain work with their own efforts in daily life.

2.1 Intelligent Home Environment

Our work on intelligent home environment is distinguished in the sense of fully supporting indispensable daily tasks with various functionalities to comply with the needs of the users from the statistical and questionnaire survey [2,3]. Design philosophy is based on the idea that the technologies and solutions for such smart house should be human-friendly, i.e. smart houses should possess high level of intelligence in their control, actions and interactions with the users, offering them high level of comfort and functionality. The effect of smart house toward its inhabitants is strongly dependable on the list of the devices, automation and control system, which build the home environment and synchronize their operation. The way of interaction of the inhabitant with the home-installed devices is also a significant aspect of smart house design. As a solution, ISH consists of several assistive systems and human-friendly interfaces shown in Fig. 1. These assistive systems can provide lots of services for the inhabitant, however, its control interfaces also may cause inconvenience for the elderly and the physically handicapped. Especially, gesture command recognition is the most challenging and one of crucial challenging subjects in out ISH [2, 3].

2.2 Soft Remote Control System (SRCS)

Even though the remote controller has been normally used in our daily life, it may lead to a serious condition if it is not placed in a reachable area especially in case of the elderly and disabled persons. In accordance with the demand, many studies on human computer interaction (HCI) have been conducted with the development of high technologies. User friendly HCI systems enable them to operate home appliances and other devices easily. As a possible way to control home appliances, some researchers have been developed on voice command recognition system [2]. Voice recognition seems to be very useful approach for the disabled or elderly persons, however, there's a well known problem that the sensitive microphone has to be kept nearby the users.

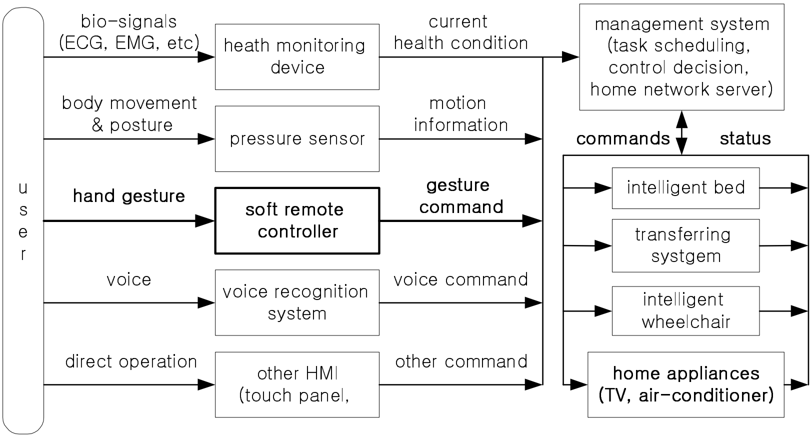


Fig. 1. Block diagram of ISH

Until now, some kinds of gesture recognizing system have been steadily researched for intelligent spaces [7, 8], however, most of them have mainly focused on the recognition of hand orientation and posture in restricted environments [7] to control a single system [8]. As described in the previous section, our soft remote control system (SRCS) recognizes hand position and pointing direction from the images of 3 color cameras equipped with pan/tilt devices and is capable to control multiple home appliances based on various hand gestures [2, 3]. The system has five modules as shown in Fig. 2: the data acquisition module, preprocessing module, hand region segmentation module, pointing recognition module and appliance control module.

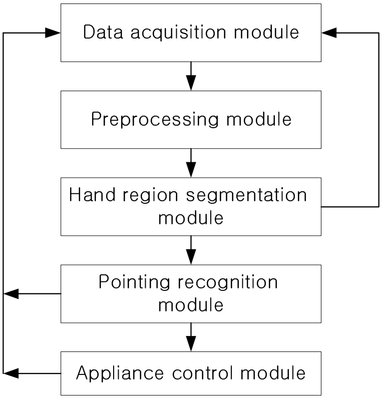


Fig. 2. Control flow diagram in SRCS

The system is capable to recognize primitive motions such as up, down, left, and right, and ten different hand gesture commands are used. Each command gesture is selected for the elderly and disabled user to operate the system easily. However, even for the same meaningful gesture, the characteristic of human action is various from person to person, and some common characteristics are found for the same meaningful gesture. As a result, during the recognition procedure, to distinguish between the command and ordinary gestures needs a very rigorous procedure, and as a result, the system normally recognize the user's ordinary behavior as command gesture, which causes wrong recognition problem.

3 Hand Gesture Recognition System

At first, the recognition system selects appropriate features from the gesture input, then, the selected features are applies to the gesture model. Based on the model output, the meaning of the gesture input is decided. Besides, by using reference data and current result, additional adaptive scheme can be decided.

3.1 Feature Selection

Some characteristic exist which commonly occurred in target gesture and has different characteristic from garbage gesture. Those characteristic may used as features to discriminate target gesture and similar gestures. Therefore, appropriate feature selection is the first step to implement the system. At first, as candidate command gesture, several motion characteristics can be selected, and in this paper, eleven candidate commands are taken [2]. And next, the feature data is to be selected to make the gesture characterized. However, too many features increase the complexity and execution time, hence, it needs to select appropriate number of feature which is essential to spot meaningful gesture. In this research, rough set theory with genetic algorithm is applied [2].

3.2 Gesture Recognition System: Conventional Fuzzy Logic Approach

Human movement from ordinary action is unstructured and unconsciously occurred, and even though variance exists in human movement, meaningful evidence and rules are to be found. This makes fuzzy logic appropriate method for this type of problem, and acquired knowledge through observation is easily applied to the system. Further more, the sensitivity to variation of parameter or environment can be reduced with the help of using the fuzzified value instead of crisp value [10]. Finally, usage of linguistic value makes fuzzy logic superior to other methods because the ambiguous characteristic of unconscious human movement can be expressed in this way. The aimed user in this research is the elderly and disabled people who have constraints in mobility and capability to move other objects. Therefore, only movement information of user's hand and face is used to spot meaningful gesture in this research. However, it's not easy to define a finite and united fuzzy rule and implement a recognition system using every feature due to the big of feature size and noisy data. Finally, any conventional fuzzy logic system using one output threshold value is not appropriate to recognize meaningful gesture from similar ones because the fuzzy output value used to be very similar from each different data [2].

4 Hand Gesture Recognition with a Type-2 Fuzzy Logic (T2FL)

As described in the previous section, to cope with the challenges with the conventional algorithm, much efficient approach is required, which has the robustness to noisy input data and adaptive characteristics to handle the various threshold conditions. In this paper, as a possible approach, adaptive type-2 fuzzy logic is presented.

4.1 Type-2 Fuzzy Logic

In this paper, to reduce the computational complexity, interval singleton T2FL is used, and its fundamental concept is briefly described. Its detailed description can be found in many articles [11-12], and to distinguish type-1 fuzzy set and type-2 fuzzy set, tilde '~' is used in the top of type-2 fuzzy variable. Consider a multi-input single output T2FL,

$$\begin{aligned} R^l:\ & IF\ x_1\ is\ \tilde{F}_1^l\ and\ x_2\ is\ \tilde{F}_2^l\ and\ \ldots\ and\ \ x_n\ is\ \tilde{F}_n^l \\ & THEN\ \ y\ is\ \tilde{G}_l \end{aligned} \tag{1}$$

where x_i, $i=1,2,\ldots,n$ and y are the input and output variable of the T2FLS, respectively, and fuzzy variables $\tilde{F}_i^l \subset U$ and $\tilde{G}_l \subset V$ are linguistic terms characterized by type-2 fuzzy membership functions(T2FMF) $\mu_{\tilde{F}_i^l}(x_i)$ and $\mu_{\tilde{G}_l}(y)$, respectively. When the fuzzy set is characterized by Gaussian membership function with mean m and a standard deviation that can make values in $[\sigma_1,\sigma_2]$, T2FMF is described as the following [11]:

$$\mu_{\tilde{F}_i^l}(x_i)=\exp\left[-\frac{1}{2}\left(\frac{x_i-m_i}{\sigma}\right)^2\right] \tag{2}$$

Corresponding to each values of σ, we will get a different membership curve. So the membership grade of certain x_i can take any of a number of possible values depending on the value σ, which means the membership grade is not a crisp number but a fuzzy set [11]. Each point of T2FMF has type-1 fuzzy membership function (T1FMF). When input vector $\mathbf{x}=[x_1\ \ x_2,\ldots,x_n]^T$ is applied, composition of the *l'th* rule in T2FLS with singleton fuzzifier and product (or min) inference rule can be simplified as follows:

$$\mu_{\mathbf{x}\circ\tilde{F}_1^l\times\tilde{F}_2^l\times\cdots\times\tilde{F}_n^l\to\tilde{G}^l}(y)=\mu_{\tilde{G}^l}(y)\Pi\left[\Pi_{i=1}^{n}\mu_{\tilde{F}_i^l}(x_i)\right] \tag{3}$$

Fig. 4 shows an overall block diagram of the T2FLS, and the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output.

Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation type reduction and the type-1 fuzzy set so obtained a type-reduced set. The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number. There exist many kinds of type reduction, such as centroid, center-of-sets, height, and modified height, the detailed descriptions of which are given in [11,12].

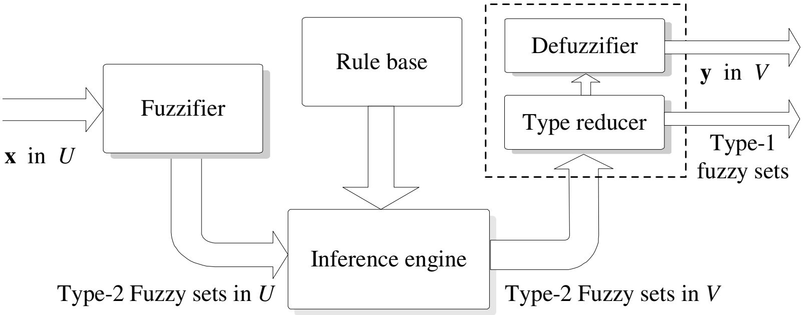


Fig. 3. Block diagram of T2FLS

Even though T2FL approach has a robustness to uncertain noisy input vector, it is still problem to handle the various threshold conditions. Using the unified output threshold value is not sufficient to distinguish meaningful gesture from similar ones because the fuzzy output value used to be very similar for each input set [2]. To remedy this problem, an adaptive scheme for fuzzy parameter using a neural network is presented in this paper. The fuzzy parameter such as output gain is to be changed to increase the defuzzified value if the input is command data while fuzzy output is to be decreased if non-command data. The overall block diagram of the proposed scheme is shown in Fig. 4.

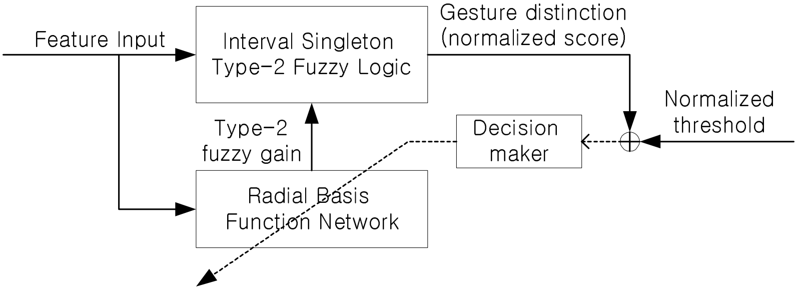


Fig. 4. Block diagram of the proposed scheme

5 Experimental Results

For the preliminary experiment, conventional and adaptive interval singleton T2FL methods are applied to discrimination of eating action and up command gesture. The characteristic can be used as features to discriminate command gesture and non-command one. As candidate features, 11 motion characteristics are listed on Table 1, and from the table, two features are selected finally based on rough set theory [2], The chosen feature set are shown on Table 2, and each of their characteristics are depicted in Figure 5.

Table 1. Candidate features for gesture spotting

- Distance change between hand and face while hand is moving
- Hold still time of hand after user finish a gesture
- Moving length of hand
- Distance between face and hand after user finish a gesture
- Consumed time of hand moving
- Eccentricity of hand moving
- Eccentricity of hand moving trajectory
- Peak frequency of hand moving
- Median value of hand moving speed
- Standard deviation of hand moving speed
- Mean value of hand moving speed

Table 2. Selected features for eating and command gesture

- Moving length of hand
- Distance between face and hand after user finish a gesture

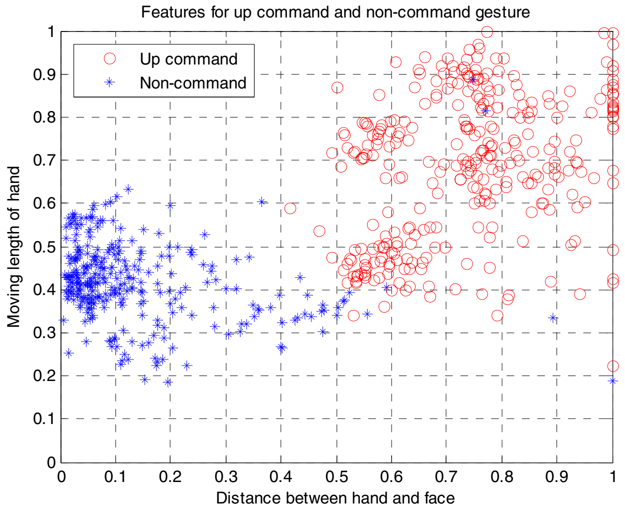


Fig. 5. Characteristics of up command and non-command gesture

Four fuzzy membership functions are used for this test, and because no negative concept exists in length and distance, only positive universe of discourse is considered. Fig. 6 and Fig. 7 show the preliminary experimental result for the conventional and the proposed ones, respectively. In each figure, the red line means the threshold value, and the proposed system can distinguish command and non-command data by changing the threshold value with the help of the radial basis function network.

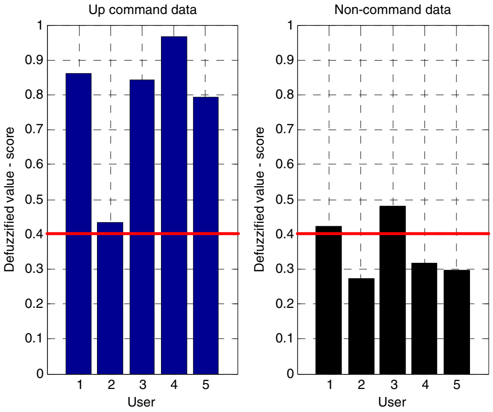


Fig. 6. Preliminary results with a conventional fuzzy logic

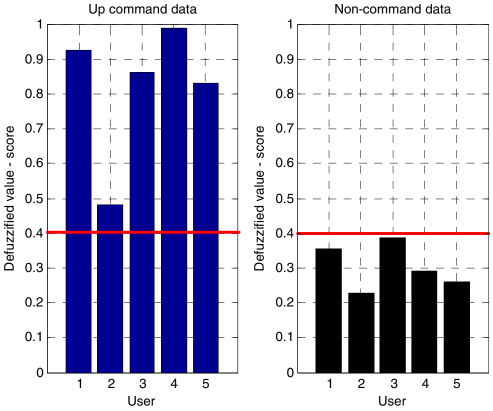


Fig. 7. Preliminary results with the proposed fuzzy logic algorithm

6 Conclusions

In this research, type-2 fuzzy logic system which discriminates a meaningful gesture and meaningless similar gestures is introduced. The proposed algorithm is applied to intelligent sweat home which has been being developed in our research center. To implement the soft-remote control system, important features of the users' gesture are selected by using rough set theory. By observing the characteristic of selected feature for each gesture, rules are generated for each case. Finally, the proposed approach is applied to control home appliances, and the preliminary experimental results show the feasibility and prospect.

Acknowledgement

This work is fully supported by the SRC/ERC program of MOST/KOSEF (Grant #R11-1999-008).

References

1. Statistical Bureau, The Management and Coordination Agency, Japan & Korea National Statistical Office, Korea.
2. S.E.Yang, Gesture Spotting using Fuzzy Garbage Model and User Adaptation, Thesis for the Master Degree, KAIST, (2006).
3. Z. Zenn Bien, Kwang-Hyun Park, and Jin-Woo Jung, Intelligent Sweet Home - Welfare-Oriented Robotic House for Assisting the Handicapped and the Elderly, Proc. of ICRA 2004, New Orleans, LA, USA, (2004).
4. Dimitar H. Stefanov, Z. Zenn Bien, and Won-Chul Bang, The Smart House for Older Persons and Persons With Physical Disabilities: Structure, Technology Arrangements, and Perspectives, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 12, no. 2, (2004).
5. Mozer. M. C., "The neural network house: An environment that adapts to its inhabitants", Proc.of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, Menlo, Park(1998), pp. 110-114.
6. H. Hagras et. al, Incremental Synchronous Learning for Embedded Agents Operating in Ubiquitous Computing Environments, Soft Computing Agents, IOS Press(2002), pp. 25-55.
7. Shin Sato, Shigeyuki Sakane, A Human-Robot Interface Using an Interactive Hand Pointer that Projects a Mark in the Real Work Space, Proc. of the IEEE ICRA (2000), pp. 589-595.
8. Neboisa Jojic, Barry Brumitt, et. al, Detection and Estimation of Pointing Gestures in Dense Disparity Maps, Automatic Face and Gesture Recognition, Proc. 4 IEEE Int. conf. (2000), p468-474.
9. L.X. Wang, Stable Adaptive Fuzzy Controllers with Application to Inverted Pendulum Tracking, IEEE Trans. on Sys. Man & Cyb. - Part B, vol. 26, no. 5, (1996), pp. 677-691.
10. H.T. Nguyen, N.R. Prasad, C.L. Walker, and E.A. Walker, A First Course in Fuzzy and Neural Control, Chapman&Hall/CRC (2003).
11. Nilesh N. Karnik, Jerry M. Mendel, and Qilian Liang, Type-2 Fuzzy Logic Systems, IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 7, NO. 6, (1999) pp643-658.
12. J. Mendel, Uncertain Rule-Based Fuzzy Logic Systems:Introduction and New Directions, Upper Saddle River, NJ: Prentice-Hall (2001).
13. Oscar Castillo, Nohe Cazarez, and Patricia Melin, Design of Stable Type-2 Fuzzy Logic Controllers based on a Fuzzy Lyapunov Approach, IEEE International Conference on Fuzzy Systems (2006), pp2331-2336.

14. Tan Woei Wan and Danesh Hasan Kamal, On-line Learning Rules For Type-2 Fuzzy Controller, IEEE International Conference on Fuzzy Systems(2006), pp 513-520.
15. Ching-Hung Lee, Hung-Yi Pan, Hua-Hsiang Chang, and Bor-Hang Wang, Decoupled Adaptive Type-2 Fuzzy Controller (DAT2FC) Design for Nonlinear TORA Systems, IEEE International Conference on Fuzzy Systems (2006), pp506-512.
16. Yasuhiro Inagaki, Hirosi Sugie, Hideyuki Aisu, Shuitiro Ono, Tatsuo Unemi, Behavior-based Intention Inference for Intelligent Robots Cooperating with Human, 1995, IEEE International Conference on Fuzzy Systems (1995), pp1695-1700.

Interval Type-1 Non-singleton Type-2 TSK Fuzzy Logic Systems Using the Hybrid Training Method RLS-BP

G.M. Mendez

Departamento de Investigacion y Desarrollo de Aplicaciones, Intelligent Systems, SA de CV,
Vista Aurora 320, Col. Linda Vista, Cd. Guadalupe, NL, México

Abstract. This paper describes a new learning methodology based on a hybrid algorithm for interval type-1 non-singleton type-2 TSK fuzzy logic systems (FLS). Using input-output data pairs during the forward pass of the training process, the interval type-1 non-singleton type-2 TSK FLS output is calculated and the consequent parameters are estimated by the recursive least-squares (RLS) method. In the backward pass, the error propagates backward, and the antecedent parameters are estimated by the back-propagation (BP) method. The proposed hybrid methodology was used to construct an interval type-1 non-singleton type-2 TSK fuzzy model capable of approximating the behaviour of the steel strip temperature as it is being rolled in an industrial Hot Strip Mill (HSM) and used to predict the transfer bar surface temperature at finishing Scale Breaker (SB) entry zone. Comparative results show the performance of the hybrid learning method (RLS-BP) against the only BP learning method.

1 Introduction

Interval type-2 (IT2) fuzzy logic systems (FLS) constitute an emerging technology. In [1] both, one-pass and back-propagation (BP) methods are presented as IT2 Mamdani FLS learning methods, but only BP is presented for IT2 Takagi-Sugeno-Kang (TSK) FLS systems. One-pass method generates a set of IF-THEN rules by using the given training data one time, and combines the rules to construct the final FLS. When BP method is used in both Mamdani and TSK FLS, none of antecedent and consequent parameters of the IT2 FLS are fixed at starting of training process; they are tuned using exclusively steepest descent method. In [1] recursive least-squares (RLS) and recursive filter (REFIL) algorithms are not presented as IT2 FLS learning methods.

The hybrid algorithm for IT2 Mamdani FLS has been already presented elsewhere [2, 3, 4] with three combinations of learning methods: RLS-BP, REFIL-BP and orthogonal least-squares (OLS)-BP, whilst the hybrid algorithm for singleton IT2 TSK FLS (IT2 TSK SFLS or IT2 ANFIS) has been presented elsewhere [5] with two combinations of learning methods: RLS-BP and REFIL-BP.

The aim of this work is to present and discuss a new hybrid learning algorithm for interval type-1 non-singleton type-2 TSK FLS (IT2 TSK NSFLS-1 or IT2 NS1 ANFIS) using RLS-BP combination in order to estimate the antecedent and consequent parameters during the training process. The proposed IT2 TSK NSFLS-1 inference system is evaluated making transfer bar surface temperature predictions at Hot Strip Mill (HSM) Finishing Scale Breaker (SB) entry zone.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 36–44, 2007.
springerlink.com

2 Proposed Methodology

2.1 Input-Output Data Pairs

Most of the industrial processes are highly uncertain, non-linear, time varying and non-stationary [2, 6], having very complex mathematical representations. Interval type-2 TSK NSFLS-1 takes easily the random and systematic components of type A or B standard uncertainty [7] of industrial measurements. The non-linearities are handled by FLS as identifiers and universal approximators of nonlinear dynamic systems [8, 9, 10, 11]. Stationary and non-stationary additive noise is modeled as a Gaussian function centered at the measurement value. In stationary additive noise the standard deviation takes a single value, whereas in non-stationary additive noise the standard deviation varies over an interval of values [1]. Such characteristics make IT2 TSK NSFLS-1 a powerful inference system to model and control industrial processes.

Only the BP learning method for IT2 TSK SFLS has been proposed in the literature and it is used as a benchmark algorithm for parameter estimation or systems identification on IT2 TSK FLS systems [1]. To the best knowledge of the authors, IT2 TSK NSFLS-1 has not been reported in the literature [1, 12, 13], using neither BP nor hybrid RLS-BP training.

One of the main contributions of this work is to implement an application of the IT2 TSK NSFLS-1 (IT2 NS1 ANFIS) using the hybrid REFIL-BP learning algorithm, capable of compensates for uncertain measurements.

2.2 Using Hybrid RLS-BP Method in Interval Type-2 TSK FLS Training

Table 1 shows the activities of the one pass learning algorithm of BP method. Both, IT2 TSK SFLS (BP) and IT2 TSK NSFLS-1 (BP) outputs are calculated during forward pass. During the backward pass, the error propagates backward and the antecedent and consequent parameters are estimated using only the BP method.

Table 1. One Pass Learning Procedure for IT2 TSK SFLS

	Forward Pass	Backward Pass
Antecedent Parameters	Fixed	BP
Consequent Parameters	Fixed	BP

The proposed hybrid algorithm (IT2 NS1 ANFIS) uses RLS during forward pass for tuning of consequent parameters as well as the BP method for tuning of antecedent parameters, as shown in Table 2. It looks like Sugeno type-1 ANFIS [13, 14], which uses the RLS-BP hybrid learning rule for type-1 FLS systems.

Table 2. Two Pass Hybrid Learning Procedure for IT2 NS1 ANFIS

	Forward Pass	Backward Pass
Antecedent Parameters	Fixed	BP
Consequent Parameters	RLS	Fixed

2.3 Adaptive Learning Algorithm

The training method is presented as in [1]: Given N input-output training data pairs, the training algorithm for E training epochs, should minimize the error function:

$$e^{(t)} = \frac{1}{2}\left[f_{IT2-FLS}\left(\mathbf{x}^{(t)}\right) - y^{(t)}\right]^2 \tag{1}$$

2.4 Hot Strip Mill

Because of the complexities and uncertainties involved in rolling operations, the development of mathematical theories has been largely restricted to two-dimensional models applicable to heat losing in flat rolling operations.

Fig. 1, shows a simplified diagram of a HSM, from the initial point of the process at the reheat furnace entry to its end at the coilers.

Besides the mechanical, electrical and electronic equipment, a big potential for ensuring good quality lies in the automation systems and the used control techniques. The most critical process in the HSM occurs in the Finishing Mill (FM). There are several mathematical model based systems for setting up the FM. There is a model-based set-up system [18] that calculates the FM working references needed to obtain gauge, width and temperature at the FM exit stands. It takes as inputs: FM exit target gage, target width and target temperature, steel grade, hardness ratio from slab chemistry, load distribution, gauge offset, temperature offset, roll diameters, load distribution, transfer bar gauge, transfer bar width and transfer bar temperature entry.

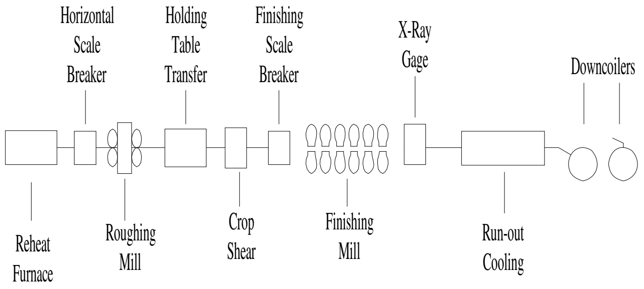


Fig. 1. Typical Hot Strip Mill

The errors in the gauge of the transfer bar are absorbed in the first two FM stands and therefore have a little effect on the target exit gauge. It is very important for the model to know the FM entry temperature accurately. A temperature error will propagate through the entire FM.

2.5 Design of the IT2 NSFLS-l

The architecture of the IT2 TSK NSFLS-1 was established in such away that its parameters are continuously optimized. The number of rule-antecedents was fixed to two; one for the Roughing Mill (RM) exit surface temperature and one for transfer bar head traveling time. Each antecedent-input space was divided in three fuzzy sets (FSs), fixing the number of rules to nine. Gaussian primary membership functions (MFs) of uncertain means were chosen for the antecedents. Each rule of the each IT2 TSK NSFLS-1 is characterized by six antecedent MFs parameters (two for left-hand and right-hand bounds of the mean and one for standard deviation, for each of the two antecedent Gaussian MFs) and six consequent parameters (one for left-hand and one for right-hand end points of each of the three consequent type-1 FSs), giving a total of twelve parameters per rule. Each input value has one standard deviation parameter, giving two additional parameters.

2.6 Noisy Input-Output Training Data

From an industrial HSM, noisy input-output pairs of three different product types were collected and used as training and checking data. The inputs are the noisy measured RM exit surface temperature and the measured RM exit to SB entry transfer bar traveling time. The output is the noisy measured SB entry surface temperature.

2.7 Fuzzy Rule Base

The IT2 TSK NSFLS-1 fuzzy rule base consists of a set of IF-THEN rules that represents the model of the system. The IT2 TSK NSFLS-1 has two inputs $x_1 \in X_1$, $x_2 \in X_2$ and one output $y \in Y$. The rule base has $M = 9$ rules of the form:

$$\begin{aligned} &R^i : IF \quad x_1 \quad is \quad \tilde{F}_1^i \quad and \quad x_2 \quad is \quad \tilde{F}_2^i, \\ &THEN \quad Y^i = \quad C_0^i + C_1^i x_1 + C_2^i x_2 \end{aligned} \tag{2}$$

where Y^i the output of the ith rule is a fuzzy type-1 set, and the parameters C_j^i, with $i = 1,2,3,\ldots,9$ and $j = 0,1,2$, are the consequent type-1 FSs.

2.8 Input Membership Functions

The primary MFs of each input of the interval type-2 NSFLS-1 are Gaussians of the form:

$$\mu_{X_k}(x_k)=\exp\left[-\frac{1}{2}\left[\frac{x_k-x_k^{'}}{\sigma_{X_k}}\right]^2\right] \tag{3}$$

where: $k=1,2$ (the number of type-2 non-singleton inputs), $\mu_{X_k}(x_k)$ is centered at $x_k=x_k^{'}$ and σ_{X_k} is the standard deviation. The standard deviation of the RM exit surface temperature measurement, σ_{X_1}, was initially set to 13.0 ^oC and the standard deviation of head end traveling time measurement, σ_{X_2}, was initially set to 2.41 s. The uncertainty of the input data is modeled as stationary additive noise using type-1 FSs.

2.9 Antecedent Membership Functions

The primary MFs for each antecedent are interval type-2 FSs described by Gaussian primary MFs with uncertain means:

$$\mu_k^i(x_k)=\exp\left[-\frac{1}{2}\left[\frac{x_k-m_k^i}{\sigma_k^i}\right]^2\right] \tag{4}$$

where $m_k^i \in \left[m_{k1}^i, m_{k2}^i\right]$ is the uncertain mean, with k =1,2 (the number of antecedents) and i = 1,2,..9 (the number of M rules), and σ_k^i is the standard deviation. The means of the antecedent FSs are uniformly distributed over the entire input space.

Table 3 shows the calculated interval values of uncertainty of x_1 input, where $[m_{11}, m_{12}]$ is the uncertain mean and σ_1 is the standard deviation for all the 9 rules. Fig. 2 shows the initial MFs of the antecedents of x_1 input.

Table 3. x_1 Input Intervals of Uncertainty

F	m_{11}	m_{12}	σ_1
S	^oC	^oC	^oC
1	950	952	60
2	1016	1018	60
3	1080	1082	60

Table 4 shows the interval values of uncertainty for x_2 input, where $[m_{21}, m_{22}]$ is the uncertain mean and σ_2 is the standard deviation for all the 9 rules. Fig. 3 shows the initial MFs of the antecedents of x_2 input.

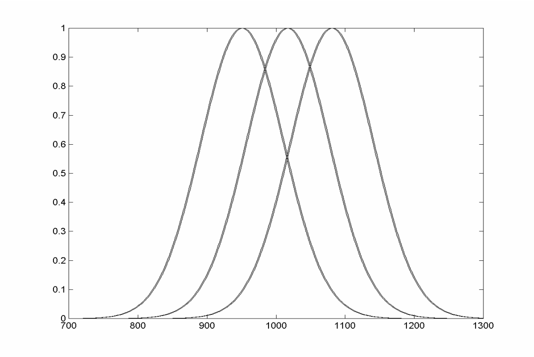


Fig. 2. MF s of the Antecedents for x_1 Input

Table 4. x_2 Input Intervals of Uncertainty

Product Type	m_{21} s	m_{22} S	σ_2 s
A	32	34	10
B	42	44	10
C	56	58	10

The standard deviation of temperature noise σ_{n1} was initially set to $1\,^{o}C$ and the standard deviation of time noise σ_{n2} was set to 1 s.

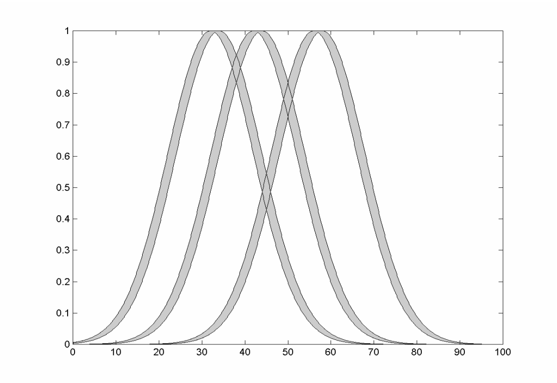


Fig. 3. MFs of the Antecedents for of x_2 Input

2.10 Consequent Membership Functions

Each consequent is an interval type-1 FS with $Y^i = \left[y_l^i, y_r^i\right]$:

$$y_l^i = \sum_{j=1}^{p} c_j^i x_j + c_0^i - \sum_{j=1}^{p} \left|x_j\right| s_j^i - s_0^i \tag{5}$$

$$y_r^i = \sum_{j=1}^{p} c_j^i x_j + c_0^i + \sum_{j=1}^{p} \left|x_j\right| s_j^i + s_0^i \tag{6}$$

where c_j^i denotes the center (mean) of C_j^i, and s_j^i denotes the spread of C_j^i, with $i = 1,2,3,..,9$. Then y_l^i and y_r^i are the consequent parameters. When only the input-output data training pairs $\left(x^{(1)} : y^{(1)}\right),\ldots,\left(x^{(N)} : y^{(N)}\right)$ are available and there is no data information about the consequents, the initial values for the centroid parameters c_j^i and s_j^i can be chosen arbitrarily in the output space [16-17]. In this work the initial values of c_j^i were set equal to 0.001 and the initial values of s_j^i equal to 0.0001.

3 Simulation Results

The IT2 TSK NSFLS-1 (RLS-BP) system was trained and used to predict the SB entry temperature, applying the RM exit measured transfer bar surface temperature and RM exit to SB entry zone traveling time as inputs. We ran fifteen epochs of training; one hundred and ten parameters were tuned using eighty seven, sixty-eight and twenty-eight input-output training data pairs per epoch, for type A, type B and type C products respectively.

The performance evaluation for the hybrid IT2 TSK NSFLS-1 (RLS-BP) system was based on root mean-squared error (RMSE) benchmarking criteria as in [1]:

$$RMSE_{IT2-FLS}(*) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left[Y(k) - f_{s2-*}\left(\mathbf{x}^{(k)}\right)\right]^{2}} \tag{7}$$

where $Y(k)$ is the output data from the input-output checking data pairs. $RMSE_{IT2-FLS}(*)$ stands for $RMSE_{TSK2,SFLS}(BP)$[the RMSE of the IT2 TSK SFLS (BP)] and for $RMSE_{TSK2,NSFLS-1}(BP)$ [the RMSE of the IT2 TSK NSFLS-1(BP)], whereas $RMSE_{TSK2,NSFLS-1}(REFIL-BP)$ [the RMSE of the IT2 TSK NSFLS-1(RLS-BP)] is obtained when the hybrid algorithm is applied to IT2 TSK NSFLS-1.

Fig. 4 shows the RMSEs of the two IT2 TSK SNFLS-1 systems and the base line IT2 TSK SFLS (BP) for fifty epochs' of training for the case for type C products. Observe that from epoch 1 to 4 the hybrid IT2 TSK NSFLS-1 (RLS-BP) has better performance than both: the IT2 TSK SFLS (BP) and the IT2 TSK NSFLS-1 (BP). From epoch 1 to 4 the RMSE of the IT2 TSK SFLS has an oscillation, meaning that it is very sensitive to its learning parameters values. At epoch 5, it reaches its minimum RMSE and is stable for the rest of training.

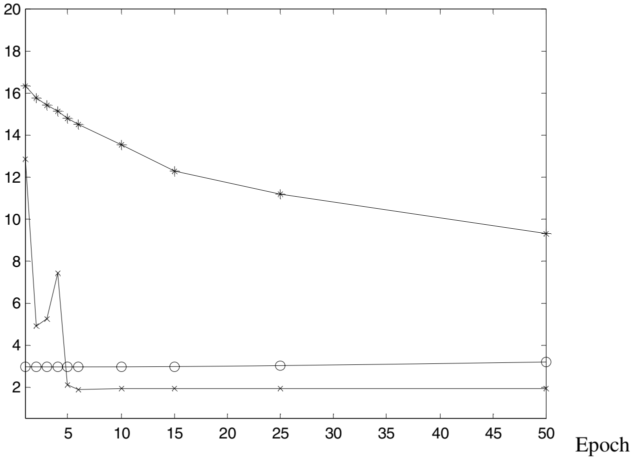


Fig. 4. (*) RMSE $_{\mathrm{TSK\ 2,\ SFLS}}$ (BP) (+) RMSE $_{\mathrm{TSK\ 2,\ NSFLS\text{-}1}}$ (BP) (o) RMSE $_{\mathrm{TSK\ 2,\ NSFLS\text{-}1}}$ (REFIL-BP)

4 Conclusions

An IT2 TSK NSFLS-1 (IT2 NS1 ANFIS) using the hybrid RLS-BP training method was tested and compared for predicting the surface temperature of the transfer bar at SB entry. The antecedent MFs and consequent centroids of the IT2 TSK NSFLS-1 tested, absorbed the uncertainty introduced by all the factors: the antecedent and consequent values initially selected, the noisy temperature measurements, and the inaccurate traveling time estimation. The non-singleton type-1 fuzzy inputs are able to compensate the uncertain measurements, expanding the applicability of IT2 NS1 ANFIS systems.

It has been shown that the proposed IT2 NS1 ANFIS system can be applied in modeling and control of the steel coil temperature. It has also been envisaged its application in any uncertain and non-linear system prediction and control.

Acknowledgements

The authors acknowledge the financial support and facilities provided by Intelligent Systems, SA de CV.

References

[1] J. M. Mendel, Uncertain rule based fuzzy logic systems: introduction and new directions, Upper Saddle River, NJ, Prentice-Hall, 2001.

[2] G. M. Mendez, A. Cavazos, L. Leduc, R. Soto, "Hot Strip Mill Temperature Prediction Using Hybrid Learning Interval Singleton Type-2 FLS," Proceedings of the IASTED International Conference on Modeling and Simulation, Palm Springs, pp. 380-385, February 2003.

[3] G. M. Mendez, A. Cavazos, L. Leduc, R. Soto, "Modeling of a Hot Strip Mill Temperature Using Hybrid Learning for Interval Type-1 and Type-2 Non-Singleton Type-2 FLS," Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Benalmádena, Spain, pp. 529-533, September 2003.

[4] G. M. Mendez, I.J. Juarez, "Orthogonal-Back Propagation Hybrid Learning Algorithm for Interval Type-1 Non-Singleton Type-2 Fuzzy Logic Systems," WSEAS Transactions on Systems, vol. 4, No. 4, March 2005.

[5] G. M. Mendez, O. Castillo, "Interval Type-2 TSK Fuzzy Logic Systems Using Hybrid Learning Algorithm," FUZZ-IEEE 2005 The international Conference on Fuzzy Systems, Reno Nevada, USA, pp. 230-235, 2005.

[6] D. Y. Lee, H. S. Cho, "Neural Network Approach to the Control of the Plate Width in Hot Plate Mills," International Joint Conference on Neural Networks, vol. 5, pp. 3391-3396, 1999.

[7] B. N. Taylor, C. E. Kuyatt, Guidelines for evaluating and expressing the uncertainty of NIST measurement results, NIST Technical Note 1297, September, 1994.

[8] L-X. Wang, "Fuzzy Systems are Universal Approximators," Proceedings of the IEEE Conf. On Fuzzy Systems, San Diego, pp. 1163-1170, 1992.

[9] L-X. Wang, J. M. Mendel, "Back-Propagation Fuzzy Systems as Nonlinear Dynamic System Identifiers," Proceedings of the IEEE Conf. On Fuzzy Systems, San Diego, CA., pp. 1409-1418, March 1992.

[10] L-X. Wang, "Fuzzy Systems are Universal Approximators," Proceedings of the IEEE Conf. On Fuzzy Systems, pp. 1163-1170, San Diego, 1992.

[11] L-X. Wang, A course in fuzzy systems and control, Upper Saddle River, NJ: Prentice Hall PTR, 1997.

[12] J.-S. R Jang,. C.-T. Sun, E. Mizutani, Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, Upper Saddle River, NJ: Prentice-Hall, 1997.

[13] J.-S R. Jang., C.-T. Sun, "Neuro-Fuzzy Modeling and Control," The Proceedings of the IEEE, Vol. 3, pp. 378-406, Mach 1995.

[14] Q. J. Liang, J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," Trans. Fuzzy Sist., Vol. 8, pp. 535-550, Oct. 2000.

[15] R. I. John, "Embedded Interval Valued Type-2 Fuzzy Sets," Proc. Of FUZZ-IEEE, pp. 1316-1321, 2002.

[16] J. M. Mendel, R.I. John, "Type-2 Fuzzy Sets Made Simple," IEEE Transactions on Fuzzy Systems, vol. 10, April 2002.

[17] J. M. Mendel, "On the importance of interval sets in type-2 fuzzy logic systems," Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001.

[18] GE Models, Users reference, vol. 1, Roanoke VA, 1993.

An Efficient Computational Method to Implement Type-2 Fuzzy Logic in Control Applications

Roberto Sepúlveda[1], Oscar Castillo[2], Patricia Melin[2], and Oscar Montiel[1]

[1] Centro de Investigación y Desarrollo de Tecnología Digital del Instituto
Politécnico Nacional (CITEDI-IPN) Av. del Parque No. 1310, Mesa de
Otay, Tijuana, B.C., México
rsepulve@citedi.mx, o.montiel@ieee.org

[2] Division of Graduate Studies, Tijuana Institute of Technology P.O. Box 4207,
Chula Vista CA 91909, USA
ocastillo@tectijuana.mx, pmelin@tectijuana.mx

Abstract. A novel structure of type 2 fuzzy logic controller is presented. The method is highly efficient regarding computational time and implementation effort. Type-2 input membership functions were optimized using the Human Evolutionary Model (HEM) considering as the objective function the Integral of Squared Error at the controllers output. Statistical tests were achieved considering how the error at the controller's output is diminished in presence of uncertainty, demonstrating that the proposed method outperforms an optimized traditional type-2 fuzzy controller for the same test conditions.

1 Introduction

In engineering as well as in the scientific field is of growing interest to use type-2 fuzzy logic controller (FLC). It is a well documented fact that type-2 FLC had demonstrated in several fields their usefulness to handle uncertainty which is an inherent characteristic of real systems. Because uncertainty and real systems are inseparable characteristics the research of novel methods to handle incomplete or not too reliable information is of great interest [13]. Recently, we have seen the use of type-2 fuzzy sets in Fuzzy Logic Systems (FLS) in different areas of application. From those including fuzzy logic systems, neural networks and genetic algorithms [8], to some papers with emphasis on the implementation of type-2 FLS [7, 11]; in others, it is explained how type-2 fuzzy sets let us model and minimize the effects of uncertainties in rule-base FLS [5, 15]. Also, a paper that provides mathematical formulas and computational flowcharts for computing the derivatives that are needed to implement steepest-descent parameter tuning algorithms for type-2 fuzzy logic systems [14]. Some research works are devoted to solve real world applications in different areas, for example in signal processing, type-2 fuzzy logic is applied in prediction of the Mackey-Glass chaotic time-series with uniform noise presence [6, 12]. In medicine, an expert system was developed for solving the problem of Umbilical Acid-Base (UAB) assessment [17]. In industry, type-2 fuzzy logic and neural networks was used in the control of non-linear dynamic plants [2, 3, 9,10]; also we can find interesting studies in the field of mobile robots [1, 4].

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 45–52, 2007.
springerlink.com

Although, the use of a type-2 FLC can be considered as a viable option to handle uncertainty, also it is well known all the deficiencies and requirements that the use of this technology implies.

In this work we are presenting a method whose goal is to simplify the implementation of a type-2 FLC without any loss of reliability in the results. In fact, this novel method reduces some of the stressful difficult to implement the traditional type-2 FLC.

The organization of this work is as follows: In section 2 is explained step by step how to implement this proposal and the method used to optimize the traditional as well as the proposed type-2 FLC. Section 3 is devoted to explain the kind and classification of experiments that were achieved, also in this section are given the experimental results. In section 4 is performed a discussion about the obtained results. Finally, in section 5 we have the conclusions.

2 Proposed Method to Implement Type-2 FLC

It is proposed to use to type-1 fuzzy systems (FS) to emulate a type-2 FS. The membership functions (MF), fuzzification process, fuzzy inference and defuzzification are type-1. The MFs are organized in such a way that they will be able to emulate the footprint of uncertainty (FOU) in a type-2 FS. To obtain the best parametric values for the MF the proposed method uses the optimized MFs, we used the Human Evolutionary Model (HEM) to achieve the optimization.

To validate the proposal, we made several comparative experiments using type-1 fuzzy traditional systems, as well as type-2 interval FS in accordance to those worked by Dr. Jerry Mendel. The tests were achieved in the experimental base shown in Fig. 1 which is a closed loop control system. The control goal is to make a tracking of the input signal r, which is applied to the systems summing junction. Note that we are using an adaptive fuzzy controller that needs to be optimized. In the feedback, with the aim of proving the proposal, we are considering two situations. One is to

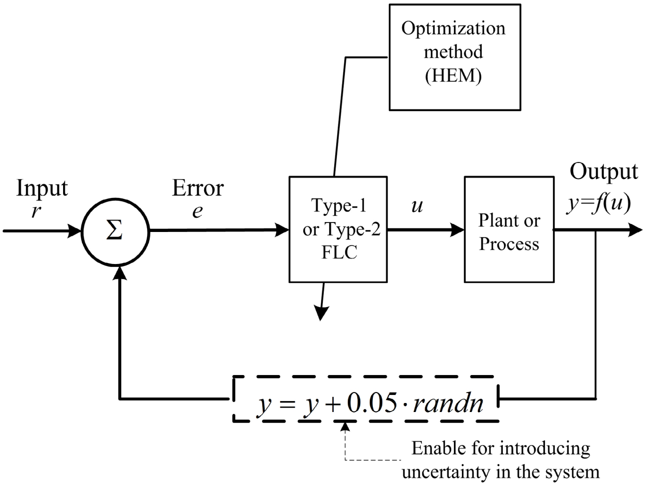

Optimization method (HEM)
Input
r
Σ
Error
e
Type-1 or Type-2 FLC
u
Plant or Process
Output
y=f(u)
$y = y + 0.05 \cdot randn$
Enable for introducing uncertainty in the system

Fig. 1. Block diagram of the system used to test the proposal solution

directly connect the system output to one summing junction. The second is to introduce noise to simulate uncertainty in the feedback data. At the summing junction output we have the error signal, which is applied to the input of the fuzzy controller, from the error signal we are obtaining a derivative signal; i.e., the change of error vs. time, which also is applied to the controllers input.

In general, the proposal solution to substitute the Mendel's type-2 FS consists in using the average of two type-1 FS, to achieve this is necessary to follow the next steps:

1. To substitute each type-2 MF for two type-1 MFs. For doing this, the FOU of each MF is substituted for two type-1 MF. In Fig. 2, the error signal (input fuzzy variable) *e* consists of three linguistic variables, they have been substituted as was explained obtaining the fuzzy sets that are shown in Fig. 3 where each fuzzy set is a type-1 MF. The first type-1 FLC (FLC1) is constructed using the upper MFs, and the second one (FLC2) with the lower MFs.

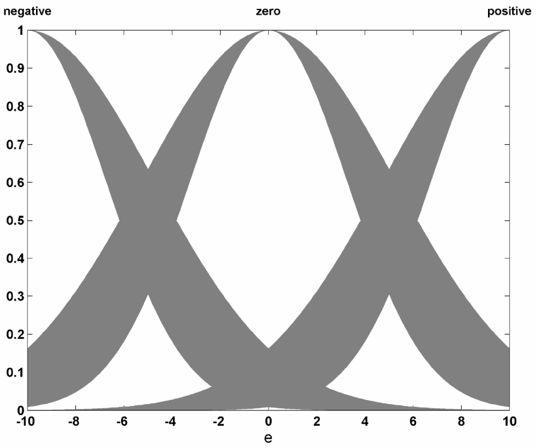


Fig. 2. Type-2 MF for the error input

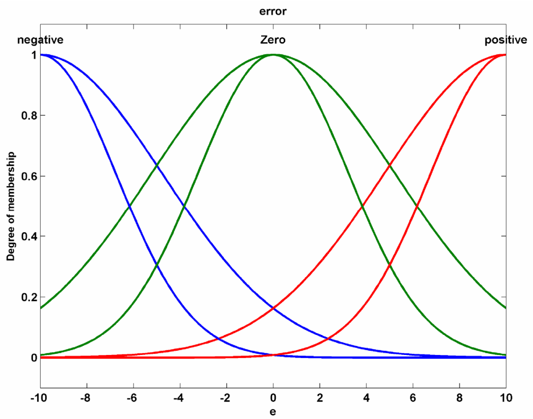


Fig. 3. Substitution of the type-2 MFs of the error input using type-1 MFs

2. To substitute the type-2 inference system, it is necessary to obtain the inference of each type-1 system in the traditional way. 1
3. To substitute the type reduction and defuzzification stages of a type-2 FS, it is necessary to obtain the defuzzification of each system as is traditionally done, and average them.

2.1 Performance Criteria

For evaluating the transient closed-loop response of a computer control system we can use the same criteria that normally are used for adjusting constants in PID (Proportional Integral Derivative) controllers. These are [18]:

Integral of Square Error (ISE).

$$\mathrm{ISE} = \int_0^\infty [e(t)]^2\, dt$$

Integral of the Absolute value of the Error (IAE).

$$\mathrm{IAE} = \int_0^\infty | e(t) |\, dt$$

Integral of the Time multiplied by the Absolute value of the Error (ITAE).

$$\mathrm{ITAE} = \int_0^\infty t\, | e(t) |\, dt$$

The selection of the criteria depends on the type of response desired, the errors will contribute different for each criterion, so we have that large errors will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, time appears as a factor, and therefore, ITAE will penalize heavily errors that occur late in time, but virtually ignores errors that occur early in time

3 Experiments

The experiments were divided in two classes:

1. The first class was to find, under different ranges for the FOU, the optimal values for the parameters of the interval type-2 MFs of the type-2 FLC of the non-linear control plant.
2. On the second class of experiments; it was realized the same as in the first class, but considering the average of the two type-1 FLC.

3.1 Class 1. Experiments with Type-2 FLC

It is a fact that type-2 FLCs offer better conditions to handle uncertainty, so the purpose of the experiments of class 1, were to find the optimal parameters of the interval type-2 MFs to control the plant in a better way.

It was used a novel evolutive algorithm; Human Evolutive Model [16], to find those optimal values and to analize the influence of the FOU, we realized several tests for different ranges of it, beginning with the thinner and finally with the broader one. Once the optimal values were found, it was tested the behavior of the type-2 FLC, for different noise levels, from 8 db to 30 db.

3.2 Class 2. Experiments with Average of Two FLCs

To control the plant, we used the proposal solution of using the average of two type-1 FLC to simulate a type-2 FLC. For these experiments, it was considered that one type-1 FLC manage and fixed the upper MFs, and the other the low MFs. Here, in the same way as in experiments of class 1, from the optimal values found for the MFs, it was tested the behavior of the average of two type-1 FLC, for different noise levels, from 8 db to 30 db.

4 Results

The HEM was the optimization method that we used. The initial setting for each range of the FOU for this evolutionary method were:

Initial population of individuals =20
Low bound of individuals=10
Upper bound of individuals=100
Number of variables=6 (Standard deviation of each of the MFs of the inputs).
Number of generations=60

The search process was repeated 30 times, always looking for the optimal parameter values to obtain the lowest ISE value.

4.1 Class 1

In figures. 4 and 5 can be seen the optimized MFs that obtained the best results in the control of the plant.

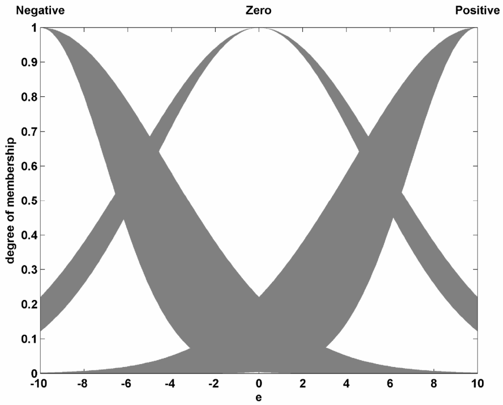


Fig. 4. Optimized MFs of the input error *e* of the type-2 FLC, for a 2.74 to 5.75 range of the FOU

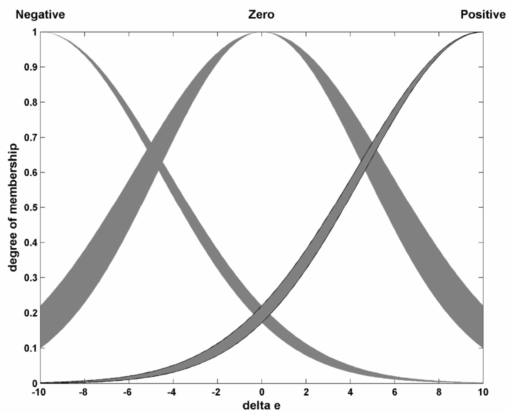


Fig. 5. Optimized MFs of the input delta-*e* of the type-2 FLC, for a 2.74 to 5.75 range of the FOU

4.2 Class 2

In figures 6 and 7, we can see the optimized MFs of the average of two type-1 FLCs, here as in Class 1, the best results were obtained in the broader range search.

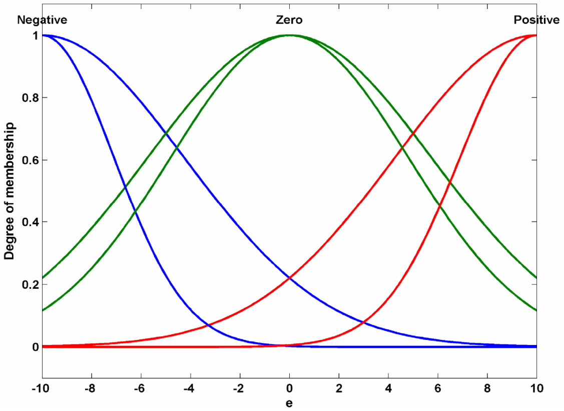


Fig. 6. Optimized MFs of the input error e of the average of two type-1 FLC, for a 2.74 to 5.75 range of the FOU

Table 1, shows a comparison of the ISE values obtained for each FLC with its optimized MFs. As can be seen, with the proposal of two optimizedtype-1 FLCs, the ISE error is lower in all the search ranges.

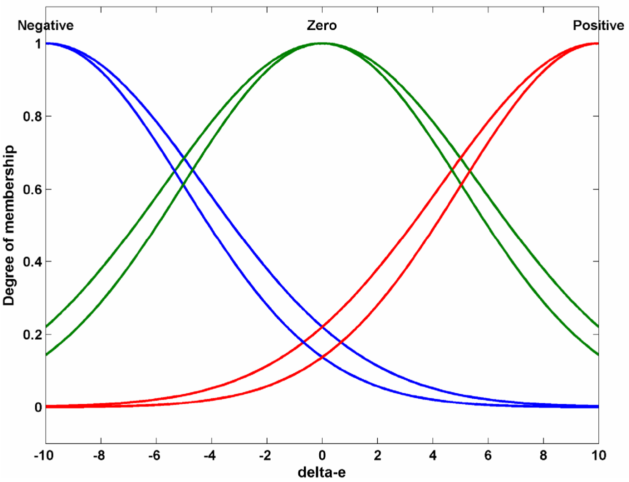


Fig. 7. Optimized MFs of the input delta-*e* of the two type-1 FLC, for a 2.74 to 5.75 range of the FOU

Table 1. Comparison values between Type-2 FLC and average of two type-1 FLCs

Search range	TYPE-2 FLC		AVERAGE TYPE-1 FLCs	
	Best ISE	AVERAGE ISE	Best ISE	**AVERAGE ISE**
3.74-4.75	4.761	4.9942	4.5619	**4.7701**
3.24-5.25	4.328	4.5060	4.2024	**4.4009**
2.74-5.75	**4.3014**	**4.4005**	**4.1950**	**4.346**

5 Conclusions

Based on the results of the experiments, we can conclude that the proposed method, that consists in using two optimized type-1 FLCs instead of a optimized traditional type-2 FLC, is a convenient and viable alternative because it offers advantages such as a highly efficient regarding computational time and implementation effort. The type-2 FLCs need to realize a complex task in each step of the process, specially in the type reduction case.

With the proposed method it is easier to optimized the parameters of the MFs of a type-1 FLC than an interval type-2 FLC.

References

[1] L. Astudillo, O. Castillo, L. T. Aguilar, Intelligent Control of an Autonomous Mobile Robot Using Type-2 Fuzzy Logic, in: Proc. Conference on Artificial Intelligence, ICAI´06, Las Vegas NV, 2006, pp. 565-570.

[2] O. Castillo, P. Melin, Soft Computing for Control of Non-linear Dynamical Systems, Springer-Verlag, Heidelberg, Germany, 2001.

[3] O. Castillo, P. Melin, A New Approach for Plant Monitoring using Type-2 Fuzzy Logic and Fractal Theory, International J. of General Systems 33 (2) (2004) 305-319.

[4] H. A. Hagras, Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 524-539.

[5] W. Hongwei, J. M. Mendel, Introduction to Uncertainty bounds and Their Use in the Design of Interval Type-2 Fuzzy Logic Systems, in: Proc. FUZZ-IEEE 2001, Melbourne, Australia, 2001, pp. 662-665.

[6] N. N. Karnik, J. M. Mendel, Applications of Type-2 Fuzzy Logic Systems to Forecasting of Time-series, Information Sciences 120 (1-4) (1999) 89-111.

[7] N. N. Karnik, J. M. Mendel, Q. Liang, Type-2 Fuzzy Logic Systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 643-658.

[8] C.-H. Lee, J.-L. Hong, Y.-C. Lin and W.-Y. Lai, Type-2 Fuzzy Neural Network Systems and Learning, International Journal of Computational Cognition 1 (4) (2003) 79-90.

[9] P. Melin, O. Castillo, A new method for adaptive model-based control of non-linear dynamic plants using type-2 fuzzy logic and neural networks, in: Proc. 12th IEEE International conference on Fuzzy Systems, St. Louis, Missouri, 2003, pp. 420-425.

[10] P. Melin, O. Castillo, A New Method for Adaptive Control of Non-Linear Plants using Type-2 Fuzzy Logic and Neural Networks, International J. of General Systems 33 (2) (2004) 289-304.

[11] J. M. Mendel, Type-2 fuzzy logic systems: Type-reduction, in: IEEE Syst., Man, Cybern. Conf., San Diego, CA, 1998, pp. 2046-2051.

[12] J. M. Mendel, Uncertainty, fuzzy logic, and signal processing, Signal Processing Journal. 80 (2000) 913-933.

[13] J. M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and new directions, Prentice Hall, New Jersey, 2001.

[14] J. M. Mendel, Computing Derivatives in Interval Type-2 Fuzzy Logic Systems, IEEE Transactions on Fuzzy Systems 12 (2004) 84-98.

[15] J. M. Mendel, R. I. B. John, Type-2 Fuzzy Sets Made Simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 117-127.

[16] O. Montiel, O. Castillo, P. Melin, A. Rodríguez, R. Sepúlveda, The Human Evolutionary Model. Journal of Intelligent System. Special Issue: Hybrid Intelligent Systems For Time Series Prediction,, 14, (.2-3) (2005) 213-236.

[17] T. Ozen, J. M. Garibaldi, Investigating Adaptation in Type-2 Fuzzy Logic Systems Applied to Umbilical Acid-Base Assessment, in: Proc. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (EUNITE 2003), Oulu, Finland, 2003.

[18] J. G. Proakis, D. G. Manolakis, DIGITAL SIGNAL PROCESSING principles, Algorithms, and Applications, Third Edition, Prentice Hall, New Jersey, 1996.

Building Fuzzy Inference Systems with the Interval Type-2 Fuzzy Logic Toolbox

J.R. Castro, O. Castillo, P. Melin, L.G. Martínez, S. Escobar, and I. Camacho

Baja California Autonomous University, UABC. Tijuana, Mexico
jrcastror@uabc.mx, luisgmo@uabc.mx, saeman@hotmail.com, iliana_m_@hotmail.com
Department of Computer Science. Tijuana Institute of Technology. Tijuana, Mexico
ocastillo@tectijuana.mx, pmelin@tectijuana.mx

Abstract. This paper presents the development and design of a graphical user interface and a command line programming Toolbox for construction, edition and simulation of Interval Type-2 Fuzzy Inference Systems. The Interval Type-2 Fuzzy Logic System Toolbox (IT2FLS), is an environment for interval type-2 fuzzy logic inference system development. Tools that cover the different phases of the fuzzy system design process, from the initial description phase, to the final implementation phase, constitute the Toolbox. The Toolbox's best qualities are the capacity to develop complex systems and the flexibility that allows the user to extend the availability of functions for working with the use of type-2 fuzzy operators, linguistic variables, interval type-2 membership functions, defuzzification methods and the evaluation of Interval Type-2 Fuzzy Inference Systems.

Keywords: Interval Type-2 Fuzzy Inference Systems, Interval Type-2 Fuzzy Logic Toolbox, Interval Type-2 Membership Functions, Footprint of Uncertainty.

1 Introduction

On the past decade, fuzzy systems have displaced conventional technologies in different scientific and system engineering applications, especially in pattern recognition and control systems. The same fuzzy technology, in approximation reasoning form, is resurging also in the information technology, where it is now giving support to decision-making and expert systems with powerful reasoning capacity and a limited quantity of rules. The fuzzy sets were presented by L.A. Zadeh in 1965 [1-3] to process / manipulate data and information affected by unprobabilistic uncertainty/imprecision. These were designed to mathematically represent the vagueness and uncertainty of linguistic problems; thereby obtaining formal tools to work with intrinsic imprecision in different type of problems; it is considered a generalization of the classic set theory. Intelligent Systems based on fuzzy logic are fundamental tools for nonlinear complex system modeling. Fuzzy sets and fuzzy logic are the base for fuzzy systems, where their objective has been to model how the brain manipulates inexact information. Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way. These type-2 fuzzy sets were originally presented by Zadeh in 1975 and are essentially "fuzzy fuzzy" sets where the fuzzy degree of

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 53–62, 2007.
springerlink.com

membership is a type-1 fuzzy set [4,6]. The new concepts were introduced by Mendel and Liang [8,9] allowing the characterization of a type-2 fuzzy set with a inferior membership function and an superior membership function; these two functions can be represented each one by a type-1 fuzzy set membership function. The interval between these two functions represents the footprint of uncertainty (FOU), which is used to characterize a type-2 fuzzy set. The uncertainty is the imperfection of knowledge about the natural process or natural state. The statistical uncertainty is the randomness or error that comes from different sources as we use it in a statistical methodology. Type-2 fuzzy sets have been applied to a wide variety of problems by Castillo and Melin [13].

2 Interval Type-2 Fuzzy Set Theory

A type-2 fuzzy set [6,7] expresses the non-deterministic truth degree with imprecision and uncertainty for an element that belongs to a set. A type-2 fuzzy set denoted by $\tilde{\tilde{A}}$, is characterized by a type-2 membership function $\mu_{\tilde{\tilde{A}}}(x,u)$, where x∈X, u ∈ $J_x^u \subseteq [0,1]$ and $0 \le \mu_{\tilde{\tilde{A}}}(x,u) \le 1$ is defined in equation (1).

$$\tilde{\tilde{A}} = \{(x,\mu_{\tilde{\tilde{A}}}(x)) \mid x \in X\} = \left\{ \int_{x\in X} \left[\int_{u\in J_x^u \subseteq [0,1]} f_x(u)/u \right] / x \right\} \tag{1}$$

An example of a type-2 membership function constructed in the IT2FLS Toolbox was composed by a Pi primary and a Gbell secondary type-1 membership functions, these are depicted in Figure 1.

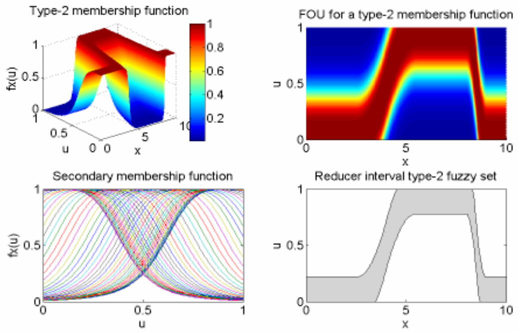


Fig. 1. FOU for Type-2 Membership Functions

If $f_x(u) = 1, \forall u \in [\underline{J}_x^u, \overline{J}_x^u] \subseteq [0,1]$, the type-2 membership function $\mu_{\tilde{\tilde{A}}}(x,u)$ is expressed by one inferior type-1 membership function, $\underline{J}_x^u \equiv \underline{\mu}_A(x)$ and one superior type-1 membership function, $\overline{J}_x^u \equiv \overline{\mu}_A(x)$ (Fig. 2), then it is called an interval type-2 fuzzy set [8] denoted by equations (2) and (3).

$$\tilde{\tilde{A}} = \left\{ \int_{x\in X} \left[\int_{u\in[\underline{\mu}_A(x),\overline{\mu}_A(x)]\subseteq[0,1]} 1/u \right] / x \right\} \tag{2}$$

If $\tilde{\tilde{A}}$ is a type-2 fuzzy Singleton, the membership function is defined by equation (6).

$$\mu_{\tilde{\tilde{A}}}(x) = \left\{ \begin{matrix} 1/1 & si\ x = x' \\ 1/0 & si\ x \neq x' \end{matrix} \right\} \tag{3}$$

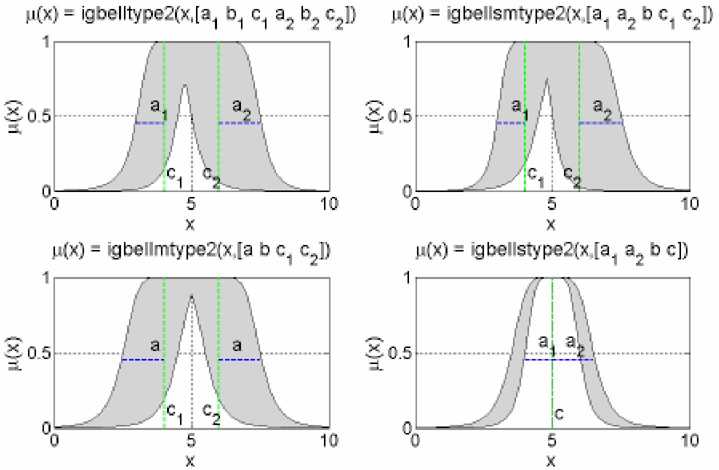


Fig. 2. FOU for Gbell Primary Interval Type-2 Membership Functions

2.1 Interval Type-2 Fuzzy Inference System

The human knowledge is expressed as a set of fuzzy rule. The fuzzy rules are basically of the form IF <Antecedent> THEN <Consequent> and expresses a fuzzy relationship or proposition. In fuzzy logic the reasoning is imprecise, it is approximated, which means that we can infer from one rule a conclusion even if the antecedent doesn't comply completely. We can count on two basic inference methods between rules and inference laws, Generalized Modus Ponens (GMP) [5,6,8,11] and Generalized Modus Tollens (GMT) that represent the extensions or generalizations of classic reasoning. The GMP inference method is known as direct reasoning and is resumed as:

Rule *IF x is A THEN y is B*
Fact *x is A'*

Conclusion *y is B'*

Where A, A', B and B' are fuzzy sets of any kind. This relationship is expressed as $B' = A' \circ (A \rightarrow B)$. Figure 3 shows an example of Interval Type-2 direct reasoning with Interval Type-2 Fuzzy Inputs. An Inference Fuzzy System is a rule base system that uses fuzzy logic, instead of Boolean logic utilized in data analysis [4,9,11,12]. Its basic structure includes four components (Fig. 4):

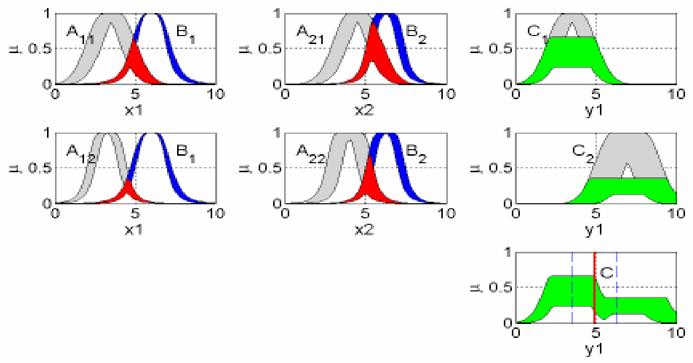


Fig. 3. Interval Type-2 Fuzzy Reasoning

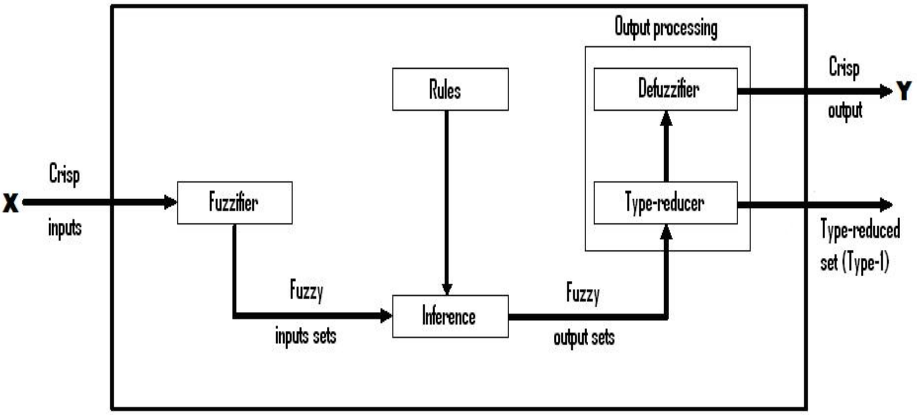
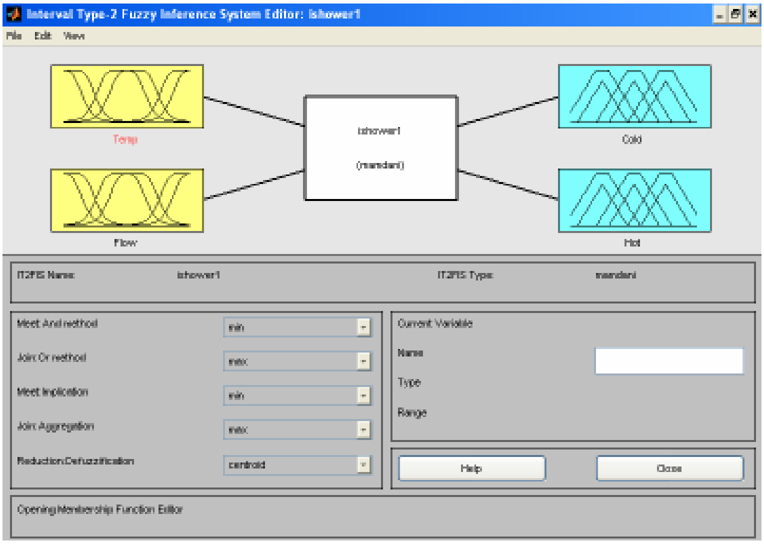


Fig. 4. Type-2 inference fuzzy system structure

3 Interval Type-2 Fuzzy Logic System Design

The Mamdani and Takagi-Sugeno-Kang Interval Type-2 Fuzzy Inference Models [9] and the design of Interval Type-2 membership functions and operators are implemented in the IT2FLS (Interval Type-2 Fuzzy Logic Systems) Toolbox which was build on top of the Matlab® commercial Fuzzy Logic Toolbox. The IT2FLS Toolbox contain the functions to create Mamdani and TSK Interval Type-2 Fuzzy Inference Systems (newfistype2.m), functions to add input-output variables and their ranges (addvartype2.m), it has functions to add 22 types of Interval Type-2 Membership functions for input-outputs (addmftype2.m), functions to add the rule matrix (addruletype2.m), it can evaluate the Interval Type-2 Fuzzy Inference Systems (evalifistype2.m), evaluate Interval Type-2 Membership functions (evalimftype2.m), it can generate the initial parameters of the Interval Type-2 Membership functions (igenparamtype2.m), it can plot the Interval Type-2 Membership functions with the



IT2FIS Name: ishower1

IT2FIS Type: mamdani

Meet And method: min

Join Or method: max

Meet Implication: min

Join Aggregation: max

Reduction Defuzzification: centroid

Current Variable

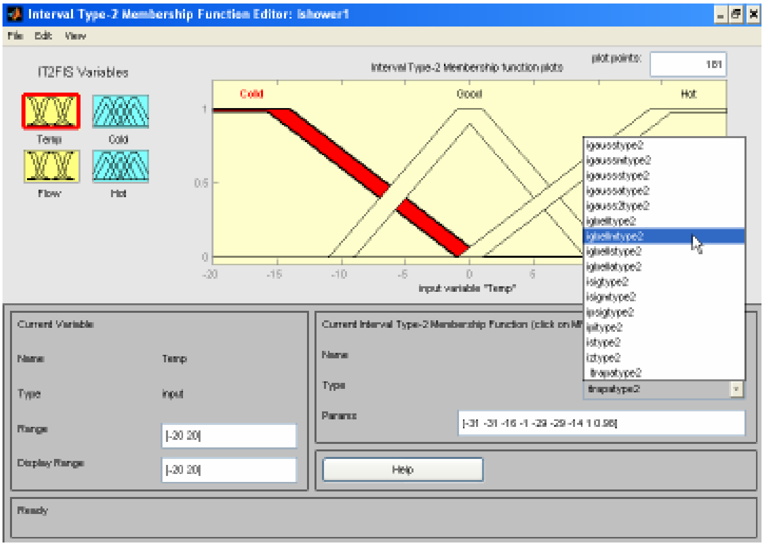
Name

Type

Range

Opening Membership Function Editor

Fig. 5. IT2FIS Editor



plot points: 181

Current Variable

Name: Temp

Type: input

Range: [-20 20]

Display Range: [-20 20]

Current Interval Type-2 Membership Function (click on MF

Name

Type: itrapatype2

Params: [-31 -31 -16 -1 -29 -29 -14 1 0.98]

Ready

Fig. 6. Interval Type-2 MF's Editor

input-output variables (plotimftype2.m), it can generate the solution surface of the Fuzzy Inference System (gensurftype2.m), it plots the Interval type-2 membership functions (plot2dtype2.m, plot2dctype2.m), a folder to evaluate the derivatives of the Interval type-2 Membership Functions (dit2mf) and a folder with different and generalized Type-2 Fuzzy operators (it2op, t2op).

The implementation of the IT2FLS GUI is analogous to the GUI used for Type-1 FLS in the Matlab® Fuzzy Logic Toolbox, thus allowing the experienced user to adapt easily to the use of IT2FLS GUI [15]. Figures 5 and 6 show the main viewport of the Interval Type-2 Fuzzy Inference Systems Structure Editor called IT2FIS (Interval Type-2 Fuzzy Inference Systems).

4 Simulation Results

We present results of a comparative analysis of the Mackey-Glass chaotic time-series forecasting study using intelligent cooperative architecture hybrid methods, with neural networks, (Mamdani, Takagi-Sugeno-Kang) type-1 fuzzy inference systems and genetic algorithms (neuro-genetic, fuzzy-genetic and neuro-fuzzy) and an interval type-2 fuzzy logic model, for the implicit knowledge acquisition in a time series behavioral data history [14]. Also we present a shower simulation and a truck backer-upper simulation with interval type-2 fuzzy logic systems using the IT2FLS Toolbox.

4.1 Mackey-Glass Chaotic Time-Series

To identify the model we make an exploratory series analysis with 5 delays, $L^5x(t)$, 6 periods and 500 training data values to forecast 500 output values. The IT2FLS(Takagi-Sugeno-Kang) system works with 4 inputs, 4 interval type-2 membership functions (igbellmtype2) for each input, 4 rules (Fig. 7) and one output with 4 interval lineal functions, it is evaluated with no normalized values. The root mean square error (RMSE) forecasted is 0.0235. Table 1 shows the RMSE differences

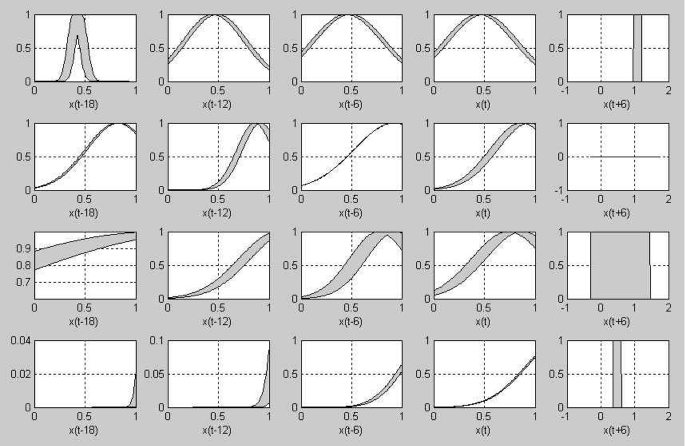


Fig. 7. IT2FLS (TSK) Rules

Table 1. Forecasting of Time Series

Methods	Mackey-Glass			
	RMSE	*trn/chk*	*epoch*	*cpu(s)**
NNFF**†	0.0595	500/500	200	13.36
CANFIS	0.0016	500/500	50	7.34
NNFF-GA†	0.0236	500/500	150	98.23
FLS(TKS)-GA†	0.0647	500/500	200	112.01
FLS(MAM)-GA†	0.0693	500/500	200	123.21
IT2FLS	0.0235	500/500	6	20.47

* POWER BOOK G4 1.5 Ghz / 512 MB RAM

** Architecture: 4-13-1 † 30 samples average

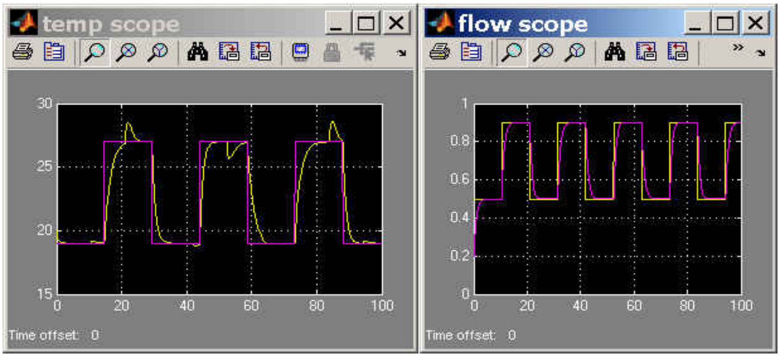


Fig. 8. Temperature and Flow. Type-1 fuzzy control.

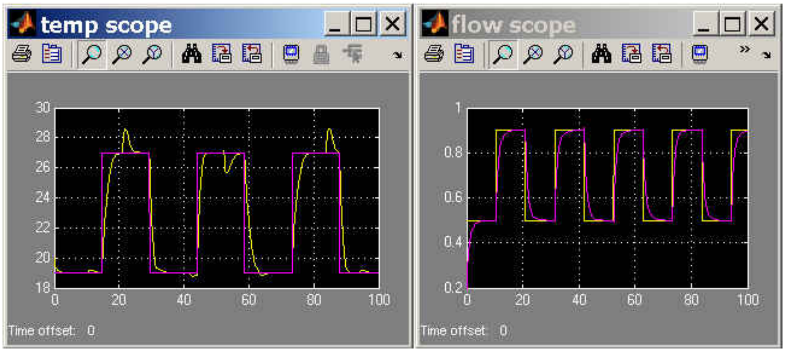


Fig. 9. Temperature and Flow. Interval type-2 fuzzy control.

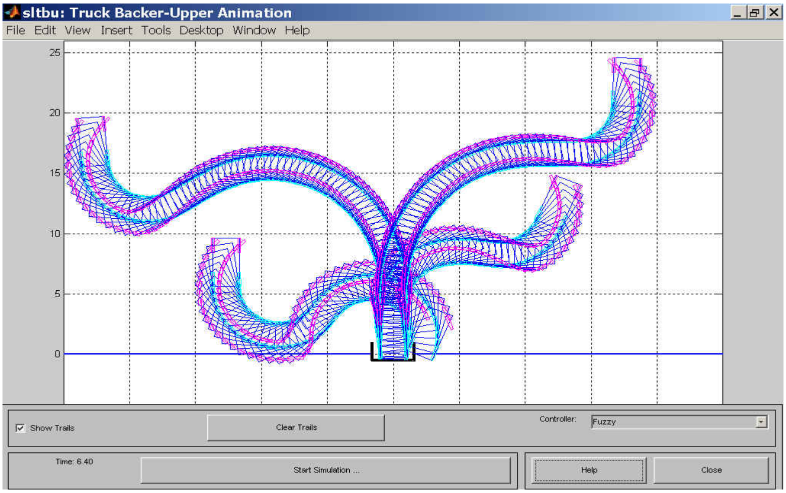
of six forecasting methods, where CANFIS and IT2FLS-TSK evaluate the best Mackey-Glass series forecasts respectively. The advantage of using the interval type-2 fuzzy logic forecasting method is that it obtains better results, even when data contains high levels of noise, furthermore we can use this method for better uncertainty series limits forecasting.

4.2 Shower Control Simulation

In figures 8 and 9 we compare the type-1 and type-2 fuzzy control results for the temperature and shower control simulation. The control variables signal of the interval type-2 fuzzy logic system show a better respond signal than the type-1 fuzzy logic system.

4.3 Truck Backer-Upper Control Simulation

In figures 10 and 11 we compare the type-1 and interval type-2 fuzzy control trajectories for the truck backer-upper control simulation. In the truck backer-upper

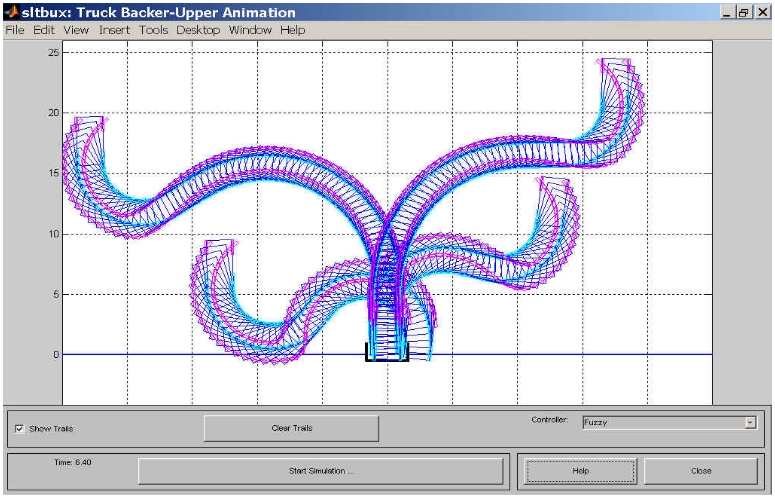

sltbu: Truck Backer-Upper Animation
File Edit View Insert Tools Desktop Window Help
25
20
15
10
5
0
Clear Trails
Start Simulation ...
Help
Close

☑ Show Trails

Controller: Fuzzy

Time: 6.40

Fig. 10. Trajectories obtained with the type-1 fuzzy control


sltbux: Truck Backer-Upper Animation
File Edit View Insert Tools Desktop Window Help
25
20
15
10
5
0
Clear Trails
Start Simulation ...
Help
Close

☑ Show Trails

Controller: Fuzzy

Time: 6.40

Fig. 11. Trajectories obtained with the interval type-2 fuzzy control

control using interval type-2 fuzzy logic, the trajectories are more stable and smooth than the type-1 fuzzy control.

5 Conclusions

The time series results show that intelligent hybrid methods and interval type-2 fuzzy models can be derived as a generalization of the autoregressive non-lineal models in the context of time series. This derivation allows a practical specification for a general class of prognosis and identification time series model, where a set of input-output variables are part of the dynamics of the time series knowledge base. This helps the application of the methodology to a series of diverse dynamics, with a very low number of causal variables to explain behavior. The results in the interval type-2 fuzzy control cases of the shower and truck backer upper have similar results to the type-1 fuzzy control with moderate uncertain footprints. To better characterize the interval type-2 fuzzy models we need to generate more case studies with better knowledge bases for the proposed problems, therefore classify the interval type-2 fuzzy model application strengths. The design and implementation done in the IT2FLS Toolbox is potentially important for research in the interval type-2 fuzzy logic area, thus solving complex problems on the different applied areas. Our future work is to improve the IT2FLS Toolbox with a better graphics user interface (GUI) and integrate a learning technique Toolbox to optimize the knowledge base parameters of the interval type-2 fuzzy inference system and design interval type-2 fuzzy neural network hybrid models.

References

1. Zadeh, L.A., “Fuzzy sets,” *Information and Control*, Vol. 8, pp. 338-353, 1965.
2. Zadeh, L.A., “Outline of a new approach to the analysis of complex systems and decision processes,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 1, pp. 28-44, Jan. 1973.
3. Zadeh, L.A., “The concept of a linguistic variable and its application to approximate reasoning, Parts 1, 2, and 3,” *Information Sciences*, 1975, 8:199-249, 8:301-357, 9:43-80.
4. Zadeh, L.A., “Fuzzy Logic,” *Computer*, Vol. 1, No. 4, pp. 83-93, 1988.
5. Zadeh, L.A., “Knowledge representation in fuzzy logic,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, pp. 89-100, 1989.
6. Karnik, N.N. and J.M. Mendel. An Introduction to Type-2 Fuzzy Logic Systems, Univ. of Southern Calif., Los Angeles, CA., June 1998b.
7. L. Zadeh, “Fuzzy logic = computing with words,” *IEEE Transactions on Fuzzy Systems*, vol. 2, pp. 103–111, 1996.
8. Q. Liang and J. Mendel, “Interval type-2 fuzzy logic systems: Theory and design,” *IEEE Transactions Fuzzy Systems*, vol. 8, pp. 535–550, 2000.
9. J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. NJ: Prentice-Hall, 2001.
10. Mizumoto, M. and K. Tanaka, “Some Properties of Fuzzy Sets of Type-2,” Information and Control, vol. 31, pp. 312-340, 1976.

11. Yager, R. R., “On the Implication Operator in Fuzzy logic,” Information Sciences, vol. 31, pp. 141-164, 1983.
12. Yager, R., “On a general class of fuzzy connectives,” *Fuzzy Sets and Systems*, 4:235-242, 1980.
13. Castillo and Melin. “A New Method for Adaptive control of Non-Linear Plants using Type-2 Fuzzy Logic and Neural Networks”, International Journal General Systems, 33 (2), pp.289-304.
14. Juan R. Castro. “Hybrid Intelligent Architecture Development for Time Series Forecasting”. Masters Degree Thesis. Tijuana Institute of Technology. December 2005.
15. Juan R. Castro. “Interval Type-2 Fuzzy Logic Toolbox”. Proceedings of International Seminar on Computational Intelligence, pp.100-108. Tijuana Institute of Technology. October 2006.

Evolutionary Computing for Topology Optimization of Type-2 Fuzzy Systems

Oscar Castillo, Alma Isabel Martinez, and Alma Cristina Martinez

Division of Graduate Studies, Tijuana Institute of Technology, Mexico

Abstract. We describe in this paper the use of hierarchical genetic algorithms for fuzzy system optimization in intelligent control. In particular, we consider the problem of optimizing the number of rules and membership functions using an evolutionary approach. The hierarchical genetic algorithm enables the optimization of the fuzzy system design for a particular application. We illustrate the approach with the case of intelligent control in a medical application. Simulation results for this application show that we are able to find an optimal set of rules and membership functions for the fuzzy system.

1 Introduction

We describe in this paper the application of a Hierarchical Genetic Algorithm (HGA) for fuzzy system optimization (Man et al. 1999). In particular, we consider the problem of finding the optimal set of rules and membership functions for a specific application (Yen and Langari 1999). The HGA is used to search for this optimal set of rules and membership functions, according to the data about the problem. We consider, as an illustration, the case of a fuzzy system for intelligent control.

Fuzzy systems are capable of handling complex, non-linear and sometimes mathematically intangible dynamic systems using simple solutions (Jang et al. 1997). Very often, fuzzy systems may provide a better performance than conventional non-fuzzy approaches with less development cost (Procyk and Mamdani 1979). However, to obtain an optimal set of fuzzy membership functions and rules is not an easy task. It requires time, experience and skills of the designer for the tedious fuzzy tuning exercise. In principle, there is no general rule or method for the fuzzy logic set-up, although a heuristic and iterative procedure for modifying the membership functions to improve performance has been proposed. Recently, many researchers have considered a number of intelligent schemes for the task of tuning the fuzzy system. The noticeable Neural Network (NN) approach (Jang and Sun 1995) and the Genetic Algorithm (GA) approach (Homaifar and McCormick 1995) to optimize either the membership functions or rules, have become a trend for fuzzy logic system development.

The HGA approach differs from the other techniques in that it has the ability to reach an optimal set of membership functions and rules without a known fuzzy system topology (Tang et al. 1998). During the optimization phase, the membership functions need not be fixed. Throughout the genetic operations (Holland 1975), a

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 63–75, 2007.
springerlink.com

reduced fuzzy system including the number of membership functions and fuzzy rules will be generated (Yoshikawa et al. 1996). The HGA approach has a number of advantages:

1. An optimal and the least number of membership functions and rules are obtained
2. No pre-fixed fuzzy structure is necessary, and
3. Simpler implementing procedures and less cost are involved.

We consider in this paper the case of automatic anesthesia control in human patients for testing the optimized fuzzy controller. We did have, as a reference, the best fuzzy controller that was developed for the automatic anesthesia control (Karr and Gentry 1993, Lozano 2003), and we consider the optimization of this controller using the HGA approach. After applying the genetic algorithm the number of fuzzy rules was reduced from 12 to 9 with a similar performance of the fuzzy controller. Of course, the parameters of the membership functions were also tuned by the genetic algorithm. We did compare the simulation results of the optimized fuzzy controllers obtained with the HGA against the best fuzzy controller that was obtained previously with expert knowledge, and control is achieved in a similar fashion. Since simulation results are similar, and the number of fuzzy rules was reduced, we can conclude that the HGA approach is a good alternative for designing fuzzy systems. We have to mention that Type-2 fuzzy systems are considered in this research work, which are more difficult to design and optimize.

2 Genetic Algorithms for Optimization

In this paper, we used a floating-point genetic algorithm (Castillo and Melin 2001) to adjust the parameter vector θ, specifically we used the Breeder Genetic Algorithm (BGA). The genetic algorithm is used to optimize the fuzzy system for control that will be described later (Castillo and Melin 2003). A BGA can be described by the following equation:

$$\mathrm{BGA}=(\mathrm{P}_g^{\,0},\ \mathrm{N},\ \mathrm{T},\ \Gamma,\Delta,\ \mathrm{HC},\ \mathrm{F},\ \mathrm{term}) \tag{1}$$

where: P_g^0=initial population, N=the size of the population, T=the truncation threshold, Γ=the recombination operator, Δ=the mutation operator, HC=the hill climbing method, F=the fitness function, $term$=the termination criterion.

The BGA uses a selection scheme called truncation selection. The %T best individuals are selected and mated randomly until the number of offspring is equal the size of the population. The offspring generation is equal to the size of the population. The offspring generation replaces the parent population. The best individual found so far will remain in the population. Self-mating is prohibited (Melin and Castillo 2002). As a recombination operator we used "extended intermediate recombination", defined as: If $x =(x_i,...x_n)$ *and* y $y=(y_1,...,y_n)$ are the parents, then the successor $z=(z_1,...,z_n)$ is calculated by:

$$z_i=x_i+\alpha_i(y_i-x_i) \qquad i=1,\ldots n \tag{2}$$

The mutation operator is defined as follows: A variable x_i is selected with probability p_m for mutation. The BGA normally uses $p_m = 1/n$. At least one variable will be mutated. A value out of the interval $[-range_i, range_i]$ is added to the variable. $range_i$ defines the mutation range. It is normally set to $(0.1\ x\ \ searchinterval_i)$. searchinterval$_i$ is the domain of definition for variable x_i. The new value z_i is computed according to

$$z_i=x_i \pm range_i \cdot \delta \tag{3}$$

The + or – sign is chosen with probability 0.5. δ is computed from a distribution which prefers small values. This is realized as follows

$$\delta = \sum_{i=0}^{15} \alpha_i 2^i \qquad \alpha_i \in 0,1 \tag{4}$$

Before mutation we set $\alpha_i=0$. Then each α_i is mutated to 1 with probability $p_\delta=1/16$. Only $\alpha_i=1$ contributes to the sum. On the average there will be just one α_i with value 1, say α_j. Then δ is given by

$$\delta = 2^{-j} \tag{5}$$

The standard BGA mutation operator is able to generate any point in the hypercube with center x defined by $x_i \pm range_i$. But it generates values much more often in the neighborhood of x. In the above standard setting, the mutation operator is able to locate the optimal x_i up to a precision of $ramge_i \cdot 2^{-150}$.

To monitor the convergence rate of the LMS algorithm, we computed a short term average of the squared error $e^2(n)$ using

$$ASE(m) = \frac{1}{K} \sum_{k=n+1}^{n+K} e^2(k) \tag{6}$$

where $m=n/K=1,2,\ldots$. The averaging interval K may be selected to be (approximately) K=10N. The effect of the choice of the step size parameter Δ on the convergence rate of LMS algorithm may be observed by monitoring the ASE(m).

2.1 Genetic Algorithm for Optimization

The proposed genetic algorithm is as follows:

1. We use real numbers as a genetic representation of the problem.
2. We initialize variable i with zero (i=0).
3. We create an initial random population P_i, in this case (P_0). Each individual of the population has n dimensions and, each coefficient of the fuzzy system corresponds to one dimension.
4. We calculate the normalized fitness of each individual of the population using linear scaling with displacement (Melin and Castillo 2002), in the following form:

$$f_i^{'} = f_i + \frac{1}{N} \sum |f_i| + \left| \min_i (f_i) \right| \qquad \forall i$$

5. We normalize the fitness of each individual using:

$$F_i = \frac{f_i'}{\sum_{i=1}^{N} f_i'} \qquad \forall i$$

6. We sort the individuals from greater to lower fitness.
7. We use the truncated selection method, selecting the %T best individuals, for example if there are 500 individuals and, then we select 0.30*500=150 individuals.
8. We apply random crossover, to the individuals in the population (the 150 best ones) with the goal of creating a new population (of 500 individuals). Crossover with it self is not allowed, and all the individuals have to participate. To perform this operation we apply the genetic operator of extended intermediate recombination as follows:

   If $x=(x_1,...,x_n)$ and $y=(y_1,...,y_n)$ are the parents, then the successors $z=(z_1,...,z_n)$ are calculated by, $z_i=x_i+\alpha_i(y_i-x_i)$ *for* $i=1,...,n$ where α is a scaling factor selected randomly in the interval $[-d,1+d]$. In intermediate recombination $d=0$, and for extended $d>0$, a good choice is $d=0.25$, which is the one that we used.
9. We apply the mutation genetic operator of BGA. In this case, we select an individual with probability $p_m=1/n$ (where n represents the working dimension, in this case n=25, which is the number of coefficients in the membership functions). The mutation operator calculates the new individuals z_i of the population in the following form: $z_i=x_i \pm range_i\ \delta$ we can note from this equation that we are actually adding to the original individual a value in the interval: $[-range_i, range_i]$ the range is defined as the search interval, which in this case is the domain of variable x_i, the sign ± is selected randomly with probability of 0.5, and is calculated using the following formula,

$$\delta = \sum_{i=0}^{m-1} \alpha_i 2^{-i} \qquad \alpha_i \in 0,1$$

   Common used values in this equation are $m=16$ y $m=20$. Before mutation we initiate with $\alpha_i=0$, then for each α_i we mutate to 1 with probability $p_\delta=1/m$.
10. Let i=i+1, and continue with step 4.

3 Evolution of Fuzzy Systems

Ever since the very first introduction of the fundamental concept of fuzzy logic by Zadeh in 1973, its use in engineering disciplines has been widely studied. Its main attraction undoubtedly lies in the unique characteristics that fuzzy logic systems possess. They are capable of handling complex, non-linear dynamic systems using simple solutions. Very often, fuzzy systems provide a better performance than conventional non-fuzzy approaches with less development cost.

However, to obtain an optimal set of fuzzy membership functions and rules is not an easy task. It requires time, experience, and skills of the operator for the tedious fuzzy tuning exercise. In principle, there is no general rule or method for the fuzzy logic set-up. Recently, many researchers have considered a number of intelligent techniques for the task of tuning the fuzzy set. Here, another innovative scheme is described (Tang et al. 1998). This approach has the ability to reach an optimal set of membership functions and rules without a known overall fuzzy set topology. The conceptual idea of this approach is to have an automatic and intelligent scheme to tune the membership functions and rules, in which the conventional closed loop fuzzy control strategy remains unchanged, as indicated in Figure 1.

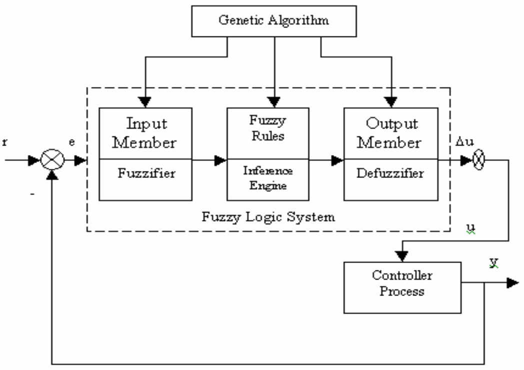


Fig. 1. Genetic algorithm for a fuzzy control system

In this case, the chromosome of a particular system is shown in Figure 2. The chromosome consists of two types of genes, the control genes and parameter genes. The control genes, in the form of bits, determine the membership function activation, whereas the parameter genes are in the form of real numbers to represent the membership functions.

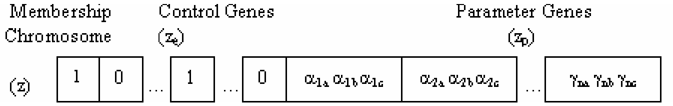

Membership Chromosome
Control Genes
(z_c)
Parameter Genes
(z_p)
(z)
1
0
...
1
...
0
$\alpha_{1a}\ \alpha_{1b}\ \alpha_{1c}$
$\alpha_{2a}\ \alpha_{2b}\ \alpha_{2c}$
...
$\gamma_{na}\ \gamma_{nb}\ \gamma_{nc}$

Fig. 2. Chromosome structure for the fuzzy system

To obtain a complete design for the fuzzy control system, an appropriate set of fuzzy rules is required to ensure system performance. At this point it should be stressed that the introduction of the control genes is done to govern the number of fuzzy subsets in the system. Once the formulation of the chromosome has been set for the fuzzy membership functions and rules, the genetic operation cycle can be performed. This cycle of operation for the fuzzy control system optimization using a genetic algorithm is illustrated in Figure 3. There are two population pools, one for storing the membership chromosomes and the other for storing the fuzzy rule chromosomes. We can see this in Figure 3 as the membership population and fuzzy rule population, respectively. Considering that there are various types of gene structure, a number of different genetic operations can be used. For the crossover operation, a one-point crossover is applied separately for both the control and parameter genes of the membership chromosomes within certain operation rates. There is no crossover operation for fuzzy rule chromosomes since only one suitable rule set can be assisted.

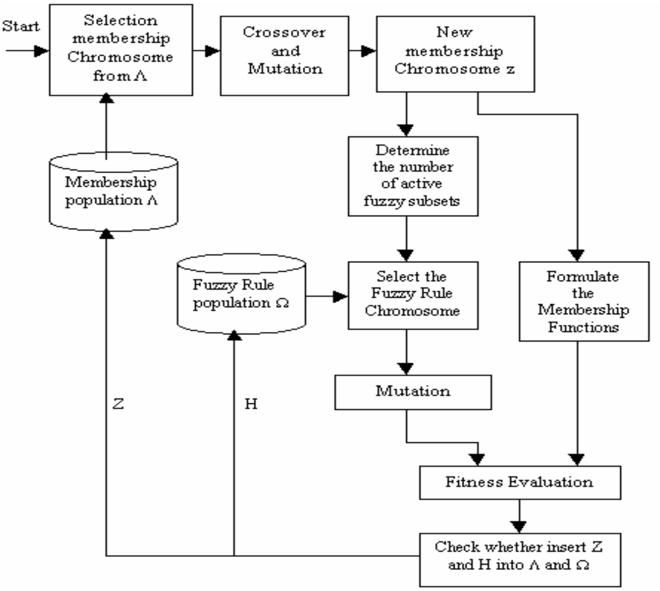

Start
Selection membership Chromosome from Λ
Crossover and Mutation
New membership Chromosome z
Membership population Λ
Determine the number of active fuzzy subsets
Fuzzy Rule population Ω
Select the Fuzzy Rule Chromosome
Formulate the Membership Functions
Mutation
Z
H
Fitness Evaluation
Check whether insert Z and H into Λ and Ω

Fig. 3. Genetic cycle for fuzzy system optimization

Bit mutation is applied for the control genes of the membership chromosome. Each bit of the control gene is flipped if a probability test is satisfied (a randomly generated number is smaller than a predefined rate). As for the parameter genes, which are real number represented, random mutation is applied.

The fitness function can be defined in this case as follows:

$$f_i = \Sigma / y(k) - r(k) / \tag{7}$$

where Σ indicates the sum for all the data points in the training set, and y(k) represents the real output of the fuzzy system and r(k) is the reference output. This fitness value measures how well the fuzzy system is approximating the real data of the problem.

4 Type-2 Fuzzy Logic

The concept of a type-2 fuzzy set, was introduced by Zadeh (Melin and Castillo 2002) as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set (Castillo and Melin 2001, Melin and Castillo 2002) where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets (Castillo and Melin 2001). When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2.

Example: Consider the case of a fuzzy set characterized by a Gaussian membership function with mean m and a standard deviation that can take values in $[\sigma_1,\sigma_2]$, i.e.,

$$\mu(x)=\exp\{-½[(x-m)/\sigma]^2\}; \quad \sigma \in [\sigma_1,\sigma_2] \tag{8}$$

Corresponding to each value of σ, we will get a different membership curve (Figure 4). So, the membership grade of any particular x (except x=m) can take any of a number of possible values depending upon the value of σ, i.e., the membership grade is not a crisp number, it is a fuzzy set. Figure 4 shows the domain of the fuzzy set associated with x=0.7.

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n (Castillo and Melin 2003). A higher-type number just indicates a higher “degree of fuzziness”. Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change; however, the basic principles of fuzzy logic are independent of the nature of membership functions and hence, do not change. In Figure 5 we show the general structure of a type-2 fuzzy system. We assume that both antecedent and consequent sets are type-2; however, this need not necessarily be the case in practice.

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output. Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation “type reduction” and call the type-1 fuzzy set so obtained a “type-reduced set”. The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number. Type-2 sets can be used to convey the uncertainties in membership functions of type-1 fuzzy sets, due to the dependence of the membership functions on available linguistic and numerical information.

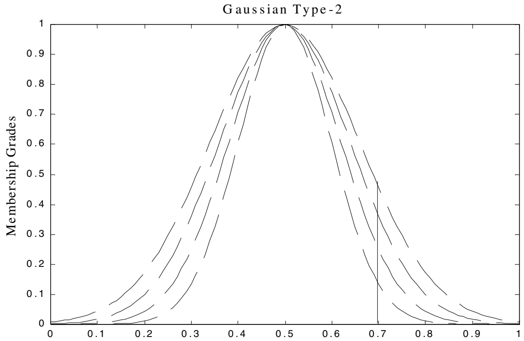


Fig. 4. A type-2 fuzzy set representing a type-1 set with uncertain deviation

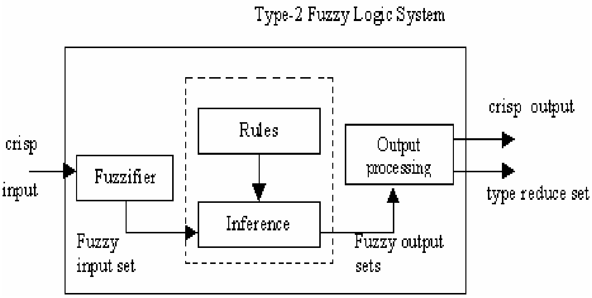


Fig. 5. Structure of a type-2 fuzzy system

5 Application to Intelligent Control

We consider the case of controlling the anesthesia given to a patient as the problem for finding the optimal fuzzy system for control (Lozano 2003). The complete implementation was done in the MATLAB programming language. The fuzzy systems were build automatically by using the Fuzzy Logic Toolbox, and genetic algorithm was coded directly in the MATLAB language. The fuzzy systems for control are the individuals used in the genetic algorithm, and these are evaluated by comparing them to the ideal control given by the experts. In other words, we compare the performance of the fuzzy systems that are generated by the genetic algorithm, against the ideal control system given by the experts in this application.

5.1 Anesthesia Control Using Fuzzy Logic

The main task of the anesthesist, during and operation, is to control anesthesia concentration. In any case, anesthesia concentration can’t be measured directly. For this reason, the anesthesist uses indirect information, like the heartbeat, pressure, and motor activity. The anesthesia concentration is controlled using a medicine, which can be given by a shot or by a mix of gases. We consider here the use of isoflurance, which is usually given in a concentration of 0 to 2% with oxygen. In Figure 6 we show a block diagram of the controller.

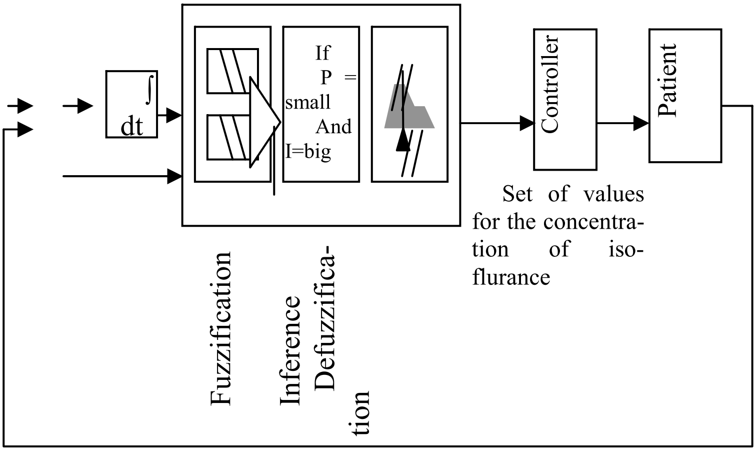


Fig. 6. Architecture of the fuzzy control system

The air that is exhaled by the patient contains a specific concentration of isoflurance, and it is re-circulated to the patient. As consequence, we can measure isoflurance concentration on the inhaled and exhaled air by the patient, to estimate isoflurance concentration on the patient's blood. From the control engineering point of view, the task by the anesthesist is to maintain anesthesia concentration between the high level W (threshold to wake up) and the low level E (threshold to success). These levels are difficult to be determine in a changing environment and also are dependent on the patient's condition. For this reason, it is important to automate this anesthesia control, to perform this task more efficiently and accurately, and also to free the anesthesist from this time consuming job. The anesthesist can then concentrate in doing other task during operation of a patient.

The first automated system for anesthesia control was developed using a PID controller in the 60's. However, this system was not very succesful due to the non-linear nature of the problem of anesthesia control. After this first attempt, adaptive control was proposed to automate anesthesia control, but robustness was the problem in this case. For these reasons, fuzzy logic was proposed for solving this problem.

5.2 Characteristics of the Fuzzy Controller

In this section we describe the main characteristics of the fuzzy controller for anesthesia control. We will define input and output variable of the fuzzy system. Also, the fuzzy rules of fuzzy controller previously designed will be described.

The fuzzy system is defined as follows:

1. Input variables: Blood pressure and Error
2. Output variable: Isoflurance concentration
3. Nine fuzzy if-then rules of the optimized system, which is the base for comparison

4. 12 fuzzy if-then rules of an initial system to begin the optimization cycle of the genetic algorithm.

   The linguistic values used in the fuzzy rules are the following:

   PB = Positive Big
   PS = Positive Small
   ZERO = zero
   NB =Negative Big
   NS = Negative Small

We show below a sample set of fuzzy rules that are used in the fuzzy inference system that is represented in the genetic algorithm for optimization.

if Blood pressure is NB and error is NB then conc_isoflurance is PS
if Blood pressures is PS then conc_isoflurance is NS
if Blood pressure is NB then conc_isoflurance is PB
if Blood pressure is PB then conc_isoflurance is NB
if Blood pressure is ZERO and error is ZERO then conc_isoflurance is ZERO
if Blood pressure is ZERO and error is PS then conc_isoflurance is NS
if Blood pressure is ZERO and error is NS then conc_isoflurance is PS
if error is NB then conc_isoflurance is PB
if error is PB then conc_isoflurance is NB
if error is PS then conc_isoflurance is NS
if Blood pressure is NS and error is ZERO then conc_isoflurance is NB
if Blood pressure is PS and error is ZERO then conc_isoflurance is PS.

5.3 Genetic Algorithm Specification

The general characteristics of the genetic algorithm that was used are the following:

NIND = 40; % Number of individuals in each subpopulation.
MAXGEN = 300; % Maximum number of generations allowed.
GGAP = .6; %"Generational gap", which is the percentage from the complete population of new individuals generated in each generation.
PRECI = 120; % Precision of binary representations.
SelCh = select('rws', Chrom, FitnV, GGAP); % Roulette wheel method for selecting the indivuals participating in the genetic operations.
SelCh = recombin('xovmp',SelCh,0.7); % Multi-point crossover as recombination method for the selected individuals.
ObjV = FuncionObjDifuso120_555(Chrom, sdifuso); Objective function is given by the error between the performance of the ideal control system given by the experts and the fuzzy control system given by the genetic algorithm.

5.4 Representation of the Chromosome

In Table 1 we show the chromosome representation, which has 120 binary positions. These positions are divided in two parts, the first one indicates the number of rules of the fuzzy inference system, and the second one is divided again into fuzzy rules to indicate which membership functions are active or inactive for the corresponding rule.

Table 1. Binary Chromosome Representation

Bit assigned	Representation
1 a 12	Which rule is active or inactive
13 a 21	Membership functions active or inactive of rule 1
22 a 30	Membership functions active or inactive of rule 2
...	Membership functions active or inactive of rule...
112 a 120	Membership functions active or inactive of rule 12

6 Simulation Results

We describe in this section the simulation results that were achieved using the hierarchical genetic algorithm for the optimization of the fuzzy control system, for the case of anesthesia control. The genetic algorithm is able to evolve the topology of the fuzzy system for the particular application. We used 300 generations of 40 individuals each to achieve the minimum error. We show in Figure 7 the final results of the genetic algorithm, where the error has been minimized. This is the case in which only nine fuzzy rules are needed for the fuzzy controller. The value of the minimum error achieved with this particular fuzzy logic controller was of 0.0064064, which is considered a small number in this application.

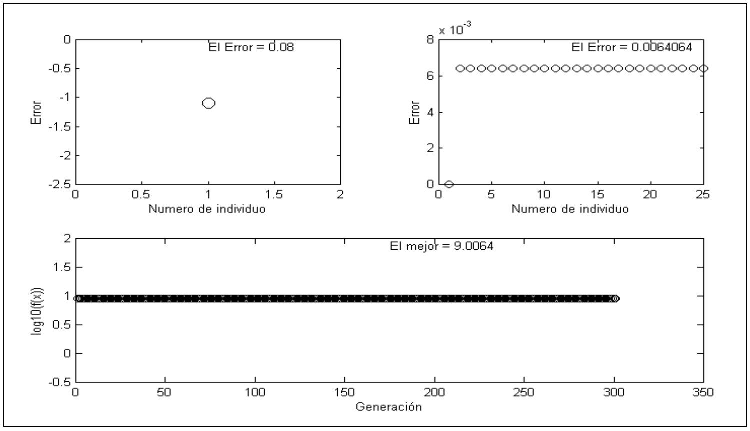


Fig. 7. Plot of the error after 300 generations of the HGA

In Figure 8 we show the simulation results of the fuzzy logic controller produced by the genetic algorithm after evolution. We used a sinusoidal input signal with unit amplitude and a frequency of 2 radians/second, with a transfer function of [1/(0.5s +1)]. In this figure we can appreciate the comparison of the outputs of both the ideal controller (1) and the fuzzy controller optimized by the genetic algorithm (2). From this figure it is clear that both controllers are very similar and as a consequence we can conclude that the genetic algorithm was able to optimize the performance of the fuzzy logic controller. We can also appreciate this fact more clearly in Figure 9, where we have amplified the simulation results from Figure 8 for a better view.

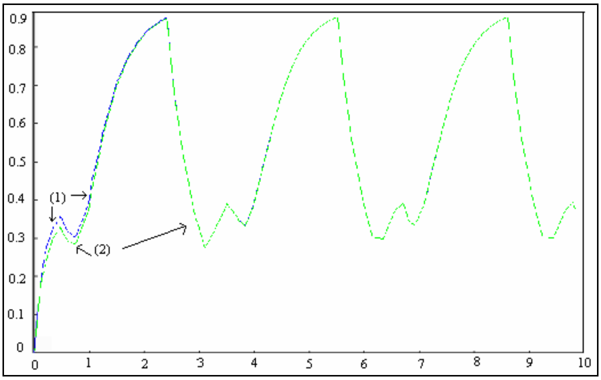


Fig. 8. Comparison between outputs of the ideal controller (1) and the fuzzy controller produced with the HGA (2)

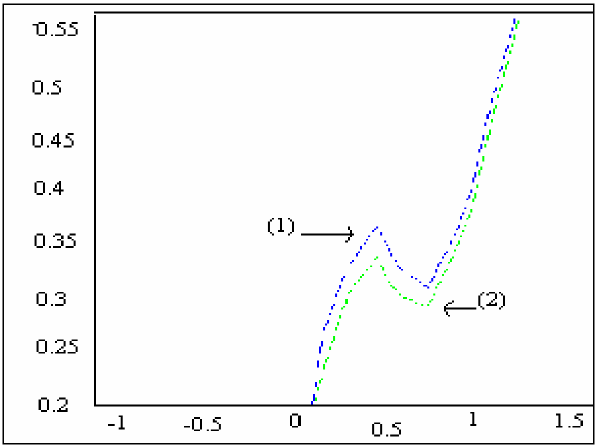


Fig. 9. Zoom in of figure 8 to view in more detail the difference between the controllers

7 Conclusions

We consider in this paper the case of automatic anesthesia control in human patients for testing the optimized fuzzy controller. We did have, as a reference, the best fuzzy controller that was developed for the automatic anesthesia control (Karr and Gentry 1993, Lozano 2003), and we consider the optimization of this controller using the HGA approach. After applying the genetic algorithm the number of fuzzy rules was reduced from 12 to 9 with a similar performance of the fuzzy controller. Of course, the parameters of the membership functions were also tuned by the genetic algorithm. We did compare the simulation results of the optimized fuzzy controllers obtained with the HGA against the best fuzzy controller that was obtained previously with expert knowledge, and control is achieved in a similar fashion.

Acknowledgments

We would like to thank the Research Grant Committee of COSNET for the financial support given to this project. We would also like to thank CONACYT for the scholarships given to the students that work in this research project (Alma Cristina Martinez and Alma Isabel Martinez).

References

O. Castillo and P. Melin (2001), “Soft Computing for Control of Non-Linear Dynamical Systems”, Springer-Verlag, Heidelberg, Germany.

O. Castillo and P. Melin (2003), “Soft Computing and Fractal Theory for Intelligent Manufacturing”, Springer-Verlag, Heidelberg, Germany.

J. Holland, (1975), "Adaptation in natural and artificial systems" (University of Michigan Press).

A. Homaifar and E. McCormick (1995), “Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms”, *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 129-139.

J.-S. R. Jang and C.-T. Sun (1995) “Neurofuzzy fuzzy modeling and control”, *Proc. IEEE*, vol. 83, pp. 378-406.

J.-S. R. Jang, C.-T. Sun, and E. Mizutani (1997), "Neuro-fuzzy and Soft Computing, A computational approach to learning and machine intelligence", , Prentice Hall, Upper Saddle River, NJ.

C.L. Karr and E.J. Gentry (1993), “Fuzzy control of pH using genetic algorithms”, *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 46-53.

A. Lozano (2003), "Optimización de un Sistema de Control Difuso por medio de algoritmos genéticos jerarquicos", Thesis, Dept. of Computer Science, Tijuana Institute of Technology, Mexico.

K.F. Man, K.S. Tang, and S. Kwong (1999), "Genetic Algorithms: Concepts and Designs", Springer Verlag.

P. Melin and O. Castillo (2002), “Modelling, Simulation and Control of Non-Linear Dynamical Systems”, Taylor and Francis, London, Great Britain.

T.J. Procyk and E.M. Mamdani (1979), “A linguistic self-organizing process controller” *Automatica*, vol. 15, no. 1, pp 15-30.

K.-S. Tang, K.-F. Man, Z.-F. Liu and S. Kwong (1998), “Minimal fuzzy memberships and rules using hierarchical genetic algorithms”, *IEEE Trans. on Industrial Electronics*, vol. 45, no. 1.

J. Yen, and R. Langari (1999), "Fuzzy Logic: intelligence, control and information", Prentice Hall, Inc.

T. Yoshikawa, T. Furuhashi and Y. Uchikawa (1996), “Emergence of effective fuzzy rules for controlling mobile robots using NA coding method”, *Proc. ICEC’96*, Nagoya, Japan, pp. 581

Part II

Fuzzy Clustering: Theory and Applications

Comparison of the Performance of Seven Classifiers as Effective Decision Support Tools for the Cytodiagnosis of Breast Cancer: A Case Study

Nicandro Cruz-Ramírez[1], Héctor-Gabriel Acosta-Mesa[1], Humberto Carrillo-Calvet[2], and Rocío-Erandi Barrientos-Martínez[1]

[1] Facultad de Física e Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho # 5, Col. Centro, C.P. 91000, Xalapa, Veracruz, México
{ncruz, heacosta}@uv.mx, erandi_bm@yahoo.com.mx
[2] Facultad de Ciencias, Universidad Nacional Autónoma de México, Circuito Exterior Ciudad Universitaria, México, D.F.
carr@servidor.unam.mx

Abstract. We evaluate the performance of seven classifiers as effective potential decision support tools in the cytodiagnosis of breast cancer. To this end, we use a real-world database containing 692 fine needle aspiration of the breast lesion cases collected by a single observer. The results show, in average, good overall classification performance in terms of five different tests: accuracy of 93.62%, sensitivity of 89.37%, specificity of 96%, PV+ of 92% and PV- of 94.5%. With this comparison, we identify and discuss the advantages and disadvantages of each of these approaches. Finally, based on these results, we give some advice regarding the selection on the classifier depending on the user's needs.

1 Introduction

The cytodiagnosis of breast cancer, using a technique called fine needle aspiration of the breast lesion (FNAB) [2], involves a process where a syringe sucks cells from breast lesions using a fine bore needle similar to those used for blood samples. Once this is done, these cells are transferred to a transport solution and sent to the pathology laboratory for a microscopic study carried out by a trained cytopathologist [2]. It is important to mention that the time it normally takes to a medical doctor to become an independent practising cytopathologist is about 5 years as a minimum. This fact can give an indication of the very complex learning process which medical doctors have to pass through. It is mainly for this reason that machine learning methods for decision-making may have two potential applications: to accelerate the training process of learning by providing guidance to the trainee on what features are the most important ones to look at; and to compare the final results to those of the trainee or even of the expert so that the decision (whether the sample taken indicates a benign or a malignant output) can be made on more robust criteria (qualitative and quantitative). The cytodiagnosis of breast cancer has a special requirement: it is mandatory to avoid the diagnosis of false positives since this will lead to the surgical removal of healthy breast(s); a situation that has to be always avoided. Such a requirement imposes an

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 79–87, 2007.
springerlink.com

additional challenge to automatic classifiers for they have to reach, as the pathologist, a specificity value close to 100%.

In a series of studies by Cross et al. [2-4], they present the classification performance of logistic regression, neural networks and decision trees on the same database we use here. In this paper, we extend those studies by including the Bayesian network framework since this approach gives complementary features to those provided by the mentioned classifiers. So, we identify more deeply the advantages and disadvantages of each classifier in order to study the potential of each method as effective support tools in the diagnosis of breast cancer.

The remainder of this paper is organized as follows. In section 2, we briefly describe the materials and methods used for the experiments reported here. In section 3, we present the experimental results of these 7 classifiers. In section 4, we discuss these results focusing on the advantages and disadvantages of each approach. Finally, in section 5, we give a conclusion and propose some future work.

2 Materials and Methods

2.1 The Dataset

The real-world database for this study comes from the field of cytodiagnosis of breast cancer using a technique called fine needle aspiration of the breast lesion (FNAB) [2-4], which is the most common confirmatory method used in the United Kingdom for this purpose. The database contains 692 consecutive specimens of FNAB received at the Department of Pathology, Royal Hallamshire Hospital in Sheffield, UK, during 1992-1993 [2, 3]. Eleven independent variables and one dependent variable form part of this dataset (see table 1). All these variables, except age, are dichotomous taking the values of present or absent indicating the presence or absence of a diagnostic feature respectively. Variable age was sorted into three different categories: 1 (up to 50 years old), 2 (51 to 70 years old) and 3 (above 70 years old). The dependent variable outcome can take on two different values: benign or malignant. In the case of a malignant outcome, such a result was confirmed by a biopsy (where available). In the case of a benign case, this result was confirmed by mammographic findings (where available) and by absence of further malignant specimens. These variables appear in the literature as being discriminatory in the cytodiagnosis of FNAB [2-4].

2.2 The Classifiers

In a classification task, given a set of unlabelled cases on the one hand, and a set of labels on the other, the problem to solve is to find a function that suitably maps each unlabelled instance to its corresponding label (class). The central research interest in this specific area is the design of automatic classifiers that can estimate this function from data. The efforts of such a research have resulted in different classification methods: regression, decision trees, Bayesian networks and neural networks, among others [9]. For the sake of brevity, we do not provide the details of any of the 7 classifiers that we present here. Instead, we only briefly describe them and refer the reader to their representative sources.

Table 1. The defined human observations used as input variables

Observed Feature	Definition
'cellular dyshesion'	True if the majority of epithelial cells are dyhesive, false if the majority of epithelial cells are in cohesive groups
'intracytoplasmic lumina'	True if 'intracytoplasmic lumina' are present in some epithelial cells, false if absent
"Three-dimensionality" of epithelial cell clusters	True if some clusters of epithelial cells are not flat (more than two nuclei thick) and this is not due to artefactual folding, false if all clusters of epithelial cells are flat
Bipolar "naked" nuclei	True if bipolar "naked" nuclei are present, false if absent
'foamy macrophages'	True if 'foamy macrophages' are present, false if absent
Nucleoli	True if more than three easily-visible nucleoli are present in some epithelial cells, false if three or fewer easily-visible nucleoli in all epithelial cells
Nuclear pleiomorphism	True if some epithelial cells have nuclear diameters twice that of other epithelial cell nuclei, false if no epithelial cell nuclei have diameters twice that of other epithelial cell nuclei
''nuclear size''	True if some epithelial cell nuclei have diameters twice that of red blood cell diameters, false if all epithelial cell nuclei have diameters less than twice that of red blood cell diameters
'necrotic epithelial cells'	True if 'necrotic epithelial cells' are present, false if absent
'apocrine change'	True if the majority of epithelial cell nuclei show 'apocrine change', false if 'apocrine change' is not present in the majority of epithelial cells

1. Logistic regression is a classic statistic technique for classification. The logistic equation for this study was built from the training set (432 cases) considering all variables in a main effects only model. For details, the reader is referred to [2].
2. C4.5 is a well-known algorithm for learning decision trees from data based on entropy measures. The importance of such procedure is its intuitive interpretation and that decision rules can be easily extracted from it. The classic reference is [7].
3. The Multilayer perceptron neural networks (MLP) used here has 11 input neurons, 1 hidden layer with 6 neurons and 1 output neuron. For more details on the architecture of the network, the reader is referred to [2]. The MLP reported here was tested with 14 and 50 training epochs.
4. In contrast to MLP, Fuzzy adaptive resonance theory mapping neural networks (ARTMAP) have only one adjustable parameter: the vigilance factor. From these networks it is possible to extract IF-THEN rules. The details of the network used here are in [2].

5. PC is an algorithm for constructing Bayesian network (BN) structures from data [8]. It uses the G^2 statistics to carry out conditional independence tests for deciding the deletion of an arc between two nodes (random variables). For more details on PC, the reader is referred to [8].
6. CBL2 is a constraint-based algorithm to build BN structures from data that uses mutual information and conditional mutual information tests to decide when to connect/disconnect a pair of nodes [1].
7. MP-Bayes is a constraint-based algorithm to build BN structures from data which uses mutual information and conditional mutual information measures, combined with the statistics T, to add/delete an arc between a pair of nodes [5]. It has been explicitly designed for behaving parsimoniously: it tries to build Bayesian network structures with the least number of arcs.

3 Experimental Methodology and Results

The breast cancer dataset (692 cases) was randomly partitioned into 462 cases for training and the remaining 230 cases for testing. Table 2 shows the results of the experiments carried out here.

Table 2. Results of accuracy, sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by the pathologist, logistic regression, C4.5, MLP (after 14 training epochs), MLP (after 50 training epochs), ARTMAP, PC, CBL2 and MP-Bayes for the holdout method. 95% confidence intervals are shown in parentheses.

Classifier	Accuracy	Sensitivity	Specificity	PV+	PV-
Pathologist	94 ± 1.56	82 (77-87)	100	100	92 (89-94)
Logistic Regression	95 ± 1.43	94 (89-99)	95 (90-97)	87 (80-95)	97 (95-99)
C4.5	94 ± 1.56	95 (89-99)	93 (90-97)	87 (80-95)	98 (95-99)
MLP (14 epochs)	95 ± 1.43	85 (76-93)	100	100	93 (90-97)
MLP (50 epochs)	94 ± 1.56	88 (80-95)	98 (96-99)	95 (90-99)	95 (91-98)
ARTMAP	94 ± 1.56	90 (84-96)	96 (93-99)	94 (89-99)	94 (90-98)
PC	93 ± 1.68	87 (79-94)	97 (94-100)	94 (88-99)	93 (89-97)
CBL2	92 ± 1.78	88 (81-95)	95 (91-98)	90 (84-97)	93 (89-97)
MP-Bayes	92 ± 1.78	88 (81-95)	94 (90-98)	89 (82-96)	93 (89-97)

4 Discussion

According to the medical literature, all ten defined observations mentioned in section 2.1 are considered relevant for the cytodiagnosis of breast cancer [2-4]. Furthermore, age is also considered relevant as it provides useful information for making the final diagnosis. Let us present now the pros and cons of each approach including that of the human expert.

4.1 Human Performance

Table 2 shows that the requirement of not having any false positives diagnosed (specificity) is completely achieved by the human expert. However, some inconsistencies (same values for a determined set of variables result in different diagnoses) have been found in the 692-case dataset used in this study: when a variable has an ambiguous value, the expert may then be biased by their knowledge about an earlier decision or is taking into account more information than that portrayed in the cytological samples. Therefore, the observed values are not strictly independent. The process or processes that cytopathologists follow to make their final diagnoses have not been yet fully understood and can only be partially explained in terms of pattern recognition with occasional use of heuristic logic [2]. All the features coded in the breast cancer dataset used in the present study were made by the expert cytopathologist, who carried out most of the processing that is probably required to solve the diagnostic problem. Hence, the information provided in such a database is subjective rather than objective. To ameliorate this problem, alternative data collection methods such as image analysis techniques could be used so that objective measures from sample raw digitalized images can be extracted. This exploration is left as future work.

4.2 Logistic Regression

Logistic regression is considered as the standard statistical technique to which the performance of other classifiers should be compared [2]. In the results presented in [2], the main variables to predict the malignancy of the outcome are increasing age, presence of ‘intracytoplasmic lumina’, presence of ‘three dimensional cell clusters’, nucleoli and ‘nuclear pleomorphism’. Furthermore, logistic regression identifies one variable as being relevant for predicting a benign diagnosis: ‘apocrine change’. Regarding the performance of logistic regression, it is well known that regression models often have unstable estimates when the number of regressors is enlarged [8] and that are variable-ordering sensitive [8]. Logistic regression shows the best performance in only 1 (with a draw with MLP) out of 5 tests. These results suggest that it is convenient to select a fewer number of regressors and check the best variable ordering for obtaining good classification estimates.

4.3 Decision Trees: C4.5

The well-known C4.5 program [7] is considered one of the state-of-the-art classifiers. Because of the lack of space, we do not draw the decision tree derived from the 462-case dataset (but it can be consulted in [3]). In such a tree, the main variables chosen by algorithm C4.5 are 7 out of 11: ‘nuclear size’, ‘intracytoplasmic lumina’, ‘apocrine change’, nucleoli, ‘bipolar naked nuclei’, age and ‘three-dimensionality of epithelial cell clusters’. Unfortunately, it does not show the interactions among the independent variables, which may help the expert know more about the nature of the phenomenon. As can be seen from table 2, this selection of variables appears to be better for obtaining good sensitivity and PV- results more than specificity and PV+ results. We cannot forget that it is the values of specificity and PV+ that matter the most. Hence, given these results, we suggest that, if C4.5 is to be used in this domain, we should keep always in mind this behavior.

4.4 MLP Neural Networks

The third automatic method in this comparison is the well-known framework of multilayer perceptron (MLP) neural networks [2]. Looking at these results, there is a decrease in both the specificity and PV+ for the case of 50 epochs, which significantly degrades the excellent performance of such a classifier when using 14 epochs. Although the performance of this method, either using 14 or 50 epochs, generally overcomes that of the remaining classifiers (in the specificity and PV+ tests), there are a number of important points to discuss here about this technique. A desirable decision support system in a medical domain has to be able to provide a good degree of explanation about its decisions. The intrinsic nature of MLP makes them a "black box", where the intermediate calculations are not visible to the external user but only the output is visible. MLP can be extremely accurate in their classification but they lack the power of inducing comprehensible structures that might help one understand the domain better, which in turn might help create new knowledge. MLP also require a number of parameters to be tuned so as they can work properly: the number of hidden layers and the number of neurons in each hidden layer. The best parameter selection that defines any neural network has in general to be determined empirically; i.e., there exist no mathematically proven procedures to choose the optimal architecture for this MLP [2]. Regarding the number of hidden layers in an MLP, it has been proven that using a bigger number of them can represent some functions that cannot be represented by using, say, two hidden layers. However, increasing the number of such layers does not always solve this problem so that some functions can still go unrepresented [6]. Hence, this kind of model cannot generalize outside the training space. This is an important limitation that should be taken into account when using this kind of classifier. Furthermore, as in the cases of logistic regression and decision trees, MLP do not represent the interactions among the input variables, stopping the user from knowing more about the problem under study.

4.5 ARTMAP Neural Networks

The limitations of the MLP approach, such as the number of adjustable parameters and the difficulty in presenting transparent explanations of the results seem to be eliminated by the ARTMAP methodology because of its architecture: it does not contain hidden layers with implicit meaning [2]. Also, ARTMAP neural networks only have one adjustable parameter, called the vigilance factor, which reduces the complexity of having a combinatory explosion to choose the best parameters in an MLP. Therefore, the optimization of the ARTMAP network is simpler. With respect to the second limitation, it is possible to extract automatically a set of rules similar to those used in rule-based expert systems so that these rules can give a good account of why the ARTMAP reached a certain conclusion. However, in comparison with the results of specificity and PV+ given by MLP either using 14 epochs or 50 epochs, ARTMAP's results of specificity and PV+ are less accurate. Hence, although ARTMAP neural networks can overcome, in general, the intrinsic problems of MLP neural networks, they still show a poorer performance than that of MLP in terms of classification accuracy in this specific domain.

Regarding the problem of the MLP about their limitation to generalize outside the training space, ARTMAP neural nets could possibly have the potential to cope with that problem in a simple and beautiful way: if a case can be instantiated significantly enough to a predefined category then this case is assigned to such a category, otherwise a new category node is created to classify that case. The problem with this approach is that if the value of the threshold is set very high, then there may be many new categories leading to a poor compression of the data, poor explanation of the output and, overall, a poor classification performance. Another related problem about the formation of new categories comes when the data items, with which the ARTMAP is fed, are presented in different order. A possible solution to this problem is to use something known as voting strategy: we need to train a certain number of ARTMAP neural networks with different item presentation orders and take the final prediction from the majority decision of all these networks.

4.6 PC, CBL2 and MP-Bayes

Because of the lack of space, we cannot show here the Bayesian network structures of these 3 procedures. However, we can summarize their results: MP-Bayes, PC and CBL2 share three variables to explain the outcome: age, ‘intracytoplasmic lumina’, nucleoli and ‘nuclear size’. Moreover, PC and CBL2 have also two more variables in common to explain the class variable: ‘cellular dyshesion’ and the ‘three-dimensionality’ of epithelial cells clusters. PC also considers variable ‘apocrine change’ as being relevant to explain the outcome whereas CBL2 considers variable ‘necrotic epithelial cells’ as being relevant for the prediction of the output. The best overall performance among these three procedures is that of PC. Compared to the performance of MP-Bayes, it seems that the three variables that are not shared by both procedures are the key to produce much better results in PC for specificity and PV+. In comparison to CBL2, it seems that variable ‘apocrine change’ causes the difference in PC to produce better results in these same tests. However, the selection of fewer variables by this procedure does not seem to produce significant worse results than those by procedures PC and CBL2. We argue here that the parsimonious nature of MP-Bayes results suitable for obtaining a good trade-off between accuracy and complexity. Furthermore, one important thing to note is that this kind of classifiers does indeed allow not only to observe the interactions between the independent variables and the dependent one but also the interaction among the independent variables themselves, which potentially permits one to understand more deeply the phenomenon under study.

5 Conclusions and Future Work

We have presented the performance of 7 classifiers as effective decision support tools for the cytodiagnosis of breast cancer. The unusual requirement of having 100% of specificity (no false positives) makes this particular medical domain a challenging area where decision-support systems can be tested to the extreme. Although the overall classification results given by all procedures have proved accurate, there is still much work to do regarding the refinement of such algorithms to nearly achieve the required 100% of specificity (with exception of MLP with 14 epochs). This can be due to the fact that cytopathologists are forced to represent continuous features in a dichotomized way. A relevant exploration would be that of the possibility of codifying such characteristics with more power and richness than that of a binary coding. To do so, it would be necessary to extend some of these classifiers' capabilities so that they could deal with both continuous and discrete variables or to simply allow pathologists to codify the features using a bigger range of possible values for each variable. Based on the results presented here, if a pathologist requires an excellent classifier regardless how it reaches its conclusions, the choice is an MLP neural network. If the pathologist needs a classifier capable of giving an easy explanation of how it arrives to its conclusions, the choice can be logistic regression, decision trees, ARTMAP neural networks or Bayesian networks. But if the pathologist needs to visualize the interactions among attributes (observed features) so that she can understand more deeply the phenomenon under investigation, then the choice is a Bayesian network. We have identified a significant ingredient of subjectivity when two or more pathologists look at the same sample: they just do not agree sometimes in their diagnoses. Thus, another interesting exploration is that of using the information coming from a more objective source such as raw digitalized images so that it is possible to reduce this subjective component. This study inspires to continue the development of the ARTMAP approach, in the sense of eliminating the "black-box" problem of MLP networks as well as trying to reach their same performance.

Acknowledgements

We are very grateful to Dr. Simon S. Cross, who kindly provided the database used in this study. The first author thanks the CONACyT (Consejo Nacional de Ciencia y Tecnología) grant number 70356 and PROMEP (Programa de Mejoramiento del Profesorado), project number PROMEP/103.5/06/0585 for the economic support given for this research. We thank the Macroproyecto Tecnologías para la Universidad de la Información y la Computación de la Universidad Nacional Autónoma de México for the economic support.

References

1. Cheng J and Greiner R. Learning Bayesian Belief Network Classifiers: Algorithms and Systems. in Proceedings of the Canadian Conference on Artificial Intelligence (CSCSI01). 2001. Ottawa, Canada.
2. Cross SS, Downs J, Drezet P, Ma Z and Harrison RF, Which Decision Support Technologies Are Appropriate for the Cytodiagnosis of Breast Cancer?, in Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis, A. Jain, A. Jain, S. Jain and L. Jain, Editors. 2000, World Scientific. p. 265-295.
3. Cross SS, Dube AK, Johnson JS, McCulloch TA, Quincey C, Harrison RF and Ma Z, Evaluation of a statistically derived decision tree for the cytodiagnosis of fine needle aspirates of the breast (FNAB). Cytopathology, 1998. **8**: p. 178-187.

4. Cross SS, Stephenson TJ, Mohammed T and Harrison RF, Validation of a decision support system for the cytodiagnosis of fine needle aspirates of the breast using a prospectively collected dataset from multiple observers in a working clinical environment. Cytopathology, 2000(11): p. 503-512.
5. Cruz-Ramirez Nicandro N-FL, Acosta-Mesa Hector Gabriel, Barrientos-Martinez Erandi, Rojas-Marcial Juan Efrain, A Parsimonious Constraint-based Algorithm to Induce Bayesian Network Structures from Data, in IEEE Proceedings of the Mexican International Conference on Computer Science ENC 2005, IEEE, Editor. 2005, IEEE: Puebla. p. 306-313.
6. Marcus GF, Rethinking Eliminative Connectionism. Cognitive Psychology, 1998. **37**: p. 243-282.
7. Quinlan JR, C4.5: Programs for Machine Learning. The Morgan Kaufmann Series in Machine Learning, ed. P. Langley. 1993: Morgan Kaufmann. 302.
8. Spirtes P, Glymour C and Scheines R, Causation, Prediction and Search. First ed. Lecture Notes in Statistics, ed. J. Berger, S. Fienberg, J. Gani, K. Krickeberg, I. Olkin and B. Singer. Vol. 81. 1993: Springer-Verlag. 526.
9. Witten IH and Frank E, Data Mining: Practical machine learning tools and techniques. Second ed. 2005: Morgan Kaufmann, San Francisco, 2005.

MFCM for Nonlinear Blind Channel Equalization

Soowhan Han[1], Sungdae Park[1], and Witold Pedrycz[2]

[1] Dept. of Multimedia Eng., Dongeui Univ., Busan, Korea 614-714
swhan@deu.ac.kr
[2] Dept. of Electrical & Computer Eng., Univ. of Alberta, Edmonton, Canada T6G 2G7
pedrycz@ee.ualberta.ca

Abstract. In this study, we present a modified Fuzzy *C*-Means (MFCM) algorithm for nonlinear blind channel equalization. The proposed MFCM searches the optimal channel output states of a nonlinear channel, based on the Bayesian likelihood fitness function instead of a conventional Euclidean distance measure. In its searching procedure, all of the possible desired channel states are constructed by the combinations of estimated channel output states. The desired state with the maximum Bayesian fitness is selected and placed at the center of a Radial Basis Function (RBF) equalizer to reconstruct transmitted symbols. In the simulations, binary signals are generated at random with Gaussian noise. The performance of the proposed method is compared with that of a hybrid genetic algorithm (GA augment by simulated annealing (SA), GASA). It is shown that a relatively high accuracy and fast search speed has been achieved.

1 Introduction

In digital communication systems, data symbols are transmitted at regular intervals. Time dispersion caused by non-ideal channel frequency response characteristics, or by multipath transmission, may create inter-symbol interference (ISI). This has become a limiting factor in many communication environments. Furthermore, the nonlinear character of ISI that often arises in high speed communication channels degrades the performance of the overall communication system [1]. To overcome this detrimental ISI effects and to achieve high-speed and reliable communication, we have to resort ourselves to nonlinear channel equalization.

The conventional approach to linear or nonlinear channel equalization requires an initial training period, with a known data sequence, to learn the channel characteristics. In contrast to standard equalization methods, the so-called blind (or self-recovering) equalization methods operate without a training sequence [2]. Because of its superiority, the blind equalization method has gained practical interest during the last few years. Most of the studies carried out so far are focused on linear channel equalization [3]-[4].

Only a few papers have dealt with nonlinear channel models. The blind estimation of Volterra kernels, which characterize nonlinear channels, was presented in [5], and a maximum likelihood (ML) method implemented via expectation-maximization (EM) was introduced in [6]. The Volterra approach suffers from enormous complexity. Furthermore the ML approach requires some prior knowledge of the nonlinear channel structure to estimate the channel parameters. The approach with a nonlinear

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 88–98, 2007.
springerlink.com

structure such as multilayer perceptrons, being trained to minimize some cost function, have been investigated in [7]. However, in this method, the structure and complexity of the nonlinear equalizer must be specified in advance. The support vector (SV) equalizer proposed by Santamaria et al. [8] can be a possible solution for both of linear and nonlinear blind channel equalization at the same time, but it still suffers from high computational cost of its iterative reweighted quadratic programming procedure. A unique approach to nonlinear channel blind equalization was offered by Lin et al. [9], in which they used the simplex GA method to estimate the optimal channel output states instead of estimating the channel parameters directly. The desired channel states were constructed from these estimated channel output states, and placed at the center of their RBF equalizer. With this method, the complex modeling of the nonlinear channel can be avoided. Recently this approach has been implemented with a hybrid genetic algorithm (that is genetic algorithm, GA merged with simulated annealing (SA); GASA) instead of the simplex GA. The resulting better performance in terms of speed and accuracy has been reported in [10]. However, for real-time use, the estimation accuracy and convergence speed in search of the optimal channel output states needs further improvement.

In this study, we propose a new modified Fuzzy *C*-Means (MFCM) algorithm to determine the optimal output states of a nonlinear channel. The FCM algorithm introduced in [11] and being widely used in pattern recognition, system modeling, and data analysis relies on the use of some distance function. Typically, this distance is viewed as the Euclidean one. In the proposed modifications, the construction stage for the possible data set of desired channel states by the elements of estimated channel output states and the selection stage by the Bayesian likelihood fitness function are added to the conventional FCM algorithm. These two additional stages make it possible to search for the optimal output states of a nonlinear blind channel. The MFCM shows the relatively high estimation accuracy combined with fast convergence speed. Its performance is compared with the one using the GASA. In the experiments, three different nonlinear channels are evaluated. The optimal output states of each of nonlinear channels are estimated using both MFCM and GASA. Using the estimated channel output states, the desired channel states are derived and placed at the center of a RBF equalizer to reconstruct transmitted symbols.

2 Nonlinear Channel Equalization Using RBF Networks

A nonlinear channel equalization system is shown in Fig. 1. A digital sequence *s(k)* is transmitted through the nonlinear channel, which is composed of a linear portion described by *H(z)* and a nonlinear component *N(z)*, governed by the following expressions,

$$\bar{y}(k) = \sum_{i=0}^{p} h(i)s(k-i) \tag{1}$$

$$\hat{y}(k) = D_1\bar{y}(k) + D_2\bar{y}(k)^2 + D_3\bar{y}(k)^3 + D_4\bar{y}(k)^4 \tag{2}$$

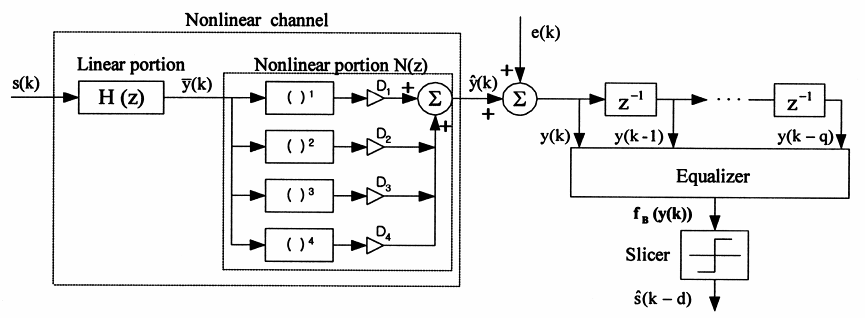


Fig. 1. The structure of a nonlinear channel equalization system

where p is the channel order and D_i is the coefficient of the i^{th} nonlinear term. The transmitted symbol sequence $s(k)$ is assumed to be an equiprobable and independent binary sequence taking values from $\{\pm 1\}$. We consider that the channel output is corrupted by an additive white Gaussian noise $e(k)$. Given this the channel observation $y(k)$ can be written as

$$y(k) = \hat{y}(k) + e(k) \tag{3}$$

If q denotes the equalizer order (number of tap delay elements in the equalizer), then there exist $M = 2^{p+q+1}$ different input sequences

$$\boldsymbol{s}(\boldsymbol{k}) = [s(k), s(k-1), \cdots, s(k-p-q)] \tag{4}$$

that may be received (where each component is either equal to 1 or –1). For a specific channel order and equalizer order, the number of input patterns that influence the equalizer is equal to M, and the input vector of equalizer without noise is

$$\hat{\boldsymbol{y}}(\boldsymbol{k}) = [\hat{y}(k), \hat{y}(k-1), \cdots, \hat{y}(k-q)] \tag{5}$$

The noise-free observation vector $\hat{\boldsymbol{y}}(\boldsymbol{k})$ is referred to as the desired channel states, and can be partitioned into two sets, $\boldsymbol{Y}_{q,d}^{+1}$ and $\boldsymbol{Y}_{q,d}^{-1}$, as shown in (6) and (7), depending on the value of $s(k\text{-}d)$, where d is the desired time delay.

$$\boldsymbol{Y}_{q,d}^{+1} = \{\ \hat{\boldsymbol{y}}(\boldsymbol{k}) \mid s(k-d) = +1\ \} \tag{6}$$

$$\boldsymbol{Y}_{q,d}^{-1} = \{\ \hat{\boldsymbol{y}}(\boldsymbol{k}) \mid s(k-d) = -1\ \} \tag{7}$$

The task of the equalizer is to recover the transmitted symbols $s(k\text{-}d)$ based on the observation vector $\boldsymbol{y(k)}$. Because of the additive white Gaussian noise, the observation vector $\boldsymbol{y(k)}$ is a random process having conditional Gaussian density functions centered at each of the desired channel states, and determining the value of $s(k\text{-}d)$ becomes a decision problem. Therefore, Bayes decision theory [12] can be applied to derive the optimal solution for the equalizer. The solution forming the optimal Bayesian equalizer is given as follows

$$f_B(\boldsymbol{y}(\boldsymbol{k}))= \sum_{i=1}^{n_s^{+1}}\exp(-\left\|\boldsymbol{y}(\boldsymbol{k})-\boldsymbol{y}_i^{+1}\right\|^2/2\sigma_e^2) - \sum_{i=1}^{n_s^{-1}}\exp(-\left\|\boldsymbol{y}(\boldsymbol{k})-\boldsymbol{y}_i^{-1}\right\|^2/2\sigma_e^2) \tag{8}$$

$$\hat{s}(k-d)=\mathrm{sgn}(f_B(\boldsymbol{y}(\boldsymbol{k}))) = \begin{cases} +1, & f_B(\boldsymbol{y}(\boldsymbol{k}))\geq 0 \\ -1, & f_B(\boldsymbol{y}(\boldsymbol{k}))<0 \end{cases} \tag{9}$$

where $\boldsymbol{y}_i^{+1}$ and $\boldsymbol{y}_i^{-1}$ are the desired channel states belonging to sets $\boldsymbol{Y}_{q,d}^{+1}$ and $\boldsymbol{Y}_{q,d}^{-1}$, respectively, and their numbers are denoted as n_s^{+1} and n_s^{-1}, and σ_e^2 is the noise variance. The desired channel states, $\boldsymbol{y}_i^{+1}$ and $\boldsymbol{y}_i^{-1}$, are derived by considering their relationship with the channel output states (as it will be explained in the next section). In this study, the optimal Bayesian decision probability (8) is implemented with the use of some RBF network. The output of this network is given as

$$f(\boldsymbol{x})=\sum_{i=1}^{n}\omega_i\phi(\frac{\left\|\boldsymbol{x}-\boldsymbol{c}_i\right\|^2}{\rho_i}) \tag{10}$$

where *n* is the number of hidden units, c_i are the centers of the receptive fields, ρ_i is the width of the i^{th} units and ω_i is the corresponding weight. The RBF network is an ideal processing means to implement the optimal Bayesian equalizer when the nonlinear function ϕ is chosen as the exponential function $\phi(x)=e^{-x}$ and all of the widths are the same and equal to ρ, which is twice as large as the noise variance σ_e^2. For the case of equiprobable symbols, the RBF network can be simplified by setting half of the weights to 1 and the other half to -1. Thus the output of this RBF equalizer is same as the optimal Bayesian decision probability in (8).

3 Desired Channel States and Channel Output States

The desired channel states, $\boldsymbol{y}_i^{+1}$ and $\boldsymbol{y}_i^{-1}$, are used as the centers of the hidden units in the RBF equalizer to reconstruct the transmitted symbols. If the channel order *p*=1 with $H(z)=0.5+1.0z^{-1}$, the equalizer order *q* is equal to 1, the time delay *d* is also set to 1, and the nonlinear portion is described by $D_1=1, D_2=0.1, D_3=0.05, D_4=0.0$ see Fig. 1, then the eight different channel states ($2^{p+q+1}=8$) may be observed at the receiver in the noise-free case. Here the output of the equalizer should be $\hat{s}(k-1)$, as shown in Table 1. From this table, it can be seen that the desired channel states $[\hat{y}(k),\hat{y}(k-1)]$ can be constructed from the elements of the dataset, called "channel output states", $\{a_1,a_2,a_3,a_4\}$, where for this particular channel we have $a_1=1.89375, a_2=-0.48125, a_3=0.53125, a_4=-1.44375$. The length of dataset, $\tilde{n}$, is determined by the channel order, *p*, such as $2^{p+1}=4$. In general, if *q*=1 and *d*=1, the desired channel states for $\boldsymbol{Y}_{1,1}^{+1}$ and $\boldsymbol{Y}_{1,1}^{-1}$ are (a_1,a_1), (a_1,a_2), (a_3,a_1), (a_3,a_2), and (a_2,a_3), (a_2,a_4), (a_4,a_3), (a_4,a_4), respectively. In the case of *d*=0, the channel states, (a_1,a_1), (a_1,a_2), (a_2,a_3), (a_2,a_4), belong to $\boldsymbol{Y}_{1,1}^{+1}$, and ($a_3,a_1$), ($a_3,a_2$), ($a_4,a_3$), ($a_4,a_4$) belong to

$\boldsymbol{Y}_{1,1}^{-1}$. This relation is valid for the channel that has a one-to-one mapping between the channel inputs and outputs [9]. Thus the desired channel states can be derived from the channel output states if we assume *p* is known, and the main problem of blind equalization can be changed to focus on finding the optimal channel output states from the received patterns.

Table 1. The relation between desired channel states and channel output states

Nonlinear channel with $H(z)=0.5+1.0z^{-1}, D_1=1, D_2=0.1, D_3=0.05, D_4=0.0$, and d=1						
Transmitted symbols			Desired channel states			Output of equalizer
$s(k)$	$s(k-1)$	$s(k-2)$	$\hat{y}(k)$	$\hat{y}(k-1)$	By channel output states, $\{a_1, a_2, a_3, a_4\}$	$\hat{s}(k-1)$
1	1	1	1.89375	1.89375	(a_1, a_1)	1
1	1	−1	1.89375	–0.48125	(a_1, a_2)	1
−1	1	1	0.53125	1.89375	(a_3, a_1)	1
−1	1	−1	0.53125	–0.48125	(a_3, a_2)	1
1	−1	1	–0.48125	0.53125	(a_2, a_3)	–1
1	−1	−1	–0.48125	–1.44375	(a_2, a_4)	–1
−1	−1	1	–1.44375	0.53125	(a_4, a_3)	–1
−1	−1	−1	–1.44375	–1.44375	(a_4, a_4)	–1

It is known that the Bayesian likelihood (*BL*), defined in (11), is maximized with the desired channel states derived from the optimal channel output states [13].

$$BL=\prod_{k=0}^{L-1}\max(f_B^{+1}(k), f_B^{-1}(k)) \tag{11}$$

where $f_B^{+1}(k)=\sum_{i=1}^{n_s^{+1}}\exp\left(-\left\|\boldsymbol{y(k)}-\boldsymbol{y}_i^{+1}\right\|^2/2\sigma_e^2\right)$, $f_B^{-1}(k)=\sum_{i=1}^{n_s^{-1}}\exp\left(-\left\|\boldsymbol{y(k)}-\boldsymbol{y}_i^{-1}\right\|^2/2\sigma_e^2\right)$ and *L* is the length of received sequences. Therefore, the *BL* is utilized as the fitness function (*FF*) of the proposed algorithm to find the optimal channel output states after taking the logarithm, which is shown in equation (12).

$$FF=\sum_{k=0}^{L-1}\log(\max(f_B^{+1}(k), f_B^{-1}(k))) \tag{12}$$

The optimal channel output states, which maximize the fitness function *FF*, cannot be obtained with the use of the conventional gradient-based methods, because the mathematical formulation between the channel output states and *FF* cannot be accomplished not knowing the channel structure [9]. For carrying out search of these optimal channel output states, we develop a modified version of the FCM (MFCM).

4 A Modified Fuzzy C-Means Algorithm (MFCM)

In comparison with the standard version of the FCM, the proposed modification of the clustering algorithm comes with two additional stages. One of them concerns the construction stage of possible data set of desired channel states with the derived elements of channel output states. The other is the selection stage for the optimal desired channel states among them based on the Bayesian likelihood fitness function. For the channel shown in Table 1, the four elements of channel output states are required to construct the optimal desired channel states. If the candidates, $\{c_1, c_2, c_3, c_4\}$, for the elements of optimal channel output states $\{a_1, a_2, a_3, a_4\}$, are extracted from the centers of a conventional FCM algorithm, twelve (4!/2) different possible data set of desired channel states can be constructed by completing matching between $\{c_1, c_2, c_3, c_4\}$and $\{a_1, a_2, a_3, a_4\}$. For the fast matching, the arrangements of $\{c_1, c_2, c_3, c_4\}$ are saved to the set *C* such as *C*(1)=1,2,3,4, *C*(2)=1,2,4,3, …,*C*(12)=3,2,1,4 before the search process starts. For example, *C*(2)=1,2,4,3 means the desired channel states is constructed with c_1 for a_1, c_2 for a_2, c_4 for a_3, and c_3 for a_4 in Table 1. At a next stage, a data set of desired channel states, which has a maximum Bayesian fitness value as described by (12), is selected. This data set is utilized as a center set used in the FCM algorithm. Subsequently the partition matrix *U* is updated and a new center set is sequentially derived with the use of this updated matrix *U*. The new four candidates for the elements of optimal output states are extracted from this new center set based on the relation presented in Table 1 (each value of the new $\{c_1, c_2, c_3, c_4\}$ is replaced with $\{a_1, a_2, a_3, a_4\}$ in the selected data set, respectively). These steps are repeated until the Bayesian likelihood fitness function has not been changed or the maximum number of iteration has been reached. The proposed MFCM algorithm can be concisely described in the form of its pseudo-code.

```
begin
      save arrangements of candidates, $\{c_1, c_2, c_3, c_4\}$, to C
      randomly initialize the candidates, $\{c_1, c_2, c_3, c_4\}$
   while (new fitness function – old fitness function) < threshold value
         for k=1 to C size
           map the arrangement of candidates, C[k] , to $\{a_1, a_2, a_3, a_4\}$
             construct a set of desired channel states
                                based on the relation shown in table 1
             calculate its fitness function (FF[k]) by equation (12)
         end
    find a data set which has a maximum FF in k=1..C
    update the membership matrix U by the data set utilized as a center set
                   in the conventional FCM algorithm
    derive a new center set by the U
         extract the candidates, $\{c_1, c_2, c_3, c_4\}$, from the new center set
based on the relation shown in table 1
      end
end
```

In the search process carried out by the MFCM, a data set for the desired channel states which exhibits a maximum fitness value is always selected, and the candidates $\{c_1, c_2, c_3, c_4\}$ for the elements of channel output states are extracted from the data set by using the pre-established relation in Table 1. This means that the set of desired channel states produced by MFCM is always close to the optimal set and it has the same structure as shown in Table 1. Thus the centers of the first half in its output present the desired channel states for $\boldsymbol{Y}_{1,1}^{+1}$ and the rest present for $\boldsymbol{Y}_{1,1}^{-1}$, or reversely. In addition, in the pseudo-code, the MFCM checks all of the possible arrangements, *C*, to find the data set which has a maximum *FF* in *while*-loop. However, for the fast searching of the MFCM, it is not necessary to keep this work during the entire procedure. The MFCM is forced to choose the data set with a maximum value of the *FF* and each value of the new candidates $\{c_1, c_2, c_3, c_4\}$ for next loop is always replaced with $\{a_1, a_2, a_3, a_4\}$ in the selected data set, respectively. Therefore, after the first couple of *while*-loops, the desired channel states constructed with the arrangement *C*(1) has the maximum *FF* and the selected index *k* is quickly going to "1". This effect will be clearly shown in our experiments.

5 Experimental Studies and Performance Assessments

To present the effectiveness of the proposed method, we consider blind equalization realized with GASA and MFCM. Three nonlinear channels in [9] are discussed. Channel 1 is shown in Table 1 while the other two channels are described as follows.

Channel 2: $H(z) = 0.5 + 1.0^{-1}, D_1 = 1, D_2 = 0.1, D_3 = -0.2, D_4 = 0.0$, and *d*=1

Channel 3: $H(z) = 0.5 + 1.0^{-1}, D_1 = 1, D_2 = 0.0, D_3 = -0.9, D_4 = 0.0$, and *d*=1

The parameters of the optimization environments for each of the algorithms are included in Table 2, and these are fixed for all experiments. The choice of these specific parameter values is not critical to the performance of GASA and MFCM. The fitness function described by (12) is utilized in both algorithms.

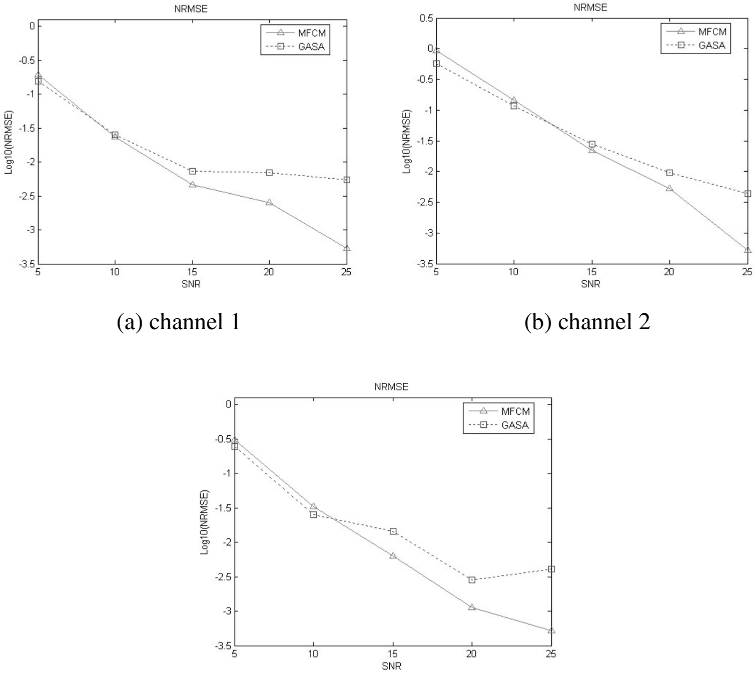
Table 2. Parameters of the optimization environments

GASA	Population size	50
	Maximum number of generation	100
	Crossover rate	0.8
	Mutation rate	0.1
	Random initial temperature	[0, 1]
	Cooling rate	0.99
MFCM	Maximum number of iteration	100
	Minimum amount of improvement	10^{-5}
	Exponent for the matrix U	2
	Random initial output states	[-1 1]

In the experiments, 10 independent simulations for each of three channels with five different noise levels (SNR=5,10,15,20 and 25db) are performed with 1,000 randomly generated transmitted symbols and the results are averaged. The MFCM and GASA have been implemented in a batch mode in facilitate comparative analysis. With this regard, we determine the normalized root mean squared errors (NRMSE)

$$\text{NRMSE}=\frac{1}{\|\boldsymbol{a}\|}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{a}-\hat{\boldsymbol{a}}_i\|^2} \tag{13}$$

where $\boldsymbol{a}$ is the dataset of optimal channel output states, $\hat{\boldsymbol{a}}_i$ is the dataset of estimated channel output states, and m is the number of experiments performed (m=10). As shown in Fig. 2, the proposed MFCM comes with the lower NRMSE for all three channels even the differences are not significant when dealing with high levels of noise. A sample of 1,000 received symbols under 5db SNR for channel 1 and its desired channel states constructed from the estimated channel output states by MFCM and GASA is shown in Fig. 3.



(a) channel 1

(b) channel 2

(c) channel 3

Fig. 2. NRMSE for the MFCM and GASA

In addition, we compared the search time of the algorithms. As mentioned in Section 4, for the fast convergence speed of MFCM, it is not necessary to check all of the possible arrangements, C, in *while*-loop during the entire searching procedure because the new candidates for next loop are always updated by using the arrangement $C(1)$.

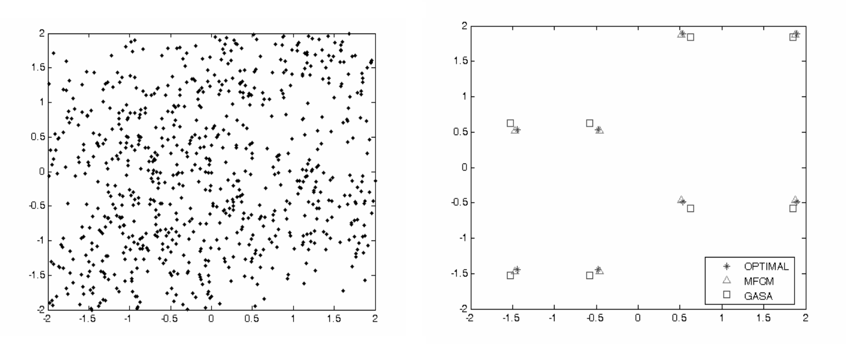


Fig. 3. A sample of received symbols for channel 1 and its desired channel states produced by MFCM and GASA

The selection index *k* for the maximum *FF* is not changed after the first couple *while*-loops, and it is quickly going to "1". Thus the *for*-loop in the pseudo-code of MFCM is skipped if the index *k* has not changed during the last 5 epochs. The search times for MFCM and GASA are included in Table 3; Notably, the proposed MFCM offers much higher search speed for all three channels and this could be attributed to its simple structure. Finally, we investigated the bit error rates (BER) when using the RBF equalizer; refer to Table 4. It becomes apparent that the BER with the estimated channel output states realized by the MFCM is almost same as the one with the optimal output states for all three channels.

Table 3. The averaged search time (in sec) for MFCM and GASA (**Simulation environment: Pentium4 2.6Ghz, 512M Memory, code written in Matlab 6.5**)

Channel	SNR	GASA	MFCM
Channel 1	5db	70.1922	0.3188
	10db	68.8266	0.2984
	15db	68.6516	0.2781
	20db	69.0344	0.3469
	25db	69.2734	0.3812
Channel 2	5db	76.2453	0.3672
	10db	76.8344	0.2766
	15db	75.475	0.2375
	20db	76.0375	0.2297
	25db	76.2812	0.225
Channel 3	5db	77.2781	0.3094
	10db	77.7781	0.2469
	15db	77.5562	0.2094
	20db	77.4016	0.2375
	25db	79.0734	0.2109

Table 4. Averaged BER (no. of errors/no. of transmitted symbols) for three nonlinear channels

Channel	SNR	with optimal states	GASA	MFCM
Channel 1	5db	0.0799	0.0822	0.0810
	10db	0.0128	0.0128	0.0127
	15db	0	0.0001	0.0001
	20db	0	0	0
	25db	0	0	0
Channel 2	5db	0.1573	0.1592	0.1595
	10db	0.0487	0.0494	0.0492
	15db	0.0040	0.0039	0.0038
	20db	0	0	0
	25db	0	0	0
Channel 3	5db	0.1078	0.1092	0.1089
	10db	0.0271	0.0274	0.0272
	15db	0.0002	0.0003	0.0002
	20db	0	0	0
	25db	0	0	0

6 Conclusions

In this paper, we have introduced a new modified fuzzy c-means clustering algorithm and showed its application to nonlinear channel blind equalization. In this approach, the highly demanding modeling of an unknown nonlinear channel becomes unnecessary as the construction of the desired channel states is accomplished directly on a basis of the estimated channel output states. It has been shown that the proposed MFCM with the Bayesian likelihood treated as the fitness function offers better performance in comparison to the solution provided by the GASA approach. In particular, MFCM successively estimates the channel output states with relatively high speed and substantial accuracy. Therefore an RBF equalizer, based on MFCM, can constitute a viable solution for various problems of nonlinear blind channel equalization. Our future research pursuits are oriented towards the use of the MFCM under more complex optimization environments, such as those encountered when dealing with channels of high dimensionality and equalizers of higher order.

References

1. Biglieri, E., Gersho, A., Gitlin, R. D., Lim, T. L.: Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission. IEEE J. Selected Areas Commun. SAC-2(5) (1984) 765-777
2. Proakis, J. G.: Digital Communications. Fourth Edition, McGraw-Hill, New York (2001)
3. Serpedin, E., Giannakis, G. B.: Blind channel identification and equalization with modulation-induced cyclostationarity. IEEE Trans. Siganl Processing 46 (1998) 1930-1944

4. Fang, Y., Chow, W. S., Ng, K. T.: Linear neural network based blind equalization. Signal Processing 76 (1999) 37-42
5. Stathaki, T., Scohyers, A.: A constrained optimization approach to the blind estimation of Volterra kernels. Proc. IEEE Int. Conf. on ASSP 3 (1997) 2373-2376
6. Kaleh, G. K., Vallet, R.: Joint parameter estimation and symbol detection for linear or nonlinear unknown channels. IEEE Trans. Commun. 42 (1994) 2406-2413
7. D. Erdogmus, D. Rende, J.C. Principe, T.F. Wong: Nonlinear channel equalization using multilayer perceptrons with information theoretic criterion. Proc. Of IEEE workshop Neural Networks and Signal Processing, MA, U.S.A. (2001) 443-451
8. I. Santamaria, C. Pantaleon, L. Vielva, J. Ibanez: Blind Equalization of Constant Modulus Signals Using Support Vector Machines. IEEE Trans. Signal Processing 52 (2004) 1773-1782
9. Lin, H., Yamashita, K.: Hybrid simplex genetic algorithm for blind equalization using RBF networks. Mathematics and Computers in Simulation 59 (2002) 293-304
10. S. Han, W. Pedrycz, C. Han: Nonlinear Channel Blind Equalization Using Hybrid Genetic Algorithm with Simulated Annealing. Mathematical and Computer Modeling 41 (2005) 697-709
11. J.C. Bezdek: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
12. Duda, R. O., Hart, P. E.: Pattern Classification and Scene Analysis. NewYork, Wiley(1973)
13. Lin, H., Yamashita, K.: Blind equalization using parallel Bayesian decision feedback equalizer. Mathematics and Computers in Simulation 56 (2001) 247-257

Fuzzy Rules Extraction from Support Vector Machines for Multi-class Classification

Adriana da Costa F. Chaves, Marley Maria B.R. Vellasco, and Ricardo Tanscheit

Electrical Engineering Department,
Pontifical Catholic University of Rio de Janeiro
achaves@inf.puc-rio.br, marley@ele.puc-rio.br,
ricardo@ele.puc-rio.br

Abstract. This paper proposes a new method for fuzzy rule extraction from trained support vector machines (SVMs) for multi-class problems. SVMs have been applied to a wide variety of application. However, SVMs are considered “black box models”, where no interpretation about the input-output mapping is provided. Some methods to reduce this limitation have already been proposed, however, they are restricted to binary classification problems and to the extraction of symbolic rules with intervals or functions in their antecedents. Hence, to improve the interpretability of the generated rules, this paper presents a new model for extracting fuzzy rules from a trained SVM. Moreover, the proposed model was developed for classification in multi-class problems. The generated fuzzy rules are presented in the format “IF x_1 is C_1 AND x_2 is C_2 AND … AND x_n is C_n , THEN $x = (x_1, x_2, \ldots x_n)$ is of class A", where $C_1, C_2, ..., C_n$ are fuzzy sets. The proposed method was evaluated in four benchmark databases (Bupa Liver Disorders, Wisconsin Breast Cancer, Iris and Wine). The results obtained demonstrate the capacity of the proposed method to generate a set of interpretable rules that explains the database and the influence of the input variables in the determination of the final class.

1 Introduction

Support vector machines (SVMs) are learning systems, based on statistical learning theory [1, 2, 3, 4, 5, 6], that have been applied with excellent generalization performance to a wide variety of applications in classification and regression [7, 8, 9, 10, 11].

Despite their excellent performance, SVMs, as well as artificial neural networks, are “black box models”, i.e., models that do not explain in a transparent way the process by which they have arrived at a given result. The resulting input-output mapping is composed of a linear combination of kernel functions [1], which is very hard to interpret.

Algorithms whose purpose is to extract useful knowledge from a trained SVM have already been proposed, among them RulExtSVM [12] and SVM+Prototypes [13]. The algorithm RulExtSVM extracts IF-THEN rules with intervals, defined by hyper-rectangular forms, in the rules’ antecedents. The SVM+Prototype method calculates ellipsoids (called prototypes) based on the obtained support vectors of each class. These ellipsoids are also used in the rules’ antecedents.

The disadvantage of the RulExtSVM is the very expensive construction of as many hyper-rectangles as the number of support vectors. In the SVM+Prototype method,

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 99–108, 2007.
springerlink.com

the construction of ellipsoids with axes non parallel to the coordinate axes reduces interpretability. This problem is intensified when the input space is of high dimension, i.e., there are many input attributes.

It must be stressed that the rules extracted from both methods generate, in their antecedents, intervals or functions. This fact reduces the interpretability of the generated rules and jeopardizes the capacity of knowledge extraction. Hence, to increase the linguistic interpretability of the generated rules, this paper presents a new methodology for extracting fuzzy rules from a trained SVM. The basic idea is that, by employing fuzzy sets in the rules' antecedents, the resulting rules will be more flexible and interpretable.

This paper is divided into four additional sections. Section 2 briefly describes the theory of support vector machines. Section 3 introduces the proposed method for extraction of fuzzy rules from trained SVMs. First the method is presented for binary classification problems; then the fuzzy rule extraction model is extended to multi-classification applications. Section 4 presents case studies, describing the benchmark databases employed and the performance obtained with the fuzzy rule extraction method. Finally, discussion and conclusion are presented in Section 5.

2 Support Vector Machines

2.1 Binary Classification

Consider a training set $\{(x_i, y_i)\}$, i ∈ {1, …, N}, where $x_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$, and N is the number of patterns. SVM solves a quadratic programming optimization problem:

$$\text{maximize} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{1}$$

$$\text{subject to} \quad 0 \le \alpha_i \le C \text{ , and } \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{2}$$

The function $K(x_i, x_j) = \Phi(x_i). \Phi(\ x_j)$ is a kernel function, where $\Phi(x_i)$ represents the mapping of input vector x_i into a higher dimensional space (called “feature space”). C is the regularization constant, a positive training parameter which establishes a trade off between the model complexity and the training error and α_i are Lagrange coefficients.

In the solution of the previous problem, $\alpha_i = 0$ except for the support vectors, which will be represented by s_i in order to distinguish them from other data samples.

There are many ways for solving the optimization problem to obtain SVM classifiers. These methods are described in [1, 3].

A data point x is classified according to the sign of the decision function:

$$f(x) = \sum_{i=1}^{Ns} \alpha_i y_i K(s_i, x) + b \ , \tag{3}$$

where b is the bias and Ns is the number of support vectors.

2.2 Multiple Class Classification

As can be observed from the previous section, SVM was originally defined for binary classification. There are basically two approaches to extend SVM for classification in k classes (k > 2): one that reduces the k-classes problem to a set of binary problems; and one that involves a generalization of the binary SVM [14, 15].

Two well known methods based on the first approach are the so-called "one-against-all" [14, 16, 17] and "one-against-one” [14, 16, 18, 19]. The one-against-all method builds k binary SVMs, each of them dedicated to separating each class from the others. The outputs of all SVMs are then combined to generate the final classification in k classes. The most common method for combining the k SVMs outputs, which will be used in the case studies of this work, is to assign the input vector to the class that provides the largest value of the decision function.

The one-against-one method is based on the construction of k(k-1)/2 binary SVMs, which separate each possible pair of classes. The final classification is obtained from the outputs of all SVMs. The basic disadvantage of this method is the larger number of SVMs that must be trained, when compared with the one-against-all method. On the other hand, each SVM is trained with just the patterns that belong to the two classes involved, while each of the k SVMs in the one-against-all method makes use of all training points in the data set. Usually, the final classification of an input pattern is accomplished by a simple voting strategy from all k(k-1)/2 SVMs.

The second approach, based on the generalization of the binary SVM, establishes a more natural way to solve the k-classes problem by constructing a decision function encompassing all classes involved. The most important method of this type is called Crammer and Singer [15], where the k-classes problem is solved by a single optimization problem (generalization of the binary SVM optimization problem). In this method, all training data are used at the same time.

As described in section 3.2, the proposed fuzzy rule extraction method can be applied to any of the above methods. A discussion of the performance of these methods is provided in section 4.

3 Fuzzy Rule Extraction Methodology

The proposed method, called FREx_SVM, is divided into three steps: Projection of Support Vectors, Definition of Fuzzy Sets, and Fuzzy Rule Extraction. For the sake of simplicity, the FREx_SVM method is first described for a binary SVM. Then, in Section 3.2, it is extended to a multi-class SVM.

3.1 FREx_SVM for Binary SVM

3.1.1 Projection of Support Vectors

In this first step of the algorithm, the support vectors obtained by the SVM are projected on the coordinate axes. There are as many projections as the input space dimension (number of input attributes of the data base under analysis).

3.1.2 Definition of Fuzzy Sets

The next step consists of defining a number of overlapping fuzzy sets to each input variable. Each fuzzy set is labeled and assigned to a triangular membership function, usually with equal domain. Suppose that each attribute of a k-dimensional input space has been divided into n fuzzy sets. In this case, the fuzzy set C_{ij} is the j^{th} set defined for the i^{th} coordinate, where $i \in \{1, \dots, k\}$ and $j \in \{1, \dots, n\}$. Figure 1 presents an example of a two-dimension input space ($k = 2$), where each variable has been partitioned into five fuzzy sets ($n = 5$).

Each support vector projection obtained in the previous step (one projection for each input variable) is then evaluated in all n membership functions, selecting the fuzzy set that provides the maximum membership degree. Let x_i be the projection of the support vector x on the i^{th} coordinate and $\mu_{C_{ij}}(x_i)$ the membership degree of x_i to the set C_{ij}. The maximum membership degree is then calculated by:

$$\max_{j\in\{1,\dots,n\}} \{\mu_{C_{ij}}(x_i)\}\,, \tag{4}$$

As shown in Figure 2, the projections of the support vector x are $x_1 = 0.61$ and $x_2 = 0.88$. These projections activate two membership functions each, with the following values: $\mu_{C14}(x_1) = 0.78$, $\mu_{C15}(x_1) = 0.22$, $\mu_{C24}(x_2) = 0.24$, and $\mu_{C25}(x_2) = 0.76$. The sets with the maximum membership degree are C_{14} e C_{25}, marked with a square. For better visualization, the sets with membership degrees equal to 0 are not shown in Figure 2.

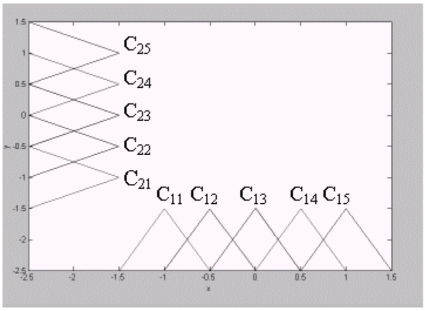


Fig. 1. Definition of fuzzy sets

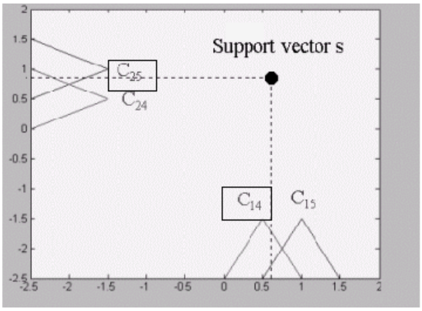


Fig. 2. Maximum membership degrees

3.1.3 Fuzzy Rules Extraction

In the final step, each support vector generates a fuzzy rule, as described below. For each support vector x, let C_{ij^i} be the fuzzy set with higher membership degree for each x_i coordinate, $i = 1, \dots, k$ and $j^i = 1, \dots, n$. The rule generated for support vector x is: IF x_1 is C_{1j^1} and … and x_k is C_{kj^k} THEN $x = (x_1, \dots, x_n)$ belongs to the class defined by the support vector x. The maximum number of rules equals the total number of obtained support vectors.

In order to evaluate each rule, two metrics - *fuzzy accuracy* and *fuzzy coverage* - were defined to assess, respectively, how accurate the rule describes the data base and how many patterns are affected by the rule. Conflicting rules (rules with the same antecedent but with different consequents) are resolved by choosing the rule with highest fuzzy accuracy. Details of these proposed metrics are presented in [20].

3.2 FREx_SVM for Multi-class SVM

Since the proposed method is based on the support vectors obtained after training all SVMs, the binary approach presented in the previous section can be easily extended to multi-class problems. Any of the multi-class methods presented in section 2.2 can be used with FREx_SVM. The diagram presented in Figure 3 describes the whole process of extracting fuzzy rules from multi-class classification problems.

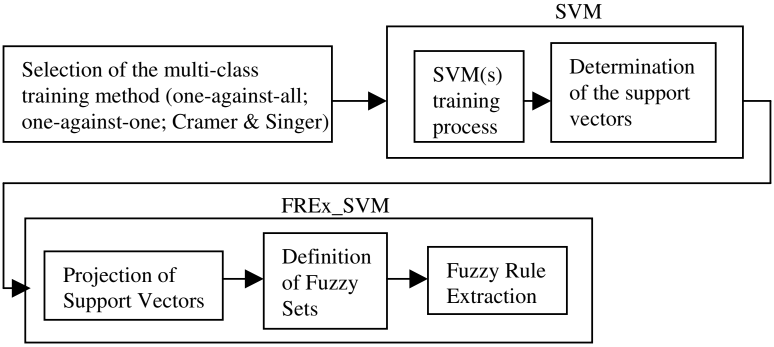

SVM
Selection of the multi-class training method (one-against-all; one-against-one; Cramer & Singer)
SVM(s) training process
Determination of the support vectors
FREx_SVM
Projection of Support Vectors
Definition of Fuzzy Sets
Fuzzy Rule Extraction

Fig. 3. Complete fuzzy rule extraction process

After selection of a specific method, the necessary SVMs are trained with the pre-processed data to obtain the support vectors associated to each class. Each method generates a set of rules. The resulting sets of rules are then evaluated by the fuzzy accuracy and fuzzy coverage metrics, so that the multi-class SVM methods can be compared.

As mentioned, the one-against-all method constructs k SVMs to separate each class from the others. Let SVM$_i$ be the one that separates class i from all others. Therefore, only support vectors from class i are considered in the generation of rules. For the one-against-one method, a SVM is trained to separate class i from class j ($i \neq j$). In this case, all support vectors obtained by this SVM are used in the rules extraction process, since they define the classes involved. The same applies for the Crammer and Singer method, that is, all support vectors are used in the process.

4 Case Studies

To evaluate the FREx_SVM method, four benchmark databases were chosen: Bupa Liver Disorders, Wisconsin Breast Cancer, Iris and Wine. These databases can be found in http://www.ics.uci.edu/~mlearn/MLRepository.html.

All SVMs were trained with two different kernel functions: linear and RBF (Gaussian functions). Three values for the regularization constant (parameter C, eq. 2) and four values for the Gaussian standard deviation were employed to train each SVM: C = 0.1, 1 and 10 and σ^2 = 1, 5, 10 and 50.

In the multi-class benchmarks - Iris and Wine, all multi-classification methods were evaluated in order to compare their performance.

In all experiments the databases were divided into two equal size disjoint sets. These two sets were interchanged for training and testing in two different experiments. The results presented in the following sections are the average results in the test sets of these two experiments.

4.1 Bupa Liver Disorders

This database consists of medical information related to liver disorders with six numeric input attributes and one binary output. In order to evaluate the performance of FREx_SVM, three situations with 3, 5, and 7 fuzzy sets, for each coordinate, were considered.

Table 1 presents the average performance obtained in the test sets with the best parameter configuration among all SVMs trained. The following performance metrics were used for comparing different configurations: the percentage of test examples that are covered by the rules (Coverage), the classification percentage error of the generated rules (Error) and the number of generated rules (Number of rules) for 3, 5 and 7 fuzzy sets. The best Coverage (95.94%) was obtained for 3 fuzzy sets, with only 28 rules.

Table 1. Best Performance of FREx_SVM and SVM - Bupa Liver Disorders

	3 fuzzy sets	5 fuzzy sets	7 fuzzy sets	SVM
Kernel	RBF com σ^2 = 50 e C = 0.1	RBF com σ^2 = 10 e C = 0.1	RBF com σ^2 = 50 e C = 0.1	Linear C = 0.1
Coverage	**95.94%**	85.51%	80.58%	100.00%
Error	**47.25%**	40.87%	36.52%	38.55%
Number of rules	**28**	74	108.5	---

The best rules obtained, regarding the fuzzy accuracy, are presented below:

IF x_1 is C_{16} and x_2 is C_{23} and x_3 is C_{31} and x_4 is C_{42} and x_5 is C_{51} and x_6 is C_{61}, THEN Class 1 — Accuracy: 0.7686, for 7 fuzzy sets.
IF x_1 is C_{14} and x_2 is C_{21} and x_3 is C_{31} and x_4 is C_{41} and x_5 is C_{51} and x_6 is C_{61}, THEN Class 2 — Accuracy: 0.7673, for 5 fuzzy sets.

4.2 Wisconsin Breast Cancer

This database, like Bupa Liver Disorders, consists of medical information related to breast cancer. There are nine numeric input attributes and one binary output. In this particular case, due to the larger number of inputs, the number of pre-defined fuzzy sets for each coordinate was restricted to 3 and 5.

As in the previous example, Table 2 presents the best configurations among all SVMs trained. All the results presented are mean values of the performance obtained in both test sets, using the same evaluation measures: Coverage, Percentage Error and Number of rules.

Table 2. Best Performance of FREx_SVM and SVM - Wisconsin Breast Cancer

	3 fuzzy sets	5 fuzzy sets	SVM
Kernel	RBF com $\sigma^2 = 1$ e C = 0.1	RBF com $\sigma^2 = 1$ e C = 0.1	Linear C = 10
Coverage	**77.90%**	63.98%	100.00%
Error	**02.49%**	00.73%	01.27%
Number of rules	**131.5**	157	---

As in the previous case, the best Coverage (77.90%) was obtained for three fuzzy sets, with 131.5 rules.

The best rule for each class is illustrated below:

IF x_1 is C_{11} and x_2 is C_{21} and x_3 is C_{31} and x_4 is C_{41} and x_5 is C_{51} and x_6 is C_{61} and x_7 is C_{71} and x_8 is C_{81} and x_9 is C_{91}, THEN Class 1 — Accuracy: 1.0000, for 3 fuzzy sets.
IF x_1 is C_{12} and x_2 is C_{23} and x_3 is C_{33} and x_4 is C_{43} and x_5 is C_{52} and x_6 is C_{63} and x_7 is C_{73} and x_8 is C_{83} and x_9 is C_{91}, THEN Class 2 — Accuracy: 1.0000, for 3 fuzzy sets.

4.3 Iris

This database is probably the best known benchmark in pattern recognition literature. It is related to classification of species of the iris plant, which are divided in three classes. There are four numeric input attributes. As in the Bupa Liver Disorder case, three configurations of input partition were evaluated: 3, 5, and 7 fuzzy sets for each coordinate.

Since this is a multi-class problem (three classes), the three multi-class classification methods described in section 2.2 were applied for training the SVMs. Table 3 shows the configurations that provided the best average classification performance of FREx_SVM in the two test sets, using the same measures as in the previous cases.

As can be seen from the results presented in Table 3, the best result was obtained in this case with five fuzzy sets, attaining coverage of 100%, with only 33.5 rules in average.

The best rule, for each class, in terms of the proposed fuzzy accuracy, is provided below:

IF x_1 is C_{11} and x_2 is C_{22} and x_3 is C_{31} and x_4 is C_{41}, THEN Class 1 — Accuracy: 0.9990, for 3 fuzzy sets.

IF x_1 is C_{13} and x_2 is C_{22} and x_3 is C_{33} and x_4 is C_{43}, THEN Class 2 — Accuracy: 0.8245, for 5 fuzzy sets.

IF x_1 is C_{14} and x_2 is C_{23} and x_3 is C_{34} and x_4 is C_{44}, THEN Class 3 — Accuracy: 0.9767, for 5 fuzzy sets.

Table 3. Best Performance of FREx_SVM and SVM - Iris

	3 fuzzy sets	5 fuzzy sets	7 fuzzy sets	SVM
Method	Crammer and Singer	One-against-one	Crammer and Singer	Crammer and Singer
Kernel	RBF $\sigma^2 = 50$ C = 0.1	RBF $\sigma^2 = 5$ C = 0.1	RBF $\sigma^2 = 5$ C = 0.1	RBF $\sigma^2 = 10$ C = 10
Coverage	100%	**100%**	96.67%	100.00%
Error	10.67%	**06.00%**	04.67%	02.00%
Number of rules	20.5	**33.5**	39.5	---

4.4 Wine

The last database tested was the wine, which is also a well known benchmark in pattern recognition literature. This database relates to three types of wine produced in a specific region of Italy. There are 13 numeric input attributes and, as usual, one classification output.

For the same reason as the breast cancer case (large number of input variables), the number of pre-defined fuzzy sets for each coordinate was restricted to 3 and 5. Again, since this is a multi-class problem, all three methods were applied for training the SVMs.

Table 4 presents the best classification performance of FREx_SVM in the test patterns, using the same metrics as before.

Table 4. Best Performance of FREx_SVM and SVM - Wine

	3 fuzzy sets	5 fuzzy sets	SVM
Method	One-against-one	One-against-one	One-against-one
Kernel	RBF $\sigma^2 = 10$ C = 0.1	RBF $\sigma^2 = 5$ C = 0.1	RBF $\sigma^2 = 10$ C = 10
Coverage	**92.13%**	51.69%	100.00%
Error	**07.87%**	0%	01.12%
Number of rules	**84**	86	---

As can be seen from Table 4, the best result was obtained with three fuzzy sets, attaining coverage of 92.13%, with 84 rules. Finally, the best rule for each class, regarding its fuzzy accuracy are the following:

IF x_1 is C_{13} and x_2 is C_{22} and x_3 is C_{32} and x_4 is C_{41} and x_5 is C_{52} and x_6 is C_{63} and x_7 is C_{73} and x_8 is C_{82} and x_9 is C_{93} and x_{10} is C_{102} and x_{11} is C_{113} and x_{12} is C_{122} and x_{13} is C_{133}, THEN Class 1 — Accuracy: 1.0000, for 3 fuzzy sets.
IF x_1 is C_{13} and x_2 is C_{22} and x_3 is C_{31} and x_4 is C_{42} and x_5 is C_{52} and x_6 is C_{63} and x_7 is C_{73} and x_8 is C_{82} and x_9 is C_{93} and x_{10} is C_{103} and x_{11} is C_{114} and x_{12} is C_{123} and x_{13} is C_{132}, THEN Class 2 — Accuracy: 1.0000, for 5 fuzzy sets.
IF x_1 is C_{14} and x_2 is C_{25} and x_3 is C_{33} and x_4 is C_{43} and x_5 is C_{53} and x_6 is C_{62} and x_7 is C_{72} and x_8 is C_{84} and x_9 is C_{92} and x_{10} is C_{104} and x_{11} is C_{112} and x_{12} is C_{122} and x_{13} is C_{133}, THEN Class 3 — Accuracy: 1.0000, for 5 fuzzy sets.

5 Conclusions

This paper presented a new method for extracting fuzzy rules from a trained SVM. As has been described, the proposed fuzzy extraction algorithm — FREx_SVM — is based on the support vectors obtained from trained SVMs. FREx_SVM consists of three simple phases, which result in a fast rule extraction algorithm.

The main advantage of the FREx_SVM method is that the generated rules have fuzzy sets in their antecedents, which increases the linguistic interpretability. Additionally, the method can be applied to multi-class problems, enhancing the range of possible applications.

It must be stressed that the main goal of FREx_SVM is to extract interpretable knowledge from a trained SVM. So, the generated rules are not actually used for classifying input patterns, but for understanding how this classification has been accomplished by the SVMs. Therefore, although the percentage errors provided by FREx_SVM in the four benchmark dataset evaluated (Bupa Liver Disorders, Wisconsin Breast Cancer, Iris and Wine databases) are larger than the values provided by the SVMs, they are not really relevant in terms of how well the extracted rules help understand the relation between the input vector and the output classification. The most important metrics are the accuracy and coverage provided by each rule. As demonstrated by the case studies, the resulting values for these two metrics were rather satisfactory, confirming the advantages of the proposed methodology.

As future work, the proposed algorithm shall include adaptive fuzzy sets. This shall improve even further accuracy and coverage of fuzzy rules, and possibly reduce the final number of extracted rules, improving interpretability even further.

References

[1] Cristianini, N. & Shawe-Taylor, J. *An Introduction to Support Vector Machines and other kernel - based learning methods.* Cambridge University Press, 2000. http://www.support-vector.net

[2] Gunn, S. *Support Vector Machines for Classification and Regression.* ISIS Technical Report, 1998. http://www.isis.ecs.soton.ac.uk/research/svm/

[3] Haykin, S *Neural Networks - A Comprehensive Foundation.* Macmillan College Publishing Company, 1999.

[4] Scholkopf, B. & Smola, A. J. *Learning with Kernels.* The MIT Press, 2002.

[5] Vapnik, V. N. An Overview of Statistical Learning Theory. In *IEEE Transactions on Neural Networks*, vol.10 (5), pp. 988-999, 1999.

[6] Vapnik, V. N. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[7] Vapnik, V. N., Golowich, S. E. and Smola, A. Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing System* 9, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. San Mateo, CA: Morgan Kaufmann, pp. 281-287, 1997.

[8] Drucker, H. D., Wu, D. H., Vapnik, V. N. Support vector machines for spam categorization. In *IEEE Trans. on Neural Networks*, vol. 10(5), pp. 1048-1054, 1999.

[9] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pp.137-142, Springer, 1998.

[10] Brown, M. Ps., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M. & Haussler, D. Knowledge-based analysis of microarray gene expression data by using support vector machines. In *Proceedings of The National Academy of Sciences of The United States of America*, vol. 97, pp. 262-267, 2000.

[11] Müller, K. R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V. N. Predicting time series with support vector machines. In *Proc. Int. Conf. Artificial Neural Networks*, 1997.

[12] Fu, X., Ong, C. J., Keerthi, S., Hung, G. G. & Goh, L. Extracting the Knowledge Embedded in Support Vector Machines. In *International Joint Conference on Neural Networks* (IJCNN'04), CDROM, Budapest, July 25-29, 2004.

[13] Nunez, H., Angulo, C. & Catalá, A.: Rule Extraction From support vectors machines. In *European Symposium on Artificial Neural Networks* (ESANN), pp. 107-112, 2002.

[14] Hsu, C.-W. & Lin, C.-J. A Comparison on Methods for Multi-class Support Vector Machines. In *IEEE Transaction on Neural Networks*, vol. 13(2), 2002, pp. 415-425.

[15] Crammer, K. & Singer, Y. On the learnability and design of output codes for multiclass problems. In *Computational Learning Theory*, 2000, pp. 35-46.

[16] Weston, J. & Watkins, C. Multi-class Support Vector Machines. Technical report CSD-TR-98-04, Royal Holloway, 1998.

[17] Rifkin, R., Klautau, A. In Defense of One-Vs-All Classification. *Journal of Machine Learning Research*, vol. 5, 2004, pp 101-141.

[18] Abe, S. & Inoue, T. Fuzzy Support Vector Machines for Multiclass Problems. In *European Symposium on Artificial Neural Networks proceedings* (ESANN), pp 113-118, 2002.

[19] Kressel, U., H.-G. Pairwise classification and support vector machines. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds), *Advances in kernel methods: Support vector learning*, Cambridge, MA: MIT Press, 1999, pp. 225-268.

[20] Chaves, A., Vellasco, M.M.B.R., Tanscheit, R. Fuzzy Rule Extraction from Support Vector Machines, 5th Int. Conf. on Hybrid Intelligent Systems (HIS05), IEEE Computer Society, Los Alamitos, California, EUA, (ISBN 0-7695-2457-5), pp. 335-340, Edited by Nadia Nejah, Luiza M. Mourelle, Marley M.B.R. Vellasco, Ajith Abraham and Mario Köppen, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), November 6-9, Rio de Janeiro – RJ, Brazil.

Density Based Fuzzy Support Vector Machines for Multicategory Pattern Classification

Frank Chung-Hoon Rhee, Jong Hoon Park, and Byung In Choi

Computation Vision and Fuzzy System Laboratory, Department of Electronic Engineering,
Hanyang University, Korea
{frhee, jhpark2, bichoi}@fuzzy.hanyang.ac.kr

Abstract. Support vector machines (SVMs) are known to be useful for separating data into two classes. However, for the multiclass case where pairwise SVMs are incorporated, unclassifiable regions can exist. To solve this problem, Fuzzy support vector machines (FSVMs) was proposed, where membership values are assigned according to the distance between patterns and the hyperplanes obtained by the "crisp" SVM. However, they still may not give proper decision boundaries for arbitrary distributed data sets. In this paper, a density based fuzzy support vector machine (DFSVM) is proposed, which incorporates the data distribution in addition to using the memberships in FSVM. As a result, our proposed algorithm may give more appropriate decision boundaries than FSVM. To validate our proposed algorithm, we show experimental results for several data sets.

Keywords: Density, Multiclass problems, Membership functions, FSVM.

1 Introduction

Support vector machines (SVM) have been used in several classification related applications and have shown to perform well over traditional learning methods [1], [2]. Since SVMs are mainly used for two class problems, there may exist some difficultly in extending them to the multiclass case. One possible extension can be achieved by considering $k(k-1)/2$ pairwise SVMs [3]. However, this method can result in having unclassifiable regions. To resolve these unclassifiable regions, a fuzzy support vector machine (FSVM) was proposed, where fuzzy memberships are assigned according to the distance between the patterns and hyperplanes obtained by SVM [5], [6], [7]. Since, FSVM does not consider the distribution of the data they may not give proper decision boundaries for data sets that are density dependent. In this paper, we propose a density based fuzzy support vector machine (DFSVM) which adjusts the decision boundary according to the density of data and the distance between the nearest patterns and hyperplanes obtained by SVM. As a result, the decision boundary obtained by our proposed algorithm can be more suitable for data that are generally distributed.

The reminder of this paper is organized as follows. In Section 2, we summarize the discussion on SVM. In Section 3, we overview the FSVM method that was proposed to resolve unclassifiable regions in multiclass problems. In Section 4, we discuss our proposed algorithm. In Section 5, we give several experiments results to show the validity of our proposed algorithm. Finally Section 6 gives the conclusions.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 109–118, 2007.
springerlink.com

2 Support Vector Machines

In general, a support vector machine (SVM) is considered to be a binary classifier, which separates a data set into two classes. For a two class data set that is linearly separable (see Fig. 1), SVM finds the maximum margin hyperplane that separates the data according to the maximum distance (margin) between the hyperplane and the nearest data from the hyperplane. In this case, the hyperplane is considered as the optimal separating hyperplane.

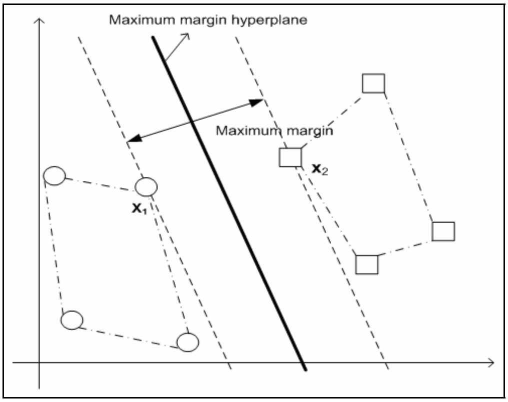


Fig. 1. Illustration of an optimal separating hyperplane

To be specific, consider training data $\{\mathbf{x}_i, y_i\}$ to be categorized into two classes, where the label of the training data y_i are 1 and -1 for class 1 and 2, respectively. Given a set of N patterns, the optimal separating hyperplane defined as $D(\mathbf{x})=\mathbf{w}\cdot\mathbf{x}+b$ can be obtained by solving the following quadratic programming problem [4].

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} \quad y_i(\mathbf{w}^{\mathrm{t}}\mathbf{x}_i + b) \geq 1, \qquad i = 1, \ldots, N, \end{aligned} \tag{2.1}$$

where parameter $\mathbf{w}$ and b denote the weight vector and bias, respectively. Eq. (2.1) can be simplified by maximizing the following object function

$$\begin{aligned} &\text{maximize} \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j, \\ &\text{subject to} \quad \sum_{i=1}^{N} y_i \alpha_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \ldots, N, \end{aligned} \tag{2.2}$$

where α is a Lagrange multiplier. By solving Eq. (2.2), the optimal weight vector $\mathbf{w}^*$and bias b^* can be obtained as

$$\mathbf{w}^* = \sum_{i=1}^{N} y_i \alpha_i \mathbf{x}_i, \tag{2.3}$$

$$b^* = -\frac{1}{2}\left[\max_{y_i=-1}(\mathbf{w}^* \cdot \mathbf{x}_i) + \min_{y_i=1}(\mathbf{w}^* \cdot \mathbf{x}_i)\right] \tag{2.4}$$

From these results, we can obtain the decision function $D(\mathbf{x})$. When $D(\mathbf{x})>0$, pattern $\mathbf{x}$ is classified as class 1. For the case where the two class data are not linearly separable, we can include a slack variable ξ_i in Eq. (2.1) which indicates the non-separable degree. In addition, to further enhance the linear separability, we can use a transformation function $\phi(\cdot)$ to map the input data into a high dimensional feature space. If the transformation is positive definite, the dot product in the feature space can be expressed by the following kernel function as

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(\mathbf{x}_i \cdot \mathbf{x}_j). \tag{2.5}$$

Therefore, Eq. (2.2) can now be expressed as

$$\begin{aligned} &\text{maximize} \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{subject to} \quad \sum_{i=1}^{N} y_i \alpha_i = 0, \quad C \geq \alpha_i \geq 0, \quad i = 1, \dots, N, \end{aligned} \tag{2.6}$$

where C is positive constant. By solving Eq. (2.6), the optimal hyperplane that separates the transformed data in a high dimensional feature space can be obtained.

For multiclass problems, one possible extension can be achieved by solving $k(k-1)/2$ pairwise SVMs. The decision function for the pair class i and j is given by

$$D_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^{\mathrm{t}} \mathbf{x} + b_{ij}, \tag{2.7}$$

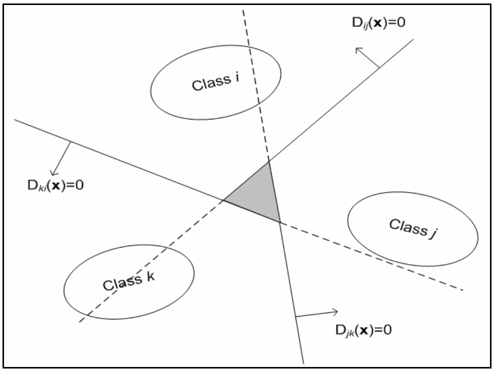


Fig. 2. Unclassifiable region due to pairwise formulation

where $D_{ij}(\mathbf{x})$=-$D_{ji}(\mathbf{x})$ and has maximum margin. For training data **x**, we can obtain

$$D_i(\mathbf{x}) = \sum_{j\neq 1, i=1}^{k} \text{sign}(D_{ij}(\mathbf{x})), \text{ where } \text{sign}(\cdot) = \begin{cases} 1 & \text{if } (\cdot)>0 \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

and categorize **x** into the class by

$$\arg \max_{i=1,\ldots,k} D_i(\mathbf{x}). \tag{2.9}$$

However, when pattern **x** in Eq. (2.9) is satisfied for two or more D_i, **x** is unclassifiable. The shaded region in Fig. 2 shows where patterns are unclassifiable.

3 Fuzzy Support Vector Machines

To resolve the unclassifiable region for the multiclass problem as shown in Fig. 2, fuzzy support vector machines (FSVM) were introduced [5], [6], [7]. In the FSVM, fuzzy membership function $m_{ij}(\mathbf{x})$ is defined in the direction perpendicular to the decision boundary (optimal separating hyperplane) $D_{ij}(\mathbf{x})$ as

$$m_{ij}(\mathbf{x}) = \begin{cases} 1 & \text{for } D_{ij}(\mathbf{x}) \geq 1, \\ D_{ij}(\mathbf{x}) & \text{otherwise} \end{cases} \tag{3.1}$$

$$D_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^{t}\mathbf{x} + b_{ij} \text{ and } D_{ij}(\mathbf{x}) = -D_{ji}(\mathbf{x}).$$

As indicated in Eq. (3.1), negative degrees of membership are allowed. The membership of **x** for class *i* can be defined using the minimum operator as

$$m_i(\mathbf{x}) = \min_{j=1,\ldots,k} m_{ij}(\mathbf{x}). \tag{3.2}$$

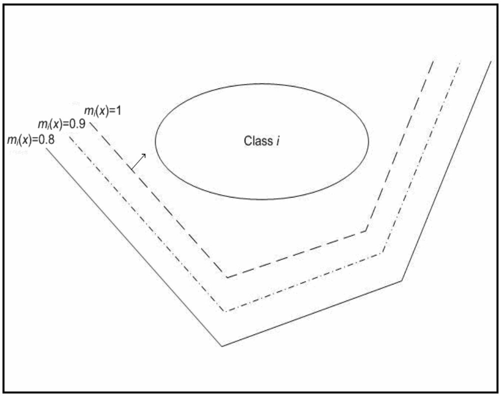


Fig. 3. Contour lines of membership functions

Fig. 3 shows an example of the contour lines of class i membership functions. Using Eq. (3.2), pattern $\mathbf{x}$ can be classified by

$$\arg\max_{i=1,\ldots,k} m_i(\mathbf{x}). \tag{3.3}$$

According to Eq. (3.3), the unclassifiable region in Fig. 2 can be resolved as shown in Fig. 4.

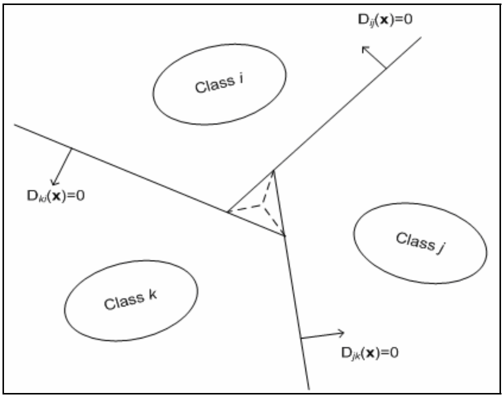


Fig. 4. Unclassifiable region in Fig. 2 resolved by the minimum operator

4 Density Based Fuzzy Support Vector Machines

In FSVM, memberships are used to resolve possible unclassifiable regions in multiclass problems. For some data sets that differ significantly, the optimal class separating hyperplanes can result to be the same for each data set. Hence, undesirable partitioning of the classes may occur. To avoid this, we propose a density based fuzzy support vector machine (DFSVM) to obtain decision boundaries which considers not only the optimal class separating hyperplanes from FSVM but also the density of the distribution of the patterns. As a result, the pattern density of each class can alter the decision boundaries in the unclassifiable region and therefore can provide desirable decision boundaries for a given data distribution. In our proposed algorithm, the class pattern densities are obtained using the parzen window technique [8]. When region R encloses a number of patterns, density $p_n(\boldsymbol{x})$ can be expressed as

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right), \tag{4.1}$$

where h_n is the length of an edge of a hypercube, V_n is the volume of the hypercube, and n is number of patterns. The window function $\varphi(u)$ can be defined as

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2} \tag{4.2}$$

The choice of h_n(or V_n) has an important effect on $p_n(\boldsymbol{x})$. If V_n is too large, the estimate will suffer from too little resolution. However, if V_n is too small the estimate will suffer from too much statistical variability. Using Eq. (4.1), the decision boundary $D_{ij}(\boldsymbol{x})$ in Eq. (3.1) can be modified as

$$D_{ij}(\mathbf{x})=(1-\alpha)(\mathbf{w}_{ij}^{\mathrm{t}}\mathbf{x}+b_{ij})+\alpha(p_{n_i}(\mathbf{x})-p_{n_j}(\mathbf{x})), \tag{4.3}$$

where i and j denote the class pair and α is a parameter that determines the weight between the FSVM and the density. Using Eq. (4.3) and Eq. (3.1), we define a new $m_{ij}(\boldsymbol{x})$. Using the minimum operator, the membership of **x** for class i can be defined as in Eq. (3.2) and pattern **x** can be classified using Eq. (3.3).

5 Experiments Results

In order to validate our proposed method, we give an example to illustrate the problems associated with the FSVM and compare the results with our proposed algorithm. In addition, we show classification results for high dimensional data sets and image segmentation.

5.1 Three Class Data

We generate a data set that consists of 255 patterns of two features and three classes (98, 93, 64 patterns for class 1, 2, 3) to illustrate the effectiveness of our proposed DFSVM. Fig. 5 shows the decision boundaries obtained by FSVM. As indicated in the figure, the data distribution of class 3 is not uniformly distributed as with the other

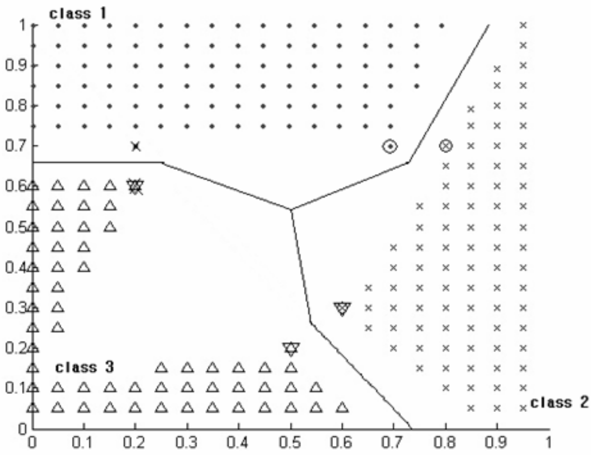


Fig. 5. FSVM decision boundaries for three class data

class patterns. Fig. 6 shows a visual representation of the memberships for each class that leads in determining the class boundaries, where brighter intensities indicate higher memberships. Since the densities of the class patterns are not taken under consideration in determining the class boundaries by FSVM, the boundaries will remain the same regardless of the distribution of patterns in the empty area that are not in the unclassified region as indicated earlier. Therefore the decision boundaries may not be suitable for some particular data distributions. Fig. 7 shows the decision boundaries resulting from our proposed DFSVM for several combinations of weight parameter α and parzen window size. As indicated, since class 3 patterns are of smaller density compared to the other classes, the decision boundaries in the unclassified region moves toward class 3. As a result, we can achieve more appropriate decision boundaries than FSVM.

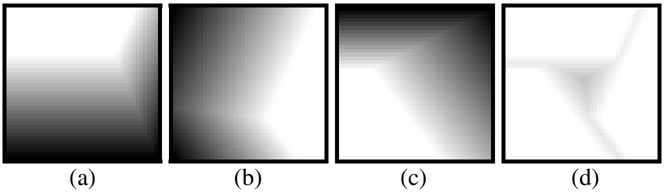
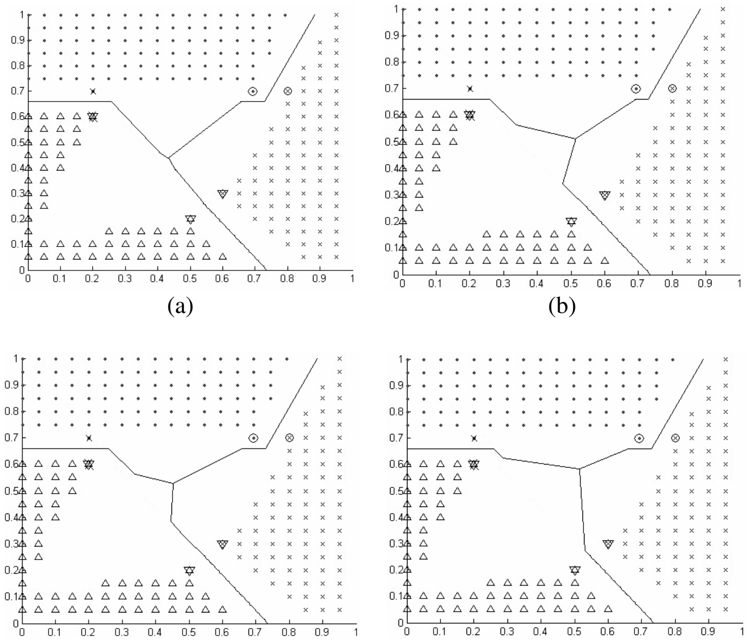

(a) (b) (c) (d)

Fig. 6. Visual representation of decision boundaries: (a) m_1, (b) m_2, (c) m_3, and (d) max $m_i(x)$



(a) (b)

(c) (d)

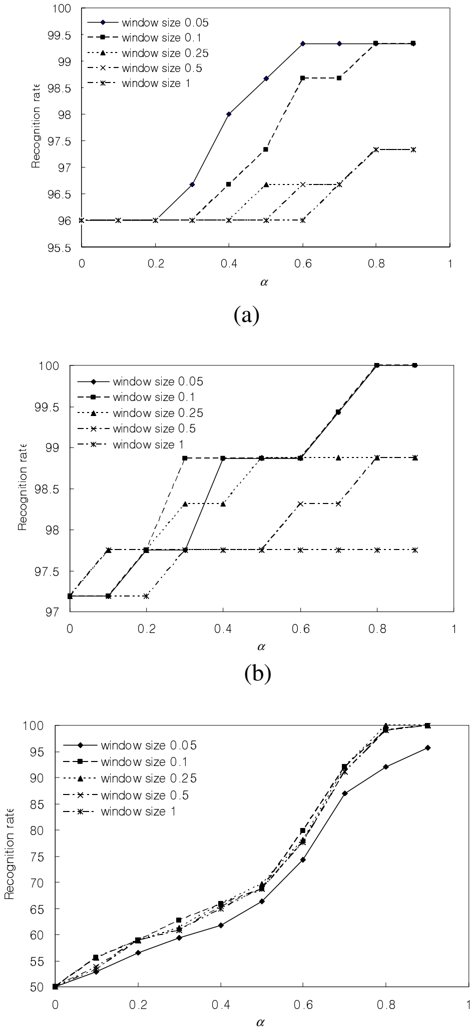
Fig. 7. Decision boundaries resulting from DFSVM: (a) α=1 and window size 0.1, (b) α=0.9 and window size 0.1, (c) α=1 and window size 0.5, and (d) α=0.9 and window size 0.5

5.2 Classification Results for High Dimensional Data

We now show classification results for high dimensional data sets as listed in Table 1.

Table 1. Benchmark data

Data	Number of data	Class	Feature
Iris	150	3	4
Wine	178	3	13
Glass	214	6	16



(a)

(b)

(c)

Fig. 8. Recognition rate results: (a) iris data, (b) wine data, and (c) glass data

We use 1 pattern for testing and the remaining N-1 patterns for training. This is repeated N times. We show the recognition rate by varying the weight α in D_{ij} and the parzen window size. We use a RBF kernel and fix parameters C=100 and σ=1. In Fig. 8, when α is 0 the results are the same as FSVM. As shown in Fig. 8, by varying α and window size our proposed DFSVM outperforms FSVM for all cases. These results indicate that the decision boundaries obtained by our proposed algorithm are more appropriate for data sets that are not uniformly distributed than those obtained by FSVM. In addition, as $\alpha \rightarrow 1$ the recognition rate improves.

5.3 Image Segmentation

For our final example we give segmentation results for 200×200 real image data. We uniformly extract 300 sample data (100 samples from each region) using two features (intensity and excess green, not shown). We set parameter C=10. The window size that gives the best results in DFSVM is 0.05. Results show that our proposed DFSVM gives a better segmentation than FSVM. The results are shown in Fig. 9.

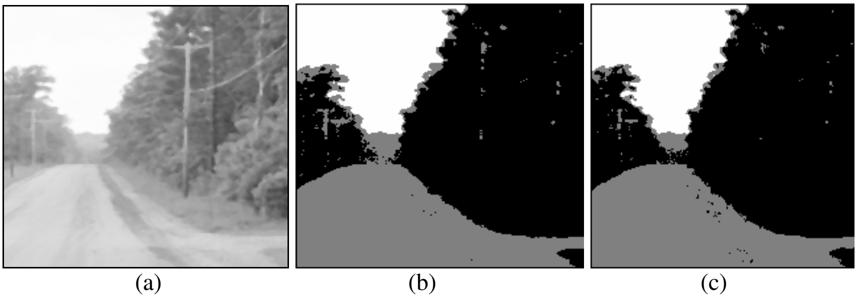


Fig. 9. Results of image segmentation for real image data: (a) original image, (b) FSVM, and (c) DFSVM

6 Conclusions

FSVM was developed to resolve unclassifiable regions that can occur in multiclass problems by incorporating fuzzy memberships. However, FSVM may not obtain suitable decision boundaries for data sets that are not uniformly distributed since the memberships are determined only by the margin of the hyperplanes and not by the density of the class patterns. To improve this, we proposed a density based fuzzy support vector machine (DFSVM), which considers the data distribution in addition to using the results in FSVM. As a result, DFSVM can be considered to be more suitable for obtaining decision boundaries for arbitrarily distributed data sets. We gave classification results for several data sets to show the validity of our proposed algorithm. However, optimal methods for selecting the appropriate weight α and parzen window size are desired. This is currently under investigation.

Acknowledgements. This research was supported by the Agency for Defense Development, Korea, through the Image Information Research Center at Korea Advanced Institute of Science & Technology.

References

1. Chekassky, V., Mulier, F.: Learning from Data Concepts, Theory, and Method. John Wiley & Sons (1998)
2. Vapnik, V. N.: Statistical Learning Theory. John Wiley & Sons (1998)
3. Kreßel, U. H.-G.: Pairwise Classification and Support Vector Machines. In SchÖlkopf, B., C. J. C. eurges and Smola (eds.): Advances in kernel methods: Support Vector Learning. Cambridge, MA: MIT Press (1999) 255-268
4. Nello, C., John, S.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)
5. Inoue, T., Abe, S.: Fuzzy Support Vector Machines for Pattern Classification. Proceedings of the International Joint Conferernce on Neural Networks (2000) 1449-1454
6. Abe, S., Inoue, T.: Fuzzy Support Vector Machines for Multiclass Problems. Proceedings of the Tenth European Symposium on Aritificial Neural Networks (2002) 113-369
7. Daisuke, T., Shingeo, A.: Fuzzy Least Squares Support Vector Machines for Multiclass Problems. Neural Networks, 16 (2003) 785-792
8. Duda, R., Hart, P., Stork, D.: Pattern Classification. 2nd edn. Wiley (2001)

A Modified FCM Algorithm for Fast Segmentation of Brain MR Images

L. Szilágyi[1,2], S.M. Szilágyi[2], and Z. Benyó[1]

[1] Budapest University of Technology and Economics, Dept. of Control Engineering and Information Technology, Budapest, Hungary
[2] Sapientia – Hungarian Science University of Transylvania, Faculty of Technical and Human Sciences, Târgu Mureş, Romania
alo@ms.sapientia.ro

Abstract. Automated brain MR image segmentation is a challenging problem and received significant attention lately. Several improvements have been made to the standard fuzzy c-means (FCM) algorithm, in order to reduce its sensitivity to Gaussian, impulse, and intensity non-uniformity noises. In this paper we present a modified FCM algorithm, which aims accurate segmentation in case of mixed noises, and performs at a high processing speed. The proposed method extracts a scalar feature value from the neighborhood of each pixel, using a filtering technique that deals with both spatial and gray level distances. These features are classified afterwards using the histogram-based approach of the enhanced FCM classifier. The experiments using synthetic phantoms and real MR images show, that the proposed method provides better results compared to other reported FCM-based techniques.

1 Introduction

By definition, image segmentation represents the partitioning of an image into non-overlapping, consistent regions, which appear to be homogeneous with respect to some criteria concerning gray level intensity and/or texture.

The fuzzy c-means (FCM) algorithm is one of the most widely used method for data clustering, and probably also for brain image segmentation [2, 3, 7]. However, in this latter case, standard FCM is not efficient by itself, as it is unable to deal with that relevant property of images, that neighbor pixels are strongly correlated. Ignoring this specificity leads to strong noise sensitivity and several other imaging artifacts.

Recently, several solutions were given to improve the performance of segmentation. Most of them involve using local spatial information: the own gray level of a pixel is not the only information that contributes to its assignment to the chosen cluster. Its neighbors also have their influence while getting a label. Pham and Prince [9] modified the FCM objective function by including a spatial penalty, enabling the iterative algorithm to estimate spatially smooth membership functions. Ahmed et al. [1] introduced a neighborhood averaging additive term into the objective function of FCM, calling the algorithm bias corrected FCM (BCFCM). This approach has its own merits in bias field estimation, but it computes the neighborhood term in every iteration step, giving the algorithm a serious computational load. Moreover, the zero gradient condition at the estimation of the bias term produces a significant amount of misclassifications [11]. Chuang et al. [6] proposed averaging the fuzzy membership

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 119–127, 2007.
springerlink.com

function values and reassigning them according to a tradeoff between the original and averaged membership values. This approach can produce accurate clustering if the tradeoff is well adjusted empirically, but it is enormously time consuming.

In order to reduce the execution time, Szilágyi et al. [12], and Chen and Zhang [5] proposed to evaluate the neighborhoods of each pixel as a pre-filtering step, and perform FCM afterwards. The averaging and median filters, followed by FCM clustering, are referred to as FCM_S1 and FCM_S2, respectively [5]. Paper [12] also pointed out, that once having the neighbors evaluated, and thus for each pixel having extracted a one-dimensional feature vector, FCM can be performed on the basis of the gray level histogram, clustering the gray levels instead of the pixels, which significantly reduces the computational load, as the number of gray levels is generally smaller by orders of magnitude. This latter quick approach, combined with an averaging pre-filter, is referred to as enhanced FCM (EnFCM) [4, 12]. All BCFCM, FCM_S1, and EnFCM suffer from the presence of a parameter denoted by α, which controls the strength of the averaging effect, balances between the original and averaged image, and whose ideal value unfortunately can be found only experimentally. Another drawback is the fact, that averaging and median filtering, besides eliminating salt-and pepper noises, also blurs relevant edges. Due to these shortcomings, Cai et al. [4] introduced a new local similarity measure, combining spatial and gray level distances, and applied it as an alternative pre-filtering to EnFCM, having this approach named fast generalized FCM (FGFCM). This approach is able to extract local information causing less blur then the averaging or median filter, but still has an experimentally adjusted parameter λ_g, which controls the effect of gray level differences.

Another remarkable approach, proposed by Pham [10], modifies the objective function of FCM by the means of an edge field, in order to eliminate the filters that produce edge blurring. This method is also significantly time consuming, because the estimation of the edge field, which is performed as an additional step in each iteration, has no analytical solution.

In this paper we propose a novel method for MR brain image segmentation that simultaneously aims high accuracy in image segmentation, low noise sensitivity, and high processing speed.

2 Methods

2.1 Standard Fuzzy C-Means Algorithm

The fuzzy c-means algorithm has successful applications in a wide variety of clustering problems. The traditional FCM partitions a set of object data into a number of c clusters based on the minimization of a quadratic objective function. When applied to segment gray level images, FCM clusters the intensity level of all pixels (x_k, $k = 1 \ldots n$), which are scalar values. The objective function to be minimized is:

$$J_{FCM} = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} (x_k - v_i)^2, \tag{1}$$

where $m > 1$ is the fuzzyfication parameter, v_i represents the prototype or centroid value of cluster i, and $u_{ik} \in [0,1]$ is the fuzzy membership function showing the degree to which pixel k belongs to cluster i. According to the definition of fuzzy sets, for any pixel k, we have $\sum_{i=1}^{c} u_{ik} = 1$. The minimization of the objective function is reached by alternately applying the optimization of J_{FCM} over $\{u_{ik}\}$ with v_i fixed, $i = 1 \ldots c$, and the optimization of J_{FCM} over $\{v_i\}$ with u_{ik} fixed, $i = 1 \ldots c$, $k = 1 \ldots n$ [3]. During each cycle, the optimal values are computed from the zero gradient conditions, and obtained as follows:

$$u_{ik}^{*} = \frac{(x_k - v_i)^{-2/(m-1)}}{\sum_{j=1}^{c} (x_k - v_j)^{-2/(m-1)}} \text{ and } v_i^{*} = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}}, \tag{2}$$

for any $i = 1 \ldots c$ and $k = 1 \ldots n$. After adequate initialization of cluster prototypes v_i, the above equations are applied alternately until the norm of the variation of vector $\mathbf{v}$ is less then a previously set small value ε.

FCM has invaluable merits in making optimal clusters, but in image processing it has severe deficiencies. The most important one is the fact that it fails to take into consideration the position of pixels, which is also relevant information in image segmentation. This drawback induced the introduction of spatial constraints into fuzzy clustering.

2.2 Fuzzy Clustering Using Spatial Constraints

Ahmed et al. [1] proposed a modification to the objective function of the traditional FCM, in order to allow the labeling of a pixel to be influenced by its immediate neighbors. This neighboring effect acts like a regularizer that biases the solution to a piecewise homogeneous labeling [1]. The objective function of BCFCM is:

$$J_{BCFCM} = \sum_{i=1}^{c} \sum_{k=1}^{n} \left[u_{ik}^{m} (x_k - v_i)^2 + \frac{\alpha}{n_k} \sum_{r \in N_k} u_{ik}^{m} (x_r - v_i)^2 \right] \tag{3}$$

where x_r represents the gray level of pixels situated in the neighborhood N_k of pixel k, and n_k is the cardinality of N_k. The parameter α controls the intensity of the neighboring effect, and unfortunately its optimal value can be found only experimentally. Having the neighbors computed in every computation cycle, this iterative algorithm performs extremely slowly. Chen and Zhang [5] reduced the time complexity of BCFCM, by previously computing the neighboring averaging term or replacing it by a median filtered term. The obtained algorithms were named FCM_S1 and FCM_S2, respectively. These algorithms outperformed BCFCM, mostly from the point of view of time complexity.

Szilágyi et al. [12] proposed a regrouping of the processing steps of BCFCM. In their approach, an averaging filter is applied first, similarly to the neighboring effect of Ahmed et al. [1]:

$$\xi_k = \frac{1}{1+\alpha}\left(x_k + \frac{\alpha}{n_k} \sum\nolimits_{r \in N_k} x_r \right), \tag{4}$$

where ξ_k denotes the filtered gray value of pixel k . This filtering is followed by an accelerated version of FCM clustering. The acceleration is based on the idea, that the number of gray levels is generally much smaller than the number of pixels. In this order, the histogram of the filtered image is computed, and not the pixels, but the gray levels are clustered [12], by minimizing the following objective function:

$$J_{EnFCM} = \sum\nolimits_{i=1}^{c} \sum\nolimits_{l=1}^{q} h_l u_{il}^m (l - v_i)^2 \ . \tag{5}$$

In this formula, h_l denotes the number of pixels having the gray level equal to l , $l \in \{1,2,\ldots,q\}$, where q is the number of gray levels. The optimization formulae in this case, for any $i = 1\ldots c$ and $l = 1\ldots q$, will be:

$$u_{ik}^* = \frac{(l - v_i)^{-2/(m-1)}}{\sum\nolimits_{j=1}^{c} (l - v_j)^{-2/(m-1)}} \text{ and } v_i^* = \frac{\sum\nolimits_{l=1}^{q} h_l u_{il}^m l}{\sum\nolimits_{l=1}^{q} h_l u_{il}^m} \ . \tag{6}$$

EnFCM drastically reduces the computation complexity of BCFCM and its relatives [4, 12]. If the averaging pre-filter is replaced by a median filter, the segmentation accuracy also improves significantly [4, 13, 14].

2.3 Fuzzy Clustering Using Spatial and Gray Level Constraints

Based on the disadvantages of the aforementioned methods, but inspired of their merits, Cai et al. [4] introduced a local (spatial and gray) similarity measure that they used to compute weighting coefficients for an averaging pre-filter. The filtered image is then subject to EnFCM-like histogram-based fast clustering. The similarity between pixels k and r is given by the following formula:

$$S_{kr} = \begin{cases} s_{kr}^{(s)} \times s_{kr}^{(g)}, & r \in N_k - \{k\} \\ 0, & r = k \end{cases} , \tag{7}$$

where $s_{kr}^{(s)}$ and $s_{kr}^{(g)}$ are the spatial and gray level components, respectively. The spatial term $s_{kr}^{(s)}$ is defined as the L_∞ -norm of the distance between pixels k and r . The gray level term is computed as

$$s_{kr}^{(g)} = \exp\left(\frac{-(x_k - x_r)^2}{\lambda_g \sigma_k^2} \right), \tag{8}$$

where σ_k denotes the average quadratic gray level distance between pixel k and its neighbors. Segmentation results are reported more accurate then in any previously presented case [4].

2.4 The Proposed Method

Probably the most relevant problem of all techniques presented above, BCFCM, EnFCM, FCM_S1, and FGFCM, is the fact that they depend on at least one parameter, whose value has to be found experimentally. The parameter α balances the effect of neighboring in case of the former three, while λ_g controls the tradeoff between spatial and gray level components in FGFCM.

The zero value in the second row of Eq. (7) implies, that in FGFCM, the filtered gray level of any pixel is computed as a weighted average of its neighbor pixel intensities. Having renounced to the original intensity of the current pixel, even if it is a reliable, noise-free value, unavoidably produces some extra blur into the filtered image. Accurate segmentation requires this kind of effects to be minimized [10].

In this paper we propose a set of modifications to EnFCM/FGFCM, in order to improve the accuracy of segmentation, without renouncing to the speed of histogram-based clustering. In other words, we need to define a complex filter that can extract relevant feature information from the image while applied as a pre-filtering step, so that the filtered image can be clustered fast afterwards based on its histogram. The proposed method consists of the following steps:

1. As we are looking for the filtered value of pixel k, we need to define a small square or diamond-shape neighborhood N_k around it. Square windows of size 3×3 were used throughout this study, but other window sizes and shapes (e.g. diamond) are also possible.
2. We search for the minimum, maximum, and median gray value within the neighborhood N_k, and we denote them by min$_k$, max$_k$ and med$_k$, respectively.
3. We replace the gray level of the maximum and minimum valued pixel with the median value (if there are more then one maxima or minima, replace them all), unless they are situated in the middle pixel k. In this latter case, pixel k remains unchanged, just labeled as unreliable value.
4. Compute the average quadratic gray level difference of the pixels within the neighborhood N_k, using the formula

$$\sigma_k = \sqrt{\frac{\sum_{r \in N_k, r \neq k} (x_r - x_k)^2}{n_k - 1}} . \tag{9}$$

5. The filter coefficients will be defined as:

$$C_{kr} = \begin{cases} c_{kr}^{(s)} \times c_{kr}^{(g)}, & r \in N_k - \{k\} \\ 1, & r = k \wedge x_k \notin \{\max_k, \min_k\} \\ 0, & r = k \wedge x_k \in \{\max_k, \min_k\} \end{cases}. \qquad (10)$$

The central pixel k will have coefficient 0 if its value was found unreliable, otherwise it has unitary coefficient. All other neighbor pixels will have coefficients $C_{kr} \in [0,1]$, depending on their space distance and gray level difference from the central pixel. In case of both terms, higher distance values will push the coefficients towards 0.

6. The spatial component $c_{kr}^{(s)}$ is a negative exponential of the Euclidean distance between the two pixels k and r: $c_{kr}^{(s)} = \exp(-L_2(k,r))$. The gray level term is defined as follows:

$$c_{kr}^{(g)} = \begin{cases} 0.5 + 0.5 \times \cos(\pi(x_r - x_k)/(4\sigma_k)), & |x_r - x_k| \le 4\sigma_k \\ 0, & |x_r - x_k| > 4\sigma_k \end{cases}. \qquad (11)$$

The above function has a bell-like shape within the interval $[-4\sigma_k, 4\sigma_k]$.

7. The extracted feature value for pixel k, representing its filtered intensity value, is obtained as a weighted average of its neighbors:

$$\tilde{x}_k = \frac{\sum_{r \in N_k} C_{kr} x_r}{\sum_{r \in N_k} C_{kr}}. \qquad (12)$$

Algorithm

We can summarize the proposed method as follows:

1. Pre-filtering step: for each pixel of the input image, compute the filtered gray level value, using Eqs. (9), (10), (11), (12).
2. Compute the histogram of the pre-filtered image, get the values h_l, $l = 1 \dots q$.
3. Initialize v_i with valid gray level values, differing from each other.
4. Compute new u_{il} fuzzy membership values, $i = 1 \dots c$, $l = 1 \dots q$, and than new v_i prototype values for the clusters, $i = 1 \dots c$, using Eq. (6).
5. If there is relevant change in the v_i values, go back to step 4. This is tested by comparing any norm of the difference between the new and the old vector $\mathbf{v}$ with a preset small constant ε.

The algorithm converges quickly. However, the number of necessary iterations depends on ε and on the initial values of v_i.

3 Results and Discussion

In this section we test and compare the accuracy of four algorithms: BCFCM, EnFCM, FGFCM, and the proposed method, on some synthetic and real images. All the following experiments used 3×3 window size for all kinds of filtering.

The noise removal performances were compared using a 256×256-pixel synthetic test image taken from IBSR [8] (see Fig. 1(a)). The rest of Fig. 1 also shows how much these methods were affected by a high degree mixed noise. Visually, the proposed method achieves best results, slightly over FGFCM, and significantly over all others.

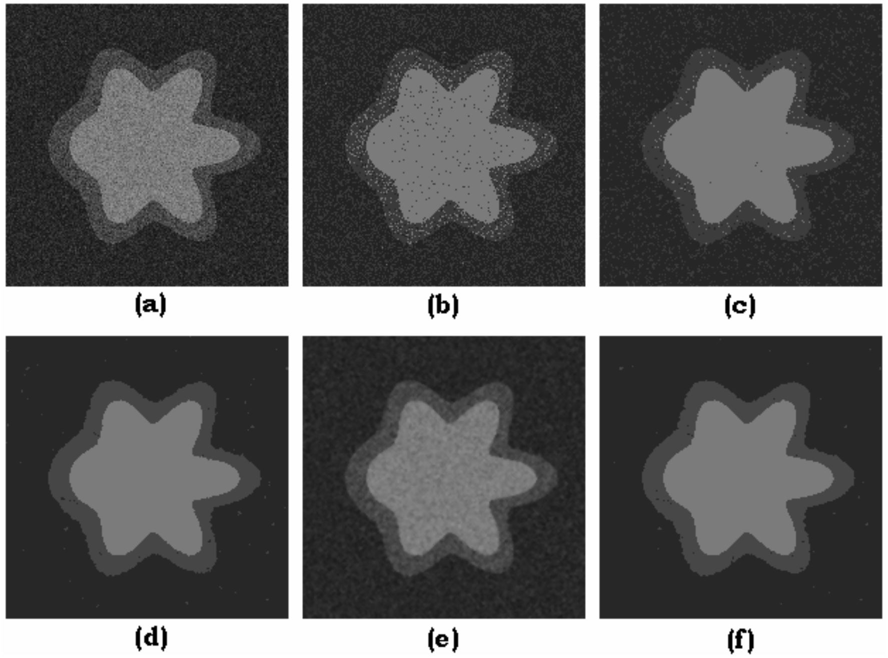


Fig. 1. Segmentation results on phantom images: (a) original, (b) segmented with traditional FCM, (c) segmented using BCFCM, (d) segmented using FGFCM, (e) filtered using the proposed pre-filtering, (f) result of the proposed segmentation

Table 1 gives a statistical analysis of the synthetic images contaminated with different noises (Gaussian noise, salt-and-pepper impulse noise, and mixtures of these) at different levels. The table reveals that the proposed filter performs best at removing all these kinds of noises. Consequently, the proposed method is suitable for segmenting images corrupted with unknown noises, and in all cases it performs at least as well as his ancestors.

Table 1. Segmentation results on synthetic test images, at different noise levels

Noise type	BCFCM	EnFCM	FGFCM	Proposed
Gaussian 1	99.768 %	99.808 %	99.898 %	99.910 %
Gaussian 2	92.403 %	99.028 %	99.595 %	99.670 %
Gaussian 3	81.455 %	97.353 %	97.025 %	97.845 %
Impulse	99.750 %	99.808 %	99.870 %	99.880 %
Mixed 1	99.655 %	99.780 %	99.870 %	99.890 %
Mixed 2	94.458 %	98.975 %	99.325 %	99.453 %

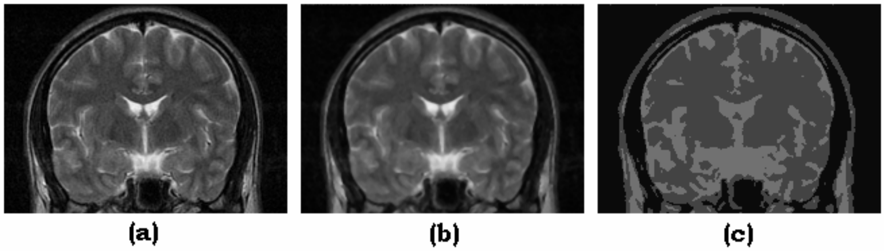


Fig. 2. Segmentation results on real MR images: (a) original, (b) filtered using the proposed method, (c) result of the proposed segmentation

We applied the proposed segmentation method to several complete head MR scans in IBSR. The dimensions of the image stacks were $256 \times 256 \times 64$ voxels. The average total processing time was around 10 seconds on a 2.2 GHz Pentium 4. Fig. 2 presents one slice of real, T2-weighted MR brain image, and its segmentation using the proposed method. Visual inspection shows, that our segmentation results are very close to the IBSR expert's manual inspection.

4 Conclusions

We have developed a modified FCM algorithm for automatic segmentation of MR brain images. The algorithm was presented as a combination of a complex pre-filtering technique and an accelerated FCM clustering performed over the histogram of the filtered image. The pre-filter uses both spatial and gray level criteria, in order to achieve efficient removal of Gaussian and impulse noises without significantly blurring the real edges.

Experiments with synthetic phantoms and real MR images show, that our proposed technique accurately segments the different tissue classes under serious noise contamination. We compared our results with other recently reported methods. Test results revealed that our approach outperformed these methods in many aspects, especially in the accuracy of segmentation and processing time.

Further works aim to reduce the sensitivity of the proposed technique to intensity non-uniformity noises, and to introduce adaptive determination of the optimal number of clusters.

References

1. Ahmed MN, Yamany SM, Mohamed N, Farag AA, Moriarty T (2002) A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Trans Med Imag 21:193–199
2. Bezdek JC, Keller JM, Krishnapuram R, Pal NR (1999) Fuzzy models and algorithms for pattern recognition and image processing, Norwell, MA:Kluwer
3. Bezdek JC, Pal SK (1991) Fuzzy models for pattern recognition. Piscataway, NJ: IEEE Press
4. Cai W, Chen S, Zhang DQ (2007) Fast and robust fuzzy c-means algorithms incorporating local information for image segmentation. Patt Recogn 40:825–838
5. Chen S, Zhang DQ (2004) Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Trans Syst Man Cybern Part B 34:1907–1916
6. Chuang KS, Tzeng HL, Chen S, Wu J, Chen TJ (2006) Fuzzy c-means clustering with spatial information for image segmentation. Comp Med Imag Graph 30:9–15
7. Hathaway RJ, Bezdek JC, Hu Y (2000) Generalized fuzzy c-means clustering strategies using L_p norm distances. IEEE Trans Fuzzy Syst 8:576–582
8. Internet Brain Segmentation Repository, http://www.cma.mgh.harvard.edu /ibsr
9. Pham DL, Prince JL (1999) Adaptive fuzzy segmentation of magnetic resonance images. IEEE Trans Med Imag 18:737–752
10. Pham DL (2003) Unsupervised tissue classification in medical images using edge-adaptive clustering. Proc 25th Ann Int Conf IEEE EMBS, pp 634–637
11. Siyal MY, Yu L (2005) An intelligent modified fuzzy c-means based algorithm for bias field estimation and segmentation of brain MRI. Patt Recogn Lett 26:2052–2062
12. Szilágyi L, Benyó Z, Szilágyi SM, Adam HS (2003) MR brain image segmentation using an enhanced fuzzy c-means algorithm. Proc 25th Ann Int Conf IEEE EMBS, pp 724–726
13. Szilágyi L, Szilágyi SM, Benyó Z (2006) Automated medical image processing methods for virtual endoscopy. Proc World Congr Med Phys Biomed Eng, pp 2267–2270
14. Szilágyi L (2007) Medical image processing methods for the development of a virtual endoscope. Period Polytech Electr Eng (TU Budapest), in press

Incorporation of Non-euclidean Distance Metrics into Fuzzy Clustering on Graphics Processing Units

Derek Anderson[1,2,*], Robert H. Luke[1,2], and James M. Keller[1]

[1] Department of Electrical and Computer Engineering, University of Missouri-Columbia
*dtaxtd@missouri.edu
[2] Biomedical and Health Informatics Research Training Program, National Library of Medicine
rhl3db@missouri.edu,
kellerj@missouri.edu

Abstract. Computational tractability of clustering algorithms becomes a problem as the number of data points, feature dimensionality, and number of clusters increase. Graphics Processing Units (GPUs) are low cost, high performance stream processing architectures used currently by the gaming, movie, and computer aided design industries. Fuzzy clustering is a pattern recognition algorithm that has a great amount of inherent parallelism that allows it to be sped up through stream processing on a GPU. We previously presented a method for offloading fuzzy clustering to a GPU, while maintaining full control over the various clustering parameters. In this work we extend that research and show how to incorporate non-Euclidean distance metrics. Our results show a speed increase of one to almost two orders of magnitude for particular cluster configurations. This methodology is particularly important for real time applications such as segmentation of video streams and high throughput problems.

1 Introduction

Many pattern recognition algorithms, such as clustering, can be sped up on platforms that utilize multiple processing cores, operate efficiently for large amounts of floating point operations (FLOPS), and natively support numerically expensive linear algebra instructions. Graphics Processing Units (GPUs) are relatively new general-purpose stream processing hardware that are well suited for these types of problems. When presented with a collection of input data, such as pixels in an image, stream processing computes a function on the entire collection, much like a kernel operation, where each stream element is assumed to be independent of all other elements. This framework naturally lends itself to image processing, but there has recently been a trend of converting many pattern recognition algorithms to GPU programs for computational speedup. However, the conversion process is typically not trivial. In order to be transferred to a GPU, most algorithms must be reformulated. In addition, GPU hardware is changing at a fast rate and programming languages are not as mature as desired. This all equates to a slight GPU programming learning curve.

Improving the computational performance of clustering is not a new concept. Shankar and Pal presented a progressive subsampling method called fast fuzzy

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 128–139, 2007.
springerlink.com

c-means (FFCM) [1]. FFCM generates a sequence of extended partitions of the entire data set by applying the original FCM to a nested sequence of increasing size subsamples. FFCM terminates when the difference between successive extended partitions is below a single threshold value. Speed is always a concern, but so is the size of the data set. In [2] Pal and Bezdek developed the extensible fast fuzzy c-means clustering (eFFCM) algorithm for the segmentation of very large digital images. In [3] Hathaway and Bezdek discuss an extension of the eFFCM method, geFFCM, to nonimage object data. It should be made clear that we are presenting a procedure to transfer fuzzy clustering to specialized hardware, which is different from many previous attempts that look to find algorithmic or mathematical reformulations. As improvements are made to the clustering algorithms, these concepts can be converted to a GPU implementation. Our research goal is to present a different computational platform for clustering.

Harris and Hanes conducted the first known work in the area of offloading clustering to a GPU in [4]. In their work, a speedup of 1.7 to 2.1 times over a CPU is reported for a NVIDIA GeForceFX 5900 Ultra. Their method is designed to handle three linearly separable clusters with a dimensionality of three. The benefit is that they can handle a large number of data points. A problem is that the proposed formulation is not scalable. As the method is presented, it is not capable of extension with respect to either the feature dimensionality size or number of cluster centers. They acknowledge this and state that it would be a large undertaking. They instead proposed to look into increasing efficiency before dealing with the problem of control over the various fuzzy clustering parameters.

In [5] we presented a generalized method for offloading fuzzy clustering, in particular, the Fuzzy C-Means (FCM) to a GPU, allowing for full control over the various clustering parameters. A computational speedup of over two orders of magnitude was observed for particular clustering configurations, i.e. variations of the number of data points, feature dimensionality, and the number of cluster centers. The metric used was the Euclidean distance. Here, we extend that research and show how to incorporate non-Euclidean distance metrics. This changes the parameter representational scheme some, but the algorithmic formulation has the biggest modifications.

The focus of this paper is the FCM, but many high throughput problems, like protein sequence structure search, are excellent candidates for GPU enhancement. We are currently using GPUs to speed up image processing. In particular, we employ GPUs for human silhouette segmentation for eldercare activity analysis and fall detection. Silhouette segmentation requires image pre-processing, feature extraction (color and texture) in various color spaces, data fusion, shadow detection and removal, and post-processing (such as morphology). Image processing is significantly sped up as long as the problem is formulated as one such that the image data remains on the GPU, i.e. avoid CPU to GPU memory transfers. In the following sections we (1) present an overview of the FCM, (2) provide an introduction to GPUs, (3) discuss the generalized algorithm for computing fuzzy clustering on a GPU with a non-Euclidean distance metric, (4) present the experiments, (5) display the results, and (6) discuss extensions to this work.

2 Clustering

Clustering is an unsupervised learning procedure that can be used to reveal patterns in a collection of data, denoted by $X = \{\vec{x}_1, \ldots \vec{x}_N\}$. Each sample vector contains K features, represented as $\vec{x}_i = (f_{i1}, \ldots f_{iK})^T$. Each cluster can be represented by a set of parameters, θ_j $(1 \le j \le C)$. In the simplest case, θ_j is a K-dimensional vector representing the jth cluster center. In the standard approach [6], the clustering algorithm alternately estimates the collection of cluster centers, $\theta = \{\theta_1, \ldots \theta_C\}$, and a membership matrix, U, where the membership of the ith sample in the jth cluster is denoted by $u_{(i,j)}$. In the Hard C-Means (HCM) clustering algorithm, cluster membership values are crisp, i.e. $u_{(i,j)} \in \{0,1\}$. Fuzzy clustering allows $u_{(i,j)} \in [0,1]$, i.e. each element can be shared by more than one cluster. Fuzzy clustering follows the principle of least commitment, which is the belief that one should never over commit or do something which might have to be later undone [7]. The Fuzzy C-Means (FCM) cost function for a C cluster problem, originally proposed by Bezdek [6], is defined as

$$J_{FCM}(\theta, U) = \sum_{j=1}^{C} \sum_{i=1}^{N} u_{(i,j)}^{q} d\left(\vec{x}_i, \theta_j\right)$$

$$\text{where } \sum_{j=1}^{C} u_{(i,j)} = 1 \text{ for } i = 1, \ldots N$$

$$\vec{x}_i \in X \text{ and } |X| = N .$$

In the J_{FCM} equation, $d(\vec{x}_i, \theta_j)$ is a distance metric and $q > 1$ is a parameter called the fuzzifier, typically $q = 2$. The update equations, found as necessary conditions to minimize J_{FCM}, for the membership values, $u_{(i,j)}^{FCM}$, cluster centers and covariances, $\theta_j = (\vec{\mu}_j, \Sigma_j)$, are

$$u_{(i,j)}^{FCM}(t) = \frac{1}{\sum_{k=1}^{C} \left(\frac{d(\vec{x}_i, \theta_j(t))}{d(\vec{x}_i, \theta_k(t))} \right)^{\frac{1}{q-1}}} \tag{1}$$

$$\vec{\mu}_j(t+1) = \frac{\sum_{i=1}^{N} \left(u_{(i,j)}^{q}(t) \vec{x}_i \right)}{\sum_{i=1}^{N} u_{(i,j)}^{q}(t)} \tag{2}$$

$$\Sigma_j(t+1) = \frac{\sum_{i=1}^{N}\left(u_{(i,j)}^{q}(t)(\vec{x}_i - \vec{\mu}_j(t+1))(\vec{x}_i - \vec{\mu}_j(t+1))^T\right)}{\sum_{i=1}^{N} u_{(i,j)}^{q}(t)} \tag{3}$$

The update equations depend on the current iteration, shown as t above. The membership values and the data points are used at each iteration to compute the next cluster centers (equation 2). The new cluster centers and the memberships are then used to update the covariances (equation 3). When we initially presented the FCM on a GPU, we used the Euclidean distance metric. In this paper we use both the Mahalanobis distance, $d_M(\vec{x}_i, \theta_j)$, and the Gustafson-Kessel (GK) distance, $d_{GK}(\vec{x}_i, \theta_j)$. At the current time, we restrict the covariance matrices to be diagonal to simplify the storage and computation.

$$d_M(\vec{x}_i, \theta_j) = \left(\left(\vec{x}_i - \vec{\mu}_j\right)^T \Sigma_j^{-1}\left(\vec{x}_i - \vec{\mu}_j\right)\right)^{1/2} \tag{4}$$

$$d_{GK}(\vec{x}_i, \theta_j) = \left(\left|\Sigma_j\right|^{1/K}\left(\left(\vec{x}_i - \vec{\mu}_j\right)^T \Sigma_j^{-1}\left(\vec{x}_i - \vec{\mu}_j\right)\right)\right)^{1/2} \tag{5}$$

3 Graphics Processing Units

Traditionally, graphics operations, such as mathematical transformations between coordinate spaces, rasterization, and shading operations have been performed on the CPU. GPUs were invented in order to offload these specialized procedures to advanced hardware better suited for the task at hand. Because of the popularity of gaming, movies, and computer-aided design, these devices are advancing at an impressive rate, given the market demand. The key to programming for a GPU is in the design of the CPU driver application, understanding of the graphics pipeline, and ability to program for GPU processors. Figure 1 shows the process of rending a triangle in the graphics pipeline.

The traditional rasterization and shading process, shown in Figure 1, begins with the specification of vertices. The vertices form a primitive that undergoes rasterization. Rasterization results in a set of fragments, which are then subject to shading. Currently there are three programmable components on a GPU. The first programmable component is the vertex processor. Vertex processors are traditionally responsible for mathematical transformations between coordinate spaces and texture coordinate generation. The next programmable unit, introduced in DirectX 10 and the Shader Model 4, is the geometry processor, which takes the transformed vertices and allows for per-primitive operations. The last programmable unit, the fragment processor, also sometimes called the pixel shader, is traditionally responsible for sampling and applying textures, lighting equations, and other advanced graphics and image processing effects. There are traditionally more fragment processors, because a few vertices are responsible for a larger number of fragments. A good GPU introduction can be found in [8, 9].

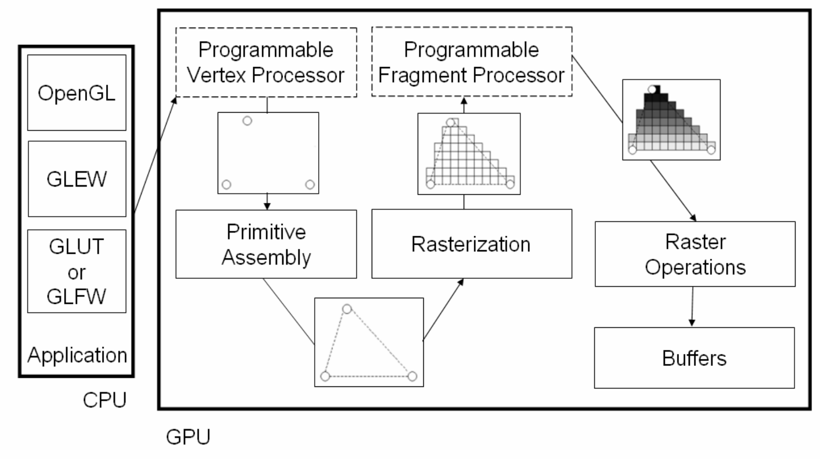


Fig. 1. Graphics pipeline stages for rendering a triangle. Programmable GPU components shown as dashed boxes.

GPUs are increasing at a faster rate, in terms of computational power, than CPUs. The annual growth in CPU processing power, as empirically noted by Moore's law, is approximately 1.4, while vertex processors are growing at a rate of 2.3 and pixel processors are growing at a rate of 1.7 [10]. GPUs also support native operations such as matrix multiplication, dot products, and computing the determinant of a matrix. GPUs are also capable of executing more floating point operations per second. A 3.0 GHz dual-core Pentium4 can execute 24.6 GFLOPS, while an NVIDIA GeForceFX 7800 can execute 165 GFLOPS per second [10]. The new NVIDIA GeForce 8800 GTX has 128 stream processors, a core clock of 575 MHz, shader clock of 1350 MHz, and is capable of over 500 GFLOPS [11].

4 Fuzzy Clustering on a GPU

The quintessential concept required to perform general-purpose computing on a GPU is that arrays are equivalent to textures. This means that pattern recognition algorithms must be structurally converted into a suitable texture format. Textures are combinations of red, green, blue, and alpha (RGBA) values. In [5] we presented the initial fuzzy clustering parameter texture packing scheme, where the metric was the Euclidean distance. We also provided a detailed section that describes implementation of GPU algorithms discussed below. Figure 2 shows our format for fuzzy clustering on a GPU with the Mahalanobis and GK distance metrics, which requires that the cluster covariances are stored, in addition to the cluster centers.

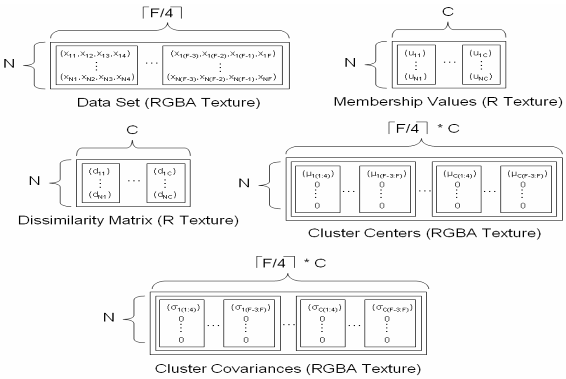

⌈F/4⌉
N
(x11,x12,x13,x14) … (x1(F-3),x1(F-2),x1(F-1),x1F)
(xN1,xN2,xN3,xN4) … (xN(F-3),xN(F-2),xN(F-1),xNF)
Data Set (RGBA Texture)
C
N
(u11) … (u1C)
(uN1) … (uNC)
Membership Values (R Texture)
C
N
(d11) … (d1C)
(dN1) … (dNC)
Dissimilarity Matrix (R Texture)
⌈F/4⌉ * C
N
(μ1(1:4)) … (μ1(F-3:F)) … (μC(1:4)) … (μC(F-3:F))
0
0
Cluster Centers (RGBA Texture)
⌈F/4⌉ * C
N
(σ1(1:4)) … (σ1(F-3:F)) … (σC(1:4)) … (σC(F-3:F))
0
0
Cluster Covariances (RGBA Texture)

Fig. 2. Fuzzy clustering parameters packed into texture memory on a GPU. Textures above are a combination of red, green, blue, and alpha (RGBA) data. In this proposed format, used to simplify the representation and resulting computation, only the diagonal of the covariance matrices are stored.

The basic idea is that the data set is packed into a single RGBA texture that has N rows, one for each data point, and $\lceil F/4 \rceil$ columns, where $\lceil * \rceil$ represents the "ceiling" of *, i.e. the smallest integer greater than or equal to *. F is the feature dimensionality, which is divided by four because four values can be packed into each pixel (RGBA). Presently, most GPUs are limited to textures of size 4096x4096, but newer GPU models, such as the NVIDIA 8800, are starting to support textures of size 8192x8192. This means that up to 8,192 data points of up to 32,768 dimensionality can be packed into a single texture. In [5] we also presented a method to bypass this texture row size limitation, allowing for much larger data sets through the packing of elements into columns sets. The resulting data set texture will therefore have $\lceil F/4 \rceil \times S$ columns, where S is the number of column sets. Using a NVIDIA 8800 we can support profiles such as 4,194,304 data points of dimensionality 4 with 4 cluster centers or 131,072 data points of dimensionality 32 with 16 cluster centers. This column-set packing scheme is shown in Figure 3.

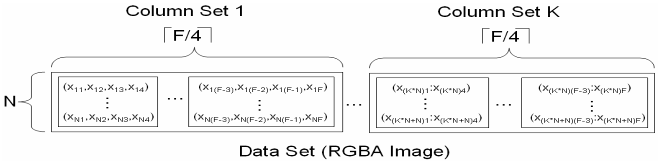

Column Set 1
⌈F/4⌉
Column Set K
⌈F/4⌉
N
(x11,x12,x13,x14) … (x1(F-3),x1(F-2),x1(F-1),x1F)
(xN1,xN2,xN3,xN4) … (xN(F-3),xN(F-2),xN(F-1),xNF)
(x(K*N)1:x(K*N)4) … (x(K*N)(F-3):x(K*N)F)
(x(K*N+N)1:x(K*N+N)4) … (x(K*N+N)(F-3):x(K*N+N)F)
Data Set (RGBA Image)

Fig. 3. Packing of samples into multiple columns in the data set texture in order to support a greater number of samples

The membership and dissimilarity textures are N rows by C columns, where C represents the number of clusters. If multiple column sets are being used, then these matrices have $C \times S$ columns. Each element in these matrices is a scalar, which is packed into a single color channel (i.e. red color channel). The cluster centers are packed into a column format, unlike the row packing scheme for the data set. The number of columns is $\lceil F/4 \rceil \times C$, or $\lceil F/4 \rceil \times C \times S$ for the column set packing method. The number of rows is equal to the number of input vectors. This will be described shortly when we introduce GPU reduction. As mentioned above, the Mahalanobis and GK distance metrics require covariance matrices. For simplicity in representation and in computation, we only store the covariance matrix diagonals. This is the only addition to our previous packing scheme [5], but entails important changes in the ordering and number of GPU programs, shown below in Figure 4. The new FCM GPU algorithm for the Mahalanobis and GK distance metrics is shown in Figure 4.

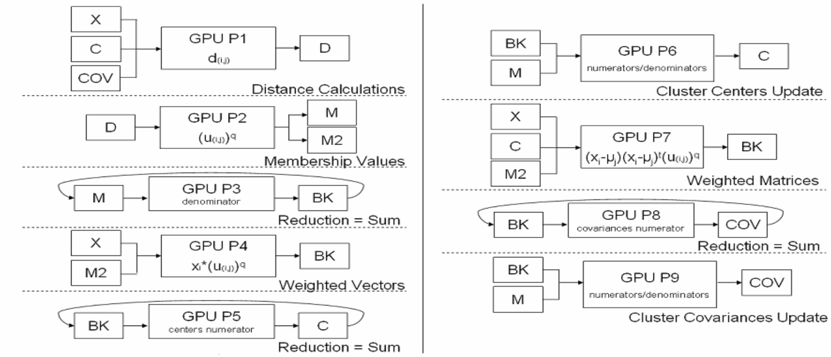


Fig. 4. Generalized GPU algorithm for fuzzy clustering with the Mahalanobis or GK distance metrics. X is the data set texture, C is the cluster center texture, M stores the membership values, COV contains the diagonals of the covariance matrices, D is the dissimilarity matrix, and M2 and BK are temporary buffers. GPU P1, program 1, calculates the dissimilarity, GPU P2 is the membership update step, GPU P3 performs a reduction on the membership matrix, GPU P4 computes the numerator terms in the cluster center update equation, GPU P5 performs a reduction to sum up the numerator terms, GPU P6 computes the new cluster centers, GPU P7 computes the numerator terms in the covariance update step, GPU P8 performs a reduction on the covariance numerator terms, and GPU P9 computes the new covariance matrices.

GPU P1, program 1, shown in Figure 4, computes the dissimilarity values. By only using diagonals of covariance matrices, the metrics are simplified and can be computed fast on the GPU, given its vector based design. The Mahalanobis distance requires (a) vector subtraction, (b) matrix inversion, which for a diagonal matrix is $\left(1/\sigma_{(r,r)}\right)$, for r=1 to K, where $\sigma_{(r,r)}$ is the rth covariance matrix diagonal term, (c) per-component vector multiplication, where the $\left(\bar{x}_i - \bar{\mu}_j\right)$ result is multiplied by the diagonal inverse terms, (d) inner product, and (e) the square root, which is a function supported by the GPU. Most of these operations can have four vector components computed in parallel, given a GPU's support for vectors of size four. The GK metric requires computing the determinant of $\sum_j$, which for a diagonal matrix is the product of the diagonals. The determinant value is raised to $(1/K)$ by the POW function on a GPU.

The following GPU passes use the memberships raised to the qth power. Thus, we compute the membership values and raise them to the qth power to avoid re-computing these values. GPU P2 computes these terms and stores the results in two textures. The reason for two textures is so that one can be used for reduction on the denominator in equations 2 and 3. A reduction in this context is the repeated application of an operation to a series of elements to produce a single scalar result. The reduction procedure here is the addition of a series of elements in texture memory, where operations are running in parallel given the number of fragments processors on a particular GPU. Reduction takes $\log_2(N)$ total passes, where each iteration, i, processes $N/2^i$ stream elements. The reduction procedure is shown in Figure 5.

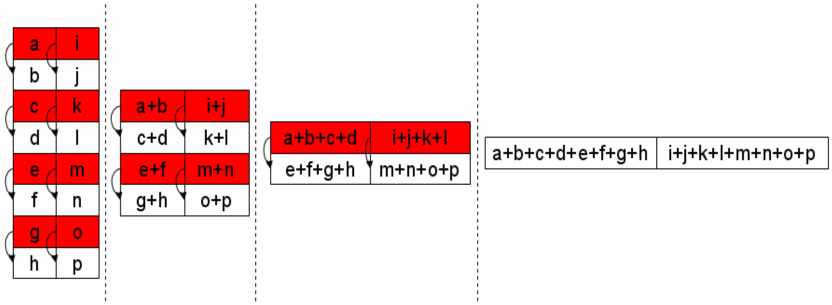


Fig. 5. GPU reduction procedure, summation operation, for a texture with two columns and 8 rows. At each step, $\log_2(N)$ total steps, $N/2^i$, where i is the iteration index, elements in a single column are pair-wised summed. Each column results in the summation of values for a respective column.

GPU P3 performs membership value reduction, i.e. the denominator of equations 2 and 3. GPU P5 and P8 use reduction to compute the sum of the numerator terms in equations 2 and 3 respectively. GPU P6 and P9 divide the reduced (summed) numerators by the reduced (summed) denominators, resulting in the updated cluster centers and covariance matrices.

The cluster centers and covariance matrices textures were shown in Figure 2 to contain zeros in all texture positions except for the first row. This is shown in order to stress the fact that only the first row stores the centers and diagonals of the covariance matrices. GPU P4 and P7 populate the cluster center and covariance matrices with the numerator terms in equations 2 and 3 respectively. The reason for all of the extra rows is to perform the reduction procedure. It is not necessary to initialize the cluster centers or covariance matrices with zeros in all rows after the first. It is also faster to not initialize the matrices past the first row. The only data that needs to be passed down to the GPU, as well as transferred back at the end of the FCM GPU algorithm, is the first row in the cluster centers and covariance matrices textures.

If using the diagonal of the covariance matrices is not acceptable, then the entire covariance matrix can be packed into texture memory and used by the GPU. Figure 6 shows a proposed texture packing scheme.

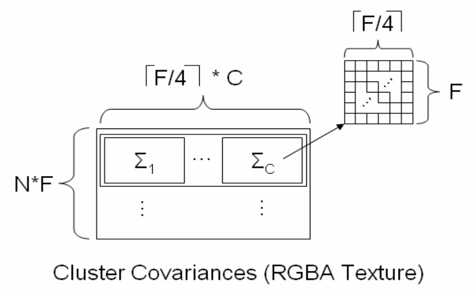


Fig. 6. Full covariance matrix packed into texture memory

If the full covariance matrix is used then less data points can be stored. This results because of the way that we perform reduction. Each element no longer occupies a single row, but rather F rows. The following changes will also need to be made if the full covariance matrix is used. GPU P1 will need to compute the matrix inverse and the matrix determinant. If the feature dimensionality is less than 5, the GPU already has a determinant function. GPU P7 will now have to compute and store the outer product. Because a GPU can only write out to the pixel that it is currently shading, i.e. not neighboring pixels, the covariance matrix will have to be computed in parts. The reduction will also have to be varied to compute a reduction for blocks of memory, i.e. the full covariance matrices. These are the reasons that drove us to use the matrix diagonal, for computational and representational simplicity.

5 Experiments

There is a trend in general-purpose GPU publications that implementation details are extremely vague, almost to the point of being non-reproducible. This provides little benefit to the community, and does not allow others to implement the procedures. We are making the source code, along with documentation, available at http://cirl.missouri.com/gpu/. You can refer to [5] or the web site in order to find out details regarding implementation, such as (1), how to use Cg, NVIDIA's C for Graphics GPU programming language, (2) how to use Frame Buffer Objects (FBO), which are used in order to keep the data on the GPU and make the algorithm go fast, (3) Pixel Buffer Objects (PBO), which can be used to speed up CPU to GPU memory (texture) transfers, (4) GPU numerical precision, such as 16bit and 32bit precision, (5) texture coordinate transformations on the GPU for texture data access, and (6) executing a GPU program, which breaks down to rendering a screen aligned quadrilateral.

There is a tradeoff in terms of time spent setting up the Cg programs, transferring texture memory from CPU to GPU, and managing the graphics pipeline. This means that there are points where it is more or equally efficient to implement the FCM on the CPU rather than on the GPU. GPUs are the most efficient when large batches of data are presented to them. We vary the number of data points, feature dimensionality, number of cluster centers, and distance metric to show performance results for various clustering profiles. The tables below show the speedup of clustering on a GPU, but do not reflect the fact that the CPU is freed up to perform additional computing. This means that one machine can be used for other computation at the same time or can be used for clustering on the CPU and GPU simultaneously, hence increasing the productivity of the entire platform.

Two unique CPU models and three unique GPU models are benchmarked below. The idea is to show differences as it relates to price, computing power, and manufacturer. We used two CPU's: (1) Intel Core 2 Duo T7200 and 2 GB of system RAM and (2) AMD Athlon 64 FX-55 and 2 GB of system RAM. The two GPUs were: (1) NVIDIA Quadro FX 2500M with 512 MB of texture memory, 24 fragment pipelines, and PCI Express X16, and (2) NVIDIA 8800 BFG GTX with 768 MB of texture memory, 128 stream processors, and PCI Express X16.

Our operating system is Windows XP with Service Pack 2 and we are running Visual Studio 2005. The GLEW version is 1.3.4, GLUT version 3.7.6, and Cg version 1.5. Streaming SIMD Extensions 2 (/arch:SSE2), whole program optimization, and maximize speed (/02) were enabled for the CPU in Visual Studio. Because we are not presenting any new metrics or clustering algorithms but rather a speedup method for fuzzy clustering, we use randomly generated clusters. No common data sets from the community were used. We needed many cluster configurations, so we generated elliptical clusters of different sizes with random means. In order to test the GPUs precision, we compared the GPU results to a CPU FCM implementation we wrote, and also to the MATLAB fcm function. In our C implementation we use the same 32bit floating point precision used in our GPU implementation. The C program has the same algorithmic design and final precision as our GPU program. We perform 100 iterations of the FCM on the CPU and the GPU in order to provide a fair comparison.

6 Results

Tables 1 and 2 show CPU over GPU processing time ratios for a clustering task where there are 32 feature dimensions. We used a single 4,096 size row, which is the common max row texture size among the various GPUs that were used for benchmarking.

Table 1. CPU/GPU processing time ratio for 4096 points, 64 clusters, 32 dimensions and the Mahalanobis distance

	GPU1 (2500M)	**GPU2 (8800)**
CPU1 (32bit)	8.51	76.45
CPU2 (64bit)	10.32	83.63

Table 2. CPU/GPU processing time ratio for 4096 points, 64 clusters, 32 dimensions and the GK distance

	GPU1 (2500M)	GPU2 (8800)
CPU1 (32bit)	8.06	74.35
CPU2 (64bit)	9.69	79.74

Tables 1 and 2 show impressive computational speed improvements for a GPU versus the CPU. Depending on the particular GPU and CPU, speed improvements of one to almost two orders of magnitude are observed. The performance numbers are entirely dependent on the GPU generation and CPU that it is compared to. As each new generation of GPU emerges, performance numbers are expected to increase given the popularity of these devices and the need for stream processing. Little computational difference is noticed between the two distance metrics. Table 3 shows the performance behavior when we keep the dimensionality and distance metric fixed, but let the number of clusters and data points vary, comparing the best CPU and GPU.

Table 3. CPU/GPU processing time ratio trend for the 32bit Intel and NVIDIA 8800. The dimensionality is fixed at 4, the number of data points and clusters are varied for the GK distance metric.

		Number of Clusters		
		4	16	64
Data Points	64	0.15	0.91	8.96
	256	0.52	2.67	30.36
	512	1.19	5.16	52.11
	1024	2.26	9.96	63.37
	4096	6.79	42.66	88.21
	8192	12.29	78.25	97.55

The results in Table 3 indicate that when a small number of clusters and data points are pushed down to the GPU performance gain is minimal. In some cases the CPU is even faster. However, as the number of data points increase, the GPU becomes faster. The largest speed improvements are noticed as the number of clusters is increased.

The last performance result is for a larger number of samples. We ran the program on 409,600 samples for both the NVIDIA 8800 and the 32bit Intel CPU. The row size was 4,096, there were 100 column sets, the dimensionality was 8, and there were 4 cluster centers. The time to process the data was 0.94 seconds.

7 Conclusions and Future Work

In this paper, we extended our previous research related to transferring fuzzy clustering to a GPU, specifically as it relates to using different distance metrics. The texture representation change is minimal, while the order and number of GPU program passes is noticeably different. While we specifically described the FCM algorithm, the general outline can be adapted to many other heavy computational techniques. This opens up the potential to perform clustering (and other) algorithms on large data sets at low cost and in a somewhat real time environment without the need for clusters of computers or other expensive dedicated hardware, for example, segmenting a continuous video stream or for bioinformatics applications (like BLAST searches).

As stated in [5], we plan to continue this line of research and find out how a cluster of low end PCs equipped with GPUs perform for larger clustering tasks. Another area of future extension surrounds very large data sets. We intend to take the eFFCM work described by Hathaway and Bezdek and implement it on a single GPU, cluster of PCs equipped with GPUs, or multiple 8800 GPUs on a single machine connected through SLI.

Acknowledgements

D. Anderson and R. Luke are pre-doctoral biomedical informatics research fellows funded by the National Library of Medicine (T15 LM07089).

References

1. B. Uma Shankar and N. R. Pal, "FFCM: An effective approach for large data sets," *Proc. 3rd Int. Conf. Fuzzy Logic, Neural nets, and Soft Computing*, IIZUKA, Fukuoka, Japan, 1994, 332-332
2. N. R. Pal and J. C. Bezdek, "Complexity reduction for "large image" processing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 32, 2002, 598-611.
3. R. J. Hathaway and J. C. Bezdek, "Extending fuzzy and probabilistic clustering to very large data sets," *Computational Statistics & Data Analysis*, 51 (2006), 215-234.
4. C. Harris and K. Haines, "Iterative Solutions using Programmable Graphics Processing Units," *14th IEEE International Conference on Fuzzy Systems 2005*. May, 2005, 12-18.
5. D. Anderson, R. Luke, and J. M. Keller, "Speedup of Fuzzy Clustering Through Stream Processing on Graphics Processing Units", *IEEE Transactions on Fuzzy Systems*, 2006, in review.
6. J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York, NY: Plenum, 1981.
7. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco, CA: W. H. Freeman, 1982.
8. M. Pharr and R. Fernando, GPU Gems 2, Addison-Wesley, 2005.
9. GPGPU, Nov. 18 2006, <http://www.gpgpu.org/>.
10. J. D. Owens, et al., "A Survey of General-Purpose Computation on Graphics Hardware," *Eurographics 2005, State of the Art Reports*, August, 2005.
11. Nvidia Corp., "GeForce 8800," Nov. 18 2006, <http:// www.nvidia.com/ page/ geforce_8800.html>.

Fuzzy C-Means, Gustafson-Kessel FCM, and Kernel-Based FCM: A Comparative Study

Daniel Graves and Witold Pedrycz

Department of Electrical and Computer Engineering, University of Alberta, 2nd Floor ECERF, 9107 - 116 Street, Edmonton, Alberta, Canada T6G 2V4
dgraves@ualberta.ca, pedrycz@ece.ualberta.ca

Abstract. This paper is concerned with a comparative study of the performance of fuzzy clustering algorithms Fuzzy C-Means (FCM), Gustafson-Kessel FCM (GK-FCM) and two variations of kernel-based FCM. One kernel-based FCM (KFCM) retains prototypes in the input space while the other (MKFCM) implicitly retains prototypes in the feature space. The two performance criteria used in the evaluation of the clustering algorithm deal with produced classification rate and reconstruction error. We experimentally demonstrate that the kernel-based FCM algorithms do not produce significant improvement over standard FCM for most data sets under investigation It is shown that the kernel-based FCM algorithms appear to be highly sensitive to the selection of the values of the kernel parameters.

Keywords: Fuzzy Clustering, Kernels, Fuzzy Kernel-based Clustering, FCM.

1 Introduction

In determining the structure in data, fuzzy clustering offers an important insight into data by producing gradual degrees of membership to individual patterns within clusters. A significant number of fuzzy clustering algorithms have been developed with widely known methods such as FCM [1] and GK-FCM [5,6]. Recently kernel-based fuzzy clustering has been developed and already gained some popularity, cf. [2,3,12,13,14,15,16]. Kernel-based clustering algorithms map features from the feature space to a higher dimensional kernel space and perform clustering in the kernel space. There are two major variations of kernel-based fuzzy clustering. One forms prototypes in the feature space (input space) while the other forms prototypes in the higher dimensional kernel space and completes an inverse mapping of prototypes from kernel space back to the feature space. They are given the acronyms KFCM and MKFCM, respectively. The research interest is to determine the performance of kernel-based clustering vis-à-vis some well known standards such as FCM and GK-FCM. Kernel-based fuzzy clustering has a larger computational demand than FCM thus it becomes of interest to assess to which extent they bring tangible benefits in terms of the quality of the produced results. The key objectives of the study are outlined as follows

- determine how well the two kernel-based FCM algorithms perform on a suite of synthetic and real data when compared with standard FCM and GK-FCM
- determine how sensitive the selection of the kernel parameters is to the quality of the clustering results

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 140–149, 2007.
springerlink.com

To offer an unbiased evaluation of the results of clustering, we introduce two criteria. The first one is a classification rate which emphasizes the role of clusters in the formation of homogeneous groups of patterns (data) belonging to the same class. The second one deals with a reconstruction error and could be sought as an internal measure of quality of the clusters. The paper is organized as follows: background presents both GK-FCM and kernel-based FCM (Section 2), evaluation criteria (Section 3), and conclusions (Section 4). Throughout the study we use the standard notation as encountered in fuzzy clustering: a finite collection of N patterns is described as $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and collection of c cluster centers (prototypes) is denoted $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$ with $\mathbf{x}, \mathbf{v} \in \Re^d$. The fuzzy partition matrix is $\mathbf{U} = [u_{ik}]$ where $u_{ik} \in [0,1]$ and the fuzzification coefficient of the FCM is denoted as $m > 1$.

2 Background

The FCM [1] minimizes a well-known dispersion performance index. The results are provided in the form of a fuzzy partition matrix as well as a collection of "c" prototypes. Given the Euclidean distance function, FCM favors spherical shapes of the clusters. Many variations of the FCM algorithm have been developed including the Gustafson-Kessel FCM, fuzzy c-means ellipsoid version, and recently kernel-based FCM. These generalizations are aimed at the development of clusters when dealing with a non-spherical geometry of data.

Gustafson-Kessel FCM

The Gustafson-Kessel FCM [5,6] extends the standard FCM algorithm by introducing an augmented version of the Euclidean distance to be in the form $\mathrm{d}^2_{GK} = (\mathbf{x}_k - \mathbf{v}_i)^{\mathrm{T}} \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i)$ where $\mathbf{A}_i$ is calculated using a scaled inverse fuzzy covariance matrix from each cluster. The Gustafson-Kessel FCM minimizes the following performance criterion:

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \, \mathrm{d}^2_{GK} = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)^{\mathrm{T}} \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i) \tag{1}$$

The optimization of the performance criterion is subject to the standard constraints:

$$\forall i,k \;\; u_{ik} \in [0,1], \; \forall i \;\; 0 < \sum_{k=1}^{N} u_{ik} < N, \text{ and } \forall k \;\; \sum_{i=1}^{c} u_{ik} = 1. \tag{2}$$

The matrix $\mathbf{A}_i$ is calculated based on the inverse of fuzzy covariance matrix $\mathbf{C}_i$

$$\mathbf{A}_i = (\rho_i |\mathbf{C}_i|)^{1/d} \mathbf{C}_i^{-1}, \; \mathbf{C}_i = \frac{\sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^{\mathrm{T}}}{\sum_{k=1}^{N} u_{ik}^m} \tag{3}$$

Obviously, we assume that $\mathrm{C_i}$ is invertible otherwise the algorithm will not produce any solution.

Kernel-Based FCM

The kernel method involves performing an arbitrary non-linear mapping Φ from the feature space (input space) to a higher dimensional and possibly infinitely dimensional kernel space [3,4,8,11]. The rationale for going to higher dimensions is that it is possible to apply a linear classifier in the kernel space that results in non-linear classification in the input space [8,11]. Dot products in the kernel feature space can be expressed by a Mercer kernel [3,4,8,11]; thus, dot products in the feature space are not explicitly computed but rather replaced by a Mercer kernel function. A mercer kernel is defined as any positive semi-definite symmetric function [4]. Common mercer kernel functions include Gaussian $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$ where $\sigma^2 > 0$ and polynomial $(\mathbf{x}^{\mathrm{T}}\mathbf{y} + \theta)^p$ where $\theta \geq 0$ and $p > 0$ [4].

There are two major classes in kernel-based fuzzy clustering in the literature which will be distinguished by calling the first KFCM[12,14] and the second MKFCM [2,13,15,16] following the acronym given in [16]. KFCM is a kernel-based FCM clustering method where the prototypes are determined in the original input space and are implicitly mapped to the kernel feature space through a kernel function. The MKFCM is similar to KFCM except the prototypes are implicitly left in the kernel feature space and thus the inverse of the mapping Φ must be approximated in order to find the prototypes in the original input space. Authors in [16] provide a method of approximating the prototypes in the original feature space.

KFCM: Prototypes in Input Space

The advantage of the KFCM clustering algorithm is that the prototypes are retained in the original input space and thus could be interpreted. KFCM minimizes the following performance criterion [12,14]:

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right\|^2 \tag{4}$$

The distance is computed in the feature space by using a Mercer kernel such that

$$\left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right\|^2 = \mathrm{K}(\mathbf{x}_k, \mathbf{x}_k) + \mathrm{K}(\mathbf{v}_i, \mathbf{v}_i) - 2\,\mathrm{K}(\mathbf{x}_k, \mathbf{v}_i) \tag{5}$$

For the Gaussian kernel function $\mathrm{K}(\mathbf{x}, \mathbf{x}) = 1$. By applying Lagrange multipliers to optimize Q with respect to u_{ik} and $\mathbf{v}_i$ for the Gaussian kernel, one obtains [12,14]:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\dfrac{1-\mathrm{K}(\mathbf{x}_k, \mathbf{v}_i)}{1-\mathrm{K}(\mathbf{x}_k, \mathbf{v}_j)}\right)^{\frac{1}{m-1}}}, \quad \mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m}\,\mathrm{K}(\mathbf{x}_k, \mathbf{v}_i)\mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}\,\mathrm{K}(\mathbf{x}_k, \mathbf{v}_i)} \tag{6}$$

Other kernels can be used however the prototype equation must be re-derived.

MKFCM Prototypes in Kernel Feature Space

The advantage of leaving the prototypes in the kernel space and finding the approximate prototypes in the feature space after clustering is that any kernel function can be used in clustering and there is possibly some additional flexibility in clustering because the prototypes aren't constrained to the feature space. The disadvantage is that the prototypes need to be approximated in the feature space from the kernel space.
The MKFCM algorithm [16] minimizes Q:

$$Q=\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^{m}\left\|\Phi(\mathbf{x}_k)-\mathbf{v}_i\right\|^2 , \quad \mathbf{v}_i=\frac{\sum_{k=1}^{N}u_{ik}^{m}\Phi(\mathbf{x}_k)}{\sum_{k=1}^{N}u_{ik}^{m}} \tag{7}$$

Assuming the Euclidean distance and substituting $\mathbf{v}_i$ into $\mathrm{d}_{ik}^2=\left\|\Phi(\mathbf{x}_k)-\mathbf{v}_i\right\|^2$ the metric in the kernel space denoted by d_{ik} as found in terms of the kernel function K

$$\mathrm{d}_{ik}^2=\left\|\Phi(\mathbf{x}_k)-\mathbf{v}_i\right\|^2=\mathrm{K}(\mathbf{x}_k,\mathbf{x}_k)-2\frac{\sum_{j=1}^{N}u_{ij}^{m}\,\mathrm{K}(\mathbf{x}_k,\mathbf{x}_j)}{\sum_{j=1}^{N}u_{ij}^{m}}+\frac{\sum_{j=1}^{N}\sum_{l=1}^{N}u_{ij}^{m}u_{il}^{m}\,\mathrm{K}(\mathbf{x}_j,\mathbf{x}_l)}{\left(\sum_{j=1}^{N}u_{ij}^{m}\right)^2} \tag{8}$$

The same constraints apply on the membership partition as in standard FCM thus Lagrange multiplies were used to minimize Q with respect to u_{ik}

$$u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\mathrm{d}_{ik}^2}{\mathrm{d}_{jk}^2}\right)^{\frac{1}{m-1}}} \tag{9}$$

The clustering is done without explicitly calculating the prototypes in either the kernel space or the feature space thus the algorithm produces a membership partition matrix only by iteratively updating u_{ik} from some initial random membership partition matrix.

Since the prototypes are retained in the kernel space, there needs to be a mechanism to determine the prototypes in the feature space. The method outlined in [16] minimizes

$$V=\left\|\Phi(\mathbf{v}_i)-\mathbf{v}_i\right\|^2=\frac{\sum_{j=1}^{N}\sum_{l=1}^{N}u_{ij}^{m}u_{il}^{m}\,\mathrm{K}(\mathbf{x}_j,\mathbf{x}_l)}{\left(\sum_{j=1}^{N}u_{ij}^{m}\right)^2}-2\frac{\sum_{j=1}^{N}u_{ij}^{m}\,\mathrm{K}(\mathbf{x}_j,\mathbf{v}_i)}{\sum_{j=1}^{N}u_{ij}^{m}}+\mathrm{K}(\mathbf{v}_i,\mathbf{v}_i) \tag{10}$$

Solving $\partial V / \partial \mathbf{v}_i = \mathbf{0}$ requires knowledge of the kernel function K . The formulae for Gaussian (left) and polynomial (right) kernels are

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} e^{-\|\mathbf{x}_k - \mathbf{v}_i\|^2/\sigma^2} \mathbf{x}_i}{\sum_{k=1}^{N} u_{ik}^{m} e^{-\|\mathbf{x}_k - \mathbf{v}_i\|^2/\sigma^2}}, \quad \mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} (\mathbf{x}_k^{\mathrm{T}} \mathbf{v}_i + \theta)^{p-1} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m} (\mathbf{x}_k^{\mathrm{T}} \mathbf{v}_i + \theta)^{p-1}} \tag{11}$$

The prototypes are approximated iteratively using a Kernel-dependant formula such as the ones given for Gaussian kernels or polynomial kernels after the evaluation of the membership partition matrix.

3 Experiments

A number of experiments were run for a series of data sets on the standard FCM algorithm, Gustafson-Kessel FCM and variations of the Kernel-based FCM algorithm for a thorough comparison of the performance of the algorithms. Three synthetic data sets were used and a number of data sets from UCI Machine Learning database were used.

The classification rate and reconstruction error are the two performance criteria that are used to evaluate the performance of the clustering algorithms in this paper. The first evaluates performance by using the correct class as in the context of a classifier. The second is a performance measure that measures the cohesiveness of patterns within a cluster and thus does not require knowledge of the correct classes.

A value of $\rho_i = 1$ was used for the GK-FCM algorithm and to ensure a non-zero determinate for the fuzzy covariance matrix, if $|\mathbf{C}_i| < 10^{-10}$ the covariance matrix was replaced by the identity matrix. FCM, Gustafson-Kessel FCM and KFCM algorithms were averaged over 100 runs while MKFCM was averaged over 20 runs due to increased overhead of the kernel method and varying the kernel parameters. The mean and standard deviation of the classification rates and reconstruction errors are reported. Some of the standard deviations are quite small in the results and thus appear as 0 in the tables. This is due to highly consistent results during the runs and the relatively large number of runs that were averaged.

3.1 Evaluation Criteria

The structure of the data discovered by a clustering algorithm can be compared with a known structure of classes to produce the classification rate. This is accomplished by transforming the fuzzy partition matrix into a partition matrix and using the maximum rule for each pattern to select the cluster with the largest membership value. Class labels are assigned to each cluster according to the class that dominates that cluster. The percentage of patterns that belong to a correctly labeled cluster is called the classification rate. The classification rate can be determined by building a contingency matrix [7]. Higher classification rates indicate better clustering results.

The reconstruction error is a fuzzy-based measure of fuzzy clustering algorithms [10]. The reconstruction error is a measure of spread of clusters from the prototypes by finding the sum of square differences between the original patterns and the decoded patterns. The decoded pattern $\tilde{\mathbf{x}}_k$ and reconstruction error r are calculated by the following formulae:

$$r = \sum_{k=1}^{N} \left\| \mathbf{x}_k - \tilde{\mathbf{x}}_k \right\|^2 , \ \tilde{\mathbf{x}}_k = \frac{\sum_{i=1}^{c} u_{ik}^m \mathbf{v}_i}{\sum_{i=1}^{c} u_{ik}^m} \tag{12}$$

Smaller values for the reconstruction error indicate better clustering results as the clusters are denser and the prototypes are further apart.

3.2 Synthetic Data Sets

Three synthetic data sets of two dimensions are used. The fuzzy XOR data set has 1000 patterns while the parallel lines and the ring data sets have 200 patterns. The fuzzy XOR data set is two strip clusters that follow an XOR pattern. The lines data set consists of two parallel lines. The ring data consists of two rings – one inside the other. There are two clusters in each synthetic data set and both clusters have an equal number of patterns.

The results for the clustering of the Fuzzy XOR data set show the Gustafson-Kessel (GK) FCM clustering algorithm is the clear winner in terms of a classification rate of 93.4% when $c = 2$ however as the number of clusters increase to 4, all clustering algorithms perform very similarly. The prototypes produced by GK FCM are at the center thus the reconstruction error is relatively large. Interestingly the polynomial MKFCM algorithm performs fairly well for $c = 2$ with an average classification rate of 70.2% by capturing three sides of the X in one cluster. The reconstruction error for MKFCM kernel-methods contends very closely with standard FCM.

The kernel methods perform significantly better for the ring data set in terms of classification rate requiring fewer clusters than GK and standard FCM. The kernel-based algorithms appear to be providing some non- spherical cluster shapes for this particular example. As figures 1 and 2 indicate, the performance of the kernel-based method is very sensitive to the choice of the numeric values of the kernels. MKFCM performs better than the other kernel method KFCM. For the Polynomial kernel a larger power of p gives better classification rates but poorer reconstruction errors.

As for the parallel line data set, the GK-FCM was able to classify the data set with a classification rate close to 100%. FCM performed fairly poorly on the line data in terms of classification rate however FCM and Polynomial MKFCM had the smallest reconstruction error. The kernel algorithms performed rather poorly except the Gaussian MKFCM algorithm was able to classify around 94% of the data however the results were highly sensitive to the kernel parameters.

Table 1. Fuzzy XOR Data Set Results

c	Clustering	Classification Rate	Reconstruction Error
2	FCM	51.1 ± 0.1% (m=2.5)	0.1047 ± 0.0005 (m=1.4)
	GK-FCM	93.4 ± 0.0% (m=3)	0.1400 ± 0.0000 (m=1.4)
	KFCM (G)	51.9 ± 3.9% (m=1.4, $\sigma^2 = 0.5$)	0.1096 ± 0.0005 (m=1.4, $\sigma^2 = 2$)
	MKFCM (G)	51.2 ± 0.1% (m=2.5, $\sigma^2 = 8$)	0.1047 ± 0.0005 (m=1.2, $\sigma^2 = 20$)
	MKFCM (P)	70.2 ± 7.6% (m=2.5, $\theta = 50$, $p = 16$)	0.1045 ± 0.0005 (m=1.4, $\theta = 15$, $p = 2$)
3	FCM	77.5 ± 8.1% (m=3)	0.0676 ± 0.0003 (m=1.4)
	GK-FCM	93.2 ± 0.4% (m=1.2)	0.0973 ± 0.0000 (m=3)
	KFCM (G)	79.9 ± 9.1% (m=1.4, $\sigma^2 = 0.5$)	0.0699 ± 0.0003 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	78.3 ± 8.2% (m=2.5, $\sigma^2 = 4$)	0.0676 ± 0.0002 (m=1.4, $\sigma^2 = 30$)
	MKFCM (P)	80.0 ± 8.3% (m=2.5, $\theta = 15$, $p = 4$)	0.0676 ± 0.0002 (m=1.4, $\theta = 35$, $p = 2$)
4	FCM	93.8 ± 0.0% (m=1.4)	0.0251 ± 0.0000 (m=2)
	GK-FCM	94.0 ± 0.0% (m=3)	0.0221 ± 0.0000 (m=3)
	KFCM (G)	93.9 ± 0.1% (m=1.4, $\sigma^2 = 2$)	0.0306 ± 0.0002 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	93.8 ± 0.1% (m=1.4, $\sigma^2 = 12$)	0.0252 ± 0.0000 (m=2, $\sigma^2 = 30$)
	MKFCM (P)	93.4 ± 0.1% (m=1.4, $\theta = 50$, $p = 2$)	0.0251 ± 0.0000 (m=2, $\theta = 25$, $p = 2$)

Table 2. Ring Data Set Results

c	Clustering	Classification Rate	Reconstruction Error
2	FCM	51.5 ± 0.6% (m=1.2)	1.3474 ± 0.0043 (m=1.4)
	GK-FCM	52.5 ± 1.9% (m=1.2)	1.3355 ± 0.0035 (m=1.4)
	KFCM (G)	61.4 ± 7.4% (m=1.2, $\sigma^2 = 1$)	1.6868 ± 0.1058 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	100 ± 0.0% (m=1.4, $\sigma^2 = 0.5$)	1.3490 ± 0.0039 (m=1.4, $\sigma^2 = 100$)
	MKFCM (P)	100 ± 0.0% (m=2, $\theta = 2$, $d = 4$)	1.3454 ± 0.0039 (m=1.4, $\theta = 15$, $d = 2$)
3	FCM	62.5 ± 2.7% (m=2)	0.8664 ± 0.0037 (m=1.4)
	GK-FCM	87.4 ± 0.9% (m=2)	1.0670 ± 0.0030 (m=2)
	KFCM (G)	73.5 ± 10.6% (m=1.2 $\sigma^2 = 2$)	1.2520 ± 0.0948 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	100 ± 0.0% (m=1.4, $\sigma^2 = 0.5$)	0.8656 ± 0.0031 (m=1.4, $\sigma^2 = 100$)
	MKFCM (P)	100 ± 0.0% (m=2.5, $\theta = 7$, $d = 8$)	0.8548 ± 0.0040 (m=1.4, $\theta = 25$, $d = 2$)
4	FCM	99.9 ± 0.1% (m=1.7)	0.5866 ± 0.0018 (m=1.7)
	GK-FCM	87.4 ± 7.3% (m=1.2)	0.9453 ± 0.1613 (m=1.7)
	KFCM (G)	89.7 ± 9.8% (m=1.7, $\sigma^2 = 2$)	0.7690 ± 0.1204 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	100 ± 0.0% (m=1.2, $\sigma^2 = 0.5$)	0.5730 ± 0.0025 (m=1.4, $\sigma^2 = 4$)
	MKFCM (P)	100 ± 0.0% (m=2, $\theta = 2$, $d = 8$)	0.5991 ± 0.0013 (m=1.4, $\theta = 50$, $d = 12$)

The average computation time on the Fuzzy XOR data set over 20 runs is given in table 3. The MKFCM algorithms require significantly more computation than FCM. The KFCM requires much less computation than the MFKCM algorithms.

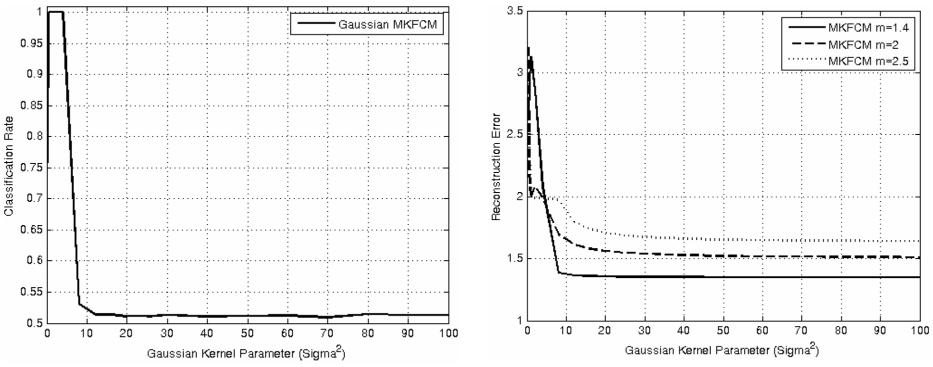


Fig. 1. Classification Rate (left) and Reconstruction Error (right) versus σ^2 for Gaussian MKFCM on Ring (c=2)

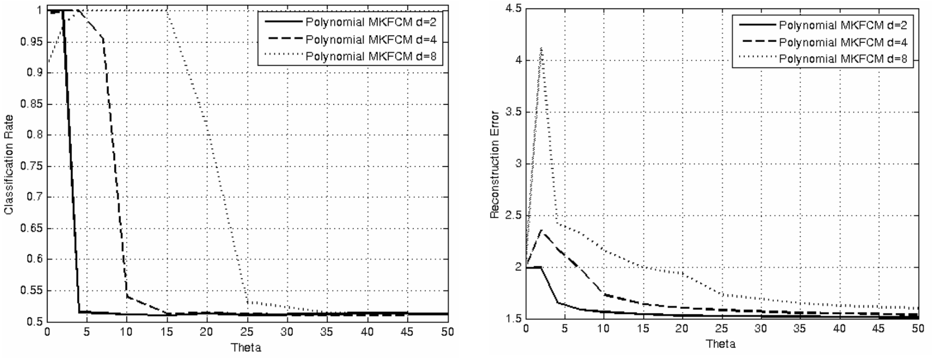


Fig. 2. Classification Rate (left) and Reconstruction Error (right) versus Polynomial Kernel Parameters for Polynomial MKFCM on Ring (c=2)

Table 3. Average Computation Time for Clustering Algorithms

Algorithm	FCM	GK-FCM	KFCM	MKFCM (G)	MKFCM (P)
Time (s)	1.36	3.43	3.21	32.93	31.68

The kernel-based algorithms do not appear to provide a magical solution to the problem of clustering and in particular with non-spherical shaped clusters. Although the kernel-based algorithms tend to perform much better for the ring cluster, there is a trade-off with the sensitivity of the kernel parameters to the clustering results.

3.3 UCI Machine Learning Data Sets

The iris, wine, liver and breast cancer UCI Machine Learning data sets [9] were included in the study. The attributes of the UCI data sets were normalized by subtracting the mean and dividing by the standard deviation. Overall standard FCM had the smallest reconstruction error and the MKFCM algorithm followed closely behind FCM. As for classification rate, the MKFCM algorithm out-performed only slightly better in most cases.

The classification rate for all clustering algorithms was fairly similar for the Iris data with the GK-FCM algorithm achieving the highest classification rate at 95.3%. The MKFCM algorithm fell slightly behind the rest with 85.6% and 88.7% for Gaussian and polynomial kernels respectively. The MKFCM algorithm performs fairly well for wine and breast cancer however there is only marginal improvement in the classification rate and the reconstruction error was similar to FCM. The GK-FCM algorithm performs fairly poorly for the wine data set with an average classification rate at 71.4% which is about 25% lower than the others. All the algorithms perform fairly similarly for the liver data set with classification rates around 60% and the highest classification rate belonging to MKFCM with the Gaussian kernel at 61.3%. The reconstruction error was the lowest for FCM at around 3.1 with MKFCM closely behind while KFCM was at 5.7.

The DELVE ring data set was used as a test data set for the MKFCM algorithm outlined in [16] in which the authors obtained 98.7% classification accuracy. Our implementation of the MKFCM algorithm based on [13,15,16] was run on the same DELVE ring data set achieving 98.6% accuracy using same parameters in [16] (m=2, $c=2, \sigma^2 = 42.25$). A kernel-based weighted FCM method in [12] obtained 96% classification on the Iris data set.

4 Conclusions

The kernel-based clustering algorithms – especially MKFCM – can cluster specific non-spherical clusters such as the ring cluster quite well outperforming FCM and GK-FCM; however overall the performance of the kernel-based methods is not very impressive due to similar or only slight increases in clustering classification rates compared to FCM. From the perspective of the reconstruction error, MKFCM often performed similar to that of FCM and KFCM. A major disadvantage of kernel-based algorithms is their sensitivity to the kernel parameters. In fact in some cases a change in the kernel parameters could reduce the classification rate by one-half and multiply the reconstruction error. Thus kernel-based fuzzy clustering requires tuning in order to achieve optimal performance. Nevertheless in most of the artificial and UCI machine learning test data sets run, the optimal performance of the kernel-based clustering algorithms is not that much of an improvement over FCM and GK-FCM.

Acknowledgments. Support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Research Chair Program (W. Pedrycz) is gratefully acknowledged.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York (1981)
2. Chiang, J.H., Hao, P.Y.: A new kernel-based fuzzy clustering approach: support vector clustering with cell growing, IEEE Transactions of Fuzzy Systems, **11**(4) (2003) 518-527
3. Girolami, M.: Mercer kernel-based clustering in feature space. IEEE Transactions on Neural Networks. **10**(5) (1999) 1000-1017
4. Herbrich, R.: Learning Kernel Classifiers, MIT Press, Cambridge Massachusetts (2002).
5. Klawonn, F., Kruse, R.: Constructing a fuzzy controller from data. Fuzzy Sets and Systems, **85** (1997) 177-193
6. Krishnapuram, R., Kim, J.: A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms, IEEE Transactions on Fuzzy Systems, **7**(4) (1999) 453-461
7. Li, T., Ma, S., Ogihara, M.: Entropy-based criterion in categorical clustering. ACM Int. Conf. Proc. Series; Vol. 69 Proc. 21st int. conf. on Machine Learning, **69** (2004) 68-75
8. Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks, **12**(2) (2001) 181-201
9. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998) http://www.ics.uci.edu/~mlearn/MLRepository.html (Accessed Feb 2007)
10. Pedrycz, W.: Knowledge-Based Clustering, J. Wiley, Hoboken, NJ (2005)
11. Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, **10** (1998) 1299-1319
12. Shen, H., Yang, J., Wang, S., Liu, X.: Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets, Soft Computing, **10**(11) (2006) 1061-1073
13. Zeyu, L., Shiwei, T., Jing, X., Jun, J.: Modified FCM clustering based on kernel mapping, Proc. of Int. Society of Optical Engineering, Vol. 4554 (2001) 241-245
14. Zhang, D.Q., Chen, S.C.: Clustering incomplete data using Kernel-based Fuzzy c-means Algorithm, Neural Processing Letters, **18**(3) (2003) 155-162
15. Zhang, D., Chen, S.: Fuzzy Clustering using Kernel Method, Proc. Int. Conf. Control and Automation (2002) 123-127
16. Zhou, S., Gan, J.: Mercer kernel fuzzy c-means algorithm and prototypes of clusters, Proc. Conf. on Int. Data Engineering and Automated Learning (2004) 613-618

Improved Fuzzy C-Means Segmentation Algorithm for Images with Intensity Inhomogeneity

Qingjie Zhao, Jingjing Song, and Yueyin Wu

School of Computer Science & Technology, Beijing Institute of Technology, Beijing 100081, P.R. China

Abstract. Image segmentation is a classic problem in computer image comprehension and related fields. Up to now, there are not any general and valid partition methods which could satisfy different purposes, especially for medical images such as magnetic resonance images, which often corrupted by multiple imaging artifacts, for example intensity inhomogeneity, noise and partial volume effects. In this paper, we propose an improved fuzzy c-means image segmentation algorithm with more accurate results and faster computation. Considering two voxels with the same intensity belonging to the same tissue, we use q intensity levels instead of n intensity values in the objective function of the fuzzy c-means algorithm, which makes the algorithm clusters much faster since q is much smaller than n. Furthermore, a gain field is incorporate in the objective function to compensate for the inhomogeneity. In addition, we use c-means clustering algorithm to initialize the centroids. This can further accelerate the clustering. The test results show that the proposed algorithm not only gives more accurate results but also makes the computation faster.

Keywords: Image segmentation, fuzzy c-means.

1 Introduction

Medical images, such as those from Magnetic Resonance (MR), give us large and complex data, so it is difficult to find an appropriate mathematic model to describe them. On the other hand, the images are often corrupted by multiple imaging artifacts, such as intensity inhomogeneity, noise and partial volume effects. The spatial intensity non-homogeneity, which may caused chiefly by the radio-frequency coil in magnetic resonance imaging [1][2], is a serious problem in analysis of MR data. The result shows a slow intensity variation of the same tissue in an image. As a result, it is prone to produce errors by using some conventional intensity-based segmentation methods, and it is also difficult to make a correct classification.

Fuzzy c-means [3-5] is a kind of unsupervised clustering algorithms. It is a good algorithm for image segmentation on account of introducing a fuzzy attribute for each image voxel [1]. However, there are still problems when segmenting MR images with the intensity non-homogeneity. Many methods have been proposed to solve this problem [1,2,6-8]. Pham and Prince propose an adaptive fuzzy c-means algorithm [2,7]. In their method, a gain field term is incorporated into the objective function of the standard fuzzy c-means algorithm to simulate the intensity non-homogeneity. On the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 150–159, 2007.
springerlink.com

other hand, the authors added one first order regularization term and one second order term into the objective function to ensure the estimated bias field is smooth and varies slowly. Without these regularization terms, a multiplier field could always be found to set the objective function to zero [2]. Besides, the algorithm is sensitive to noise, and the computation of each iteration step is complex. Ahmed et al propose a bias correction fuzzy c-means algorithm [1]. The authors improved the algorithm by including immediate neighborhood in the standard objective function of fuzzy c-means algorithm, that is, the neighborhood effect acts as a regularization term. Hence the bias correction fuzzy c-means algorithm is insensitive to salt and pepper noise, but the computational load is heavy. He et al [6] take into account noise and intensity nonuniformity in MR images. This algorithm is utilized to segment three-dimensional multi-spectral MR images. For medical image segmentation, the existing algorithms often take much time in computing, and therefore may be inconvenient for clinical applications. Now, many researchers try to find a fast segmentation algorithm for the real-time clinical applications. Enlightening by the standard fuzzy c-means algorithm and Ahmed et al's algorithm, Szilagyi [9] split up the two major steps of the latter, and introduce a new factor, and as a result the amount of required calculations is considerably reduced. However, the authors do not consider the MR imaging with brightness variation caused by the nonuniformity, so it is unsuitable for using this method to classify the imaging corrupted by the intensity inhomogeneities.

In this paper, we present a novel and fast algorithm for fuzzy segmentation of MR imaging data corrupted by the intensity inhomogeneity. The algorithm incorporates a gain field in the objective function of the fast fuzzy c-means algorithm to compensate for the inhomogeneity in the MR images. The c-means clustering algorithm is used to initialize the centroids to further accelerate the algorithm.

2 Fuzzy C-Means Algorithm and an Improved Method

2.1 C-Means Algorithm

The general problem in clustering is to partition a set of vectors into groups having similar values. In traditional clustering, there are c clusters with means (or centroids) m_1, m_2, …, m_c. A least square error criterion is used to measure how close the data are to the clusters. The objective function is

$$J_{CM} = \sum_{k=1}^{c} \sum_{j=1}^{n} \left\| x_j - m_k \right\|^2 \tag{1}$$

J_{CM} is the sum of all square errors for all clusters. x_j is a point in the image feature space, which is an object of data such as intensity value. x_j and m_k may have more than one dimensions. The norm operator $\|\cdot\|$ represents the standard Euclidean distance. This criterion tries to make the degree of similarity high in the same cluster and low between the different clusters.

Firstly, we choose randomly *c* objects which represents initially the centroids of the *c* clusters. For the other objects, the distances to the *c* centroids are computed respectively. Then according to the distances, an object is assigned to the closest cluster. The means of the new cluster are updated and the above steps are repeated, until the criterion is satisfied or the iteration is achieved.

2.2 Fuzzy C-Means Algorithm

From objective function (1), we use a factor d_{jk} to denote the degree of classification. Then the objective function becomes

$$J'_{CM}=\sum_{k=1}^{c}\sum_{j=1}^{n}d_{jk}\left\|x_j-m_k\right\|^2 \tag{2}$$

If the *j*th sample x_j belongs to the *k*th cluster, then $d_{jk}=1$; if not $d_{jk}=0$. Since the degree of classification is somewhat fuzzy, we use a degree of belongingness μ_{jk} to replace the binary factor d_{jk}. And then we get the objective function of the fuzzy c-means (FCM) algorithms:

$$J_{FCM}=\sum_{k=1}^{c}\sum_{j=1}^{n}\mu_{jk}^{p}\left\|x_j-m_k\right\|^2 \tag{3}$$

$$\sum_{k=1}^{c}\mu_{jk}=1, j=1, 2, \ldots, n$$

μ_{jk} is the grade of membership of the *j*th voxel belonging to class *k*, and the parameter *p* is a weighting exponent on each fuzzy membership and determines the amount of “fuzziness” of a classified result, commonly p>1, typically p=2. The objective function J_{FCM} is minimized when high membership values are assigned to the voxels whose intensity values are close to the centroid of a particular class, and low membership values are assigned to ones with intensity values far from the centroid.

2.3 Bias Correction Fuzzy C-Means Algorithm

Base on the fuzzy c-means algorithm, Ahmed et al propose a bias correction fuzzy c-means algorithm (BCFCM) [1]. They model the intensity inhomogeneity coming from the RF winding. Suppose the value of the *j*th voxel equals a real value and a slow bias field,

$$y_j=x_j+\beta_j, \qquad j=1, 2, \ldots, n \tag{4}$$

Where y_j and x_j is respectively the measured value and the real value of voxel *j*. *n* is the voxel number in a MR image.

Then, they improve the algorithm by including an immediate neighborhood in the standard objective function of fuzzy c-means algorithm. The neighborhood effect acts as a regularization term. Hence this algorithm is insensitive to salt and pepper noise. The new objective function is

$$J_B = \sum_{k=1}^{c}\sum_{j=1}^{n}\mu_{jk}^{p}\left\|x_j - m_k\right\|^2 + \frac{\alpha}{n_R}\sum_{k=1}^{c}\sum_{j=1}^{n}\mu_{jk}^{p}\Big(\sum_{x_r \in n_j}\left\|x_r - m_k\right\|^2\Big) \tag{5}$$

where n_j stands for the set of neighbors of x_j and n_R is the cardinality of the set. The neighbor effect term is controlled by the parameterα. The relative importance of the regularizing term is inversely proportional to the signal to noise ratio of MR images.

Substituting Equation (4) into Equation (5), we have

$$J_B = \sum_{k=1}^{c}\sum_{j=1}^{n}\mu_{jk}^{p}\left\|y_j - \beta_j - m_k\right\|^2 + \frac{\alpha}{n_R}\sum_{k=1}^{c}\sum_{j=1}^{n}\mu_{jk}^{p}\Big(\sum_{x_r \in n_j}\left\|y_r - \beta_r - m_k\right\|^2\Big) \tag{6}$$

Taking the derivative of J_B with respect to μ_{jk}, m_k and β_k. and then making it to be zero, we can get the increment expressions of the three parameters.

Fig.1 shows the segmented results for a synthetic test image by using the fuzzy c-means algorithm and the bias correction fuzzy c-means algorithm. The test image is classified into three tissues: gray matter, white matter and background. Fig.1(a) is T1-weighted image with 5% Gaussian noise and 20% intensity non-homogeneity. Fig.1(b) is the result by using the fuzzy c-means algorithm, and Fig.1(c) is the result by using the bias correction fuzzy c-means algorithm. From the results, we can find that the fuzzy c-means algorithm provides an inaccurate segmentation because of the noise and the fuzziness in the image. And the contour of white matter is unclear in Fig.1(b). However, the bias correction fuzzy c-means algorithm gives a very smooth result, but because the neighborhood is considered in this algorithm the segmentation loses some details.

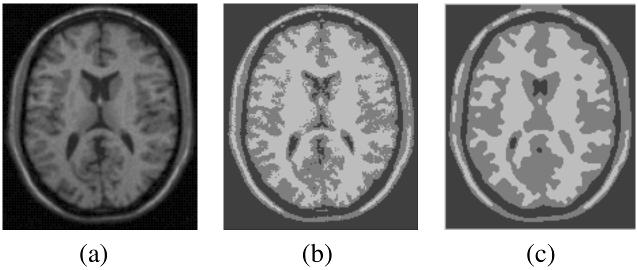


Fig. 1. Segmentation on a test MR image. (a))Original image. (b) FCM segmentation. (c) BCFCM segmentation.

3 Our Algorithm

In the adaptive fuzzy c-means algorithm [2], a gain field term is incorporated into the objective function of the standard fuzzy c-means algorithm to simulate the intensity non-homogeneity, and one first order regularization term and one second order term are added into also. The algorithm is sensitive to noise, and the computation of each iteration step is complex. The bias correction fuzzy c-means algorithm [1] includes an immediate neighborhood in the standard fuzzy c-means algorithm to act as a regularization term. Hence the algorithm is insensitive to the salt and pepper noise, but the computation load is very heavy.

In this section, we give a gain field correction fast fuzzy c-means algorithm. The new algorithm can produce more accurate results and makes the computation much faster.

3.1 Fast Fuzzy C-Means Algorithm

From the traditional fuzzy c-means algorithm, we can find that the computation of each iteration step must be conducted on the whole data set, which needs lots of computations. From the objective function of traditional fuzzy c-means algorithm, we can easily find that if two voxels have the same value of intensity, they will belong to the same class. In an image, the intensity level of a voxel is typically between 0~255, so the set of intensity levels is much less than that of voxels. We can change the fuzzy c-means algorithm by using q intensity levels instead of n intensity values in the objective function.

Suppose there are q intensity levels. We use h_i denotes the number of voxels with value x_i in MR data. It is similar to the histogram of an image, so we can easily get h_i (i=1, 2, …, q). According to the definition, we have

$$\sum_{i=1}^{q} h_i = n$$

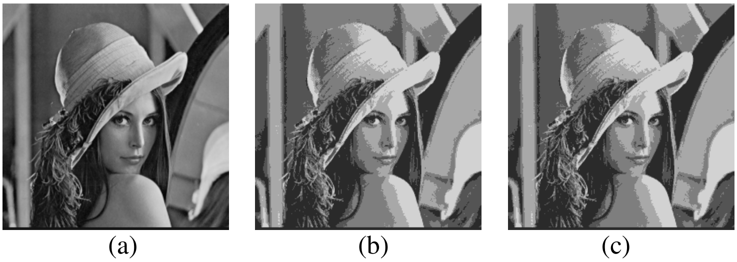
And the objective function used for image segmentation can be written as:

$$J_{FFCM} = \sum_{k=1}^{c}\sum_{i=1}^{q} h_i \mu_{ik}^{p} \left\| x_i - m_k \right\|^2 \tag{7}$$

In this way, the set of data used is reduced. For an image with size 256×256 and intensity levels 0~255, the data number is reduced from 65536 to 256.

Since q is much smaller than n, the fast fuzzy c-means (FFCM) algorithm is on a very smaller data space than the conventional fuzzy c-means algorithm. As a result the former is much faster than the latter.

Fig.2 shows the segmented results for a familiar image by using the fuzzy c-means algorithm and the fast fuzzy c-means algorithm. The image is classified into five sections. Fig.2(a) is the original image. Fig.2(b) is the result by using the fuzzy c-means algorithm, and Fig.2(c) is the result by using the fast fuzzy c-means algorithm. In both cases the clustering results are similar but the fast fuzzy c-means algorithm clusters more rapidly.



(a) (b) (c)

Fig. 2. FCM and FFCM segmentation results (*c*=5). (a) Original image. (b) FCM segmentation. (c) FFCM segmentation.

3.2 Gain Field Correction Fast Fuzzy C-Means Algorithm

Suppose a voxel at position *j* is modeled as a product of the "true" signal intensity multiplied by a slowly varying factor g called gain field [8], namely,

$$y_j = g_j x_j + noise(j), j = 1, 2, \ldots, n \tag{8}$$

where y_j and x_j are the observed intensity values and the true intensity values respectively at voxel *j*. *noise*(*j*) is the independent white Gaussian distributed noise at voxel *j*. n is the total number of voxels in a MR image.

In order to reduce the influence of non-homogeneity caused by RF coils, and to incorporate the gain field into the fast fuzzy c-means mechanism, we combine (7) and (8) to yield:

$$J'_G = \sum_{k=1}^{c} \sum_{i=1}^{q} h_i \mu_{ik}^{p} \left\| y_i - g_i m_k \right\|^2 \tag{9}$$

$$\sum_{i=1}^{q} h_i = n$$

Then fuzzy segmentation is achieved by minimizing J'_G. Adding to the formula with a Lagrange multiplier, we have

$$J_G = \sum_{k=1}^{c} \sum_{i=1}^{q} h_i \mu_{ik}^{p} \left\| y_i - g_i m_k \right\|^2 + \sum_{i=1}^{q} \lambda_i (1 - \sum_{k=1}^{c} \mu_{ik}) \tag{10}$$

Taking the derivative of J_G with respect to μ_{ik}, and making it to be zero, we get:

$$\partial J_G / \partial \mu_{ik} = p h_i \mu_{ik}^{p-1} (y_i - g_i m_k)^2 - \lambda_i = 0$$

$$\mu_{ik} = (\lambda_i / p h_i)^{1/(p-1)} (y_i - g_i m_k)^{-2/(p-1)}$$

Using $\sum_{l=1}^{c} \mu_{il} = 1$, we have

$$(\lambda_i / ph_i)^{1/(p-1)} \sum_{l=1}^{c} (y_i - g_i m_l)^{-2/(p-1)} = 1$$

$$\mu_{ik} = \sum_{l=1}^{c} (\frac{y_i - g_i m_k}{y_i - g_i m_l})^{-2/(p-1)} \tag{11}$$

Taking the derivative of J_G with respect to m_k, and making it to be zero, we obtain:

$$\partial J_G / \partial m_k = -2 \cdot \sum_{i=1}^{q} h_i \mu_{ik}^{p} g_i (y_i - g_i m_k) = 0$$

$$m_k = \sum_{i=1}^{q} h_i \mu_{ik}^{p} g_i y_i \Big/ \sum_{i=1}^{q} h_i \mu_{ik}^{p} g_i^2 \tag{12}$$

Similarly, the gain field can be estimated by taking the derivative of J_G to g_i, and making it to be zero.

$$\partial J_G / \partial g_i = -2 \cdot \sum_{k=1}^{c} h_i \mu_{ik}^{p} m_k (y_i - g_i m_k) = 0$$

$$g_i = \sum_{k=1}^{c} h_i \mu_{ik}^{p} m_k y_i \Big/ \sum_{k=1}^{c} h_i \mu_{ik}^{p} m_k^2 \tag{13}$$

It is seemed that the objective function J_G can be minimized directly by using formula (11), (12) and (13), however, there are two problems we should consider. Firstly, a multiplier field should vary slowly and smoothly. In Pham and Prince's algorithm [2,7], they add the first order regularization term and the second order regularization term into the objective function to ensure the estimated field being smooth and varying slowly. In [1], the authors improve the algorithm by including a term that takes into account immediate neighborhood into the objective function, but since the neighborhood is considered, the segmental results will lose some details.

We use another method to solve this problem. An iterative low-pass filter is used to filter the estimated gain field by using Equation (13). The strategy is based on that the multiplier field is of lower frequency and other parts are of higher frequency. On the other hand, the estimated multiplier field g_i can not be directly filtered by an iterative low-pass filter, because g_i loses the two-dimensional space information. We should transform g_i to a two-dimensional multiplier field image firstly. From the objective function (9), we can notice that if a voxel has the intensity i, the value in the two-dimensional gain field image should be g_i. According to this relation, we can easily get the gain field image, and then the low-pass filter is used on it. Of course, we should transform the gain field image to g_i again for the next iteration.

Proper initial centroids will improve the accuracy and reduce the number of iterations. If the initial centroids are very far from the real initial centroids, the segmentation may fail. Thus selecting good initial centroids is also a very important step. Considering the fast convergence of the c-means algorithm, we can use this algorithm first, and then make the results as the initial centroids of our gain field correction fast fuzzy c-means algorithm.

The steps of our algorithm can be described as the following:

(1) Make the results of the c-means algorithm as the initial centroids, and initialize g_i (i=1,2,…,q) with 1.

(2) Update μ_{ik} using (11).

(3) Update centroids m_k using (12).

(4) Update gain field g_i using (13).

(5) Transform g_i to gain field image.

(6) Filter the gain field image using an iterative low-pass filter.

(7) Transform gain field image to g_i.

(8) Return step (2) and repeat until the error criterion is satisfied.

4 Experiments

In this section, we describe the application of our algorithm on magnetic resonance images, and compare the segmentation results with the fuzzy c-means algorithm and the bias correction fuzzy c-means. The algorithm is implemented in Virtual C++ on a PC with Intel Celeron 2.66GHz processor, 512M RAM and NVIDIA GeForce4 MX 4000 graphics card. We set the parameter fuzzy index p=2, the termination criterion $\varepsilon = 0.01$.

Fig.3(a) is a real MR image, obtained from the Medical Image Processing Group of Institute of Automation, Chinese Academy of Sciences. We classify the MR image into 3 classes: gray matter (GM), white matter (WM) and background. In Fig.3(b), the result of traditional fuzzy c-means shows much fuzzier than that of our algorithm in Fig.3(c). The result from our algorithm comes near to the true tissue classification.

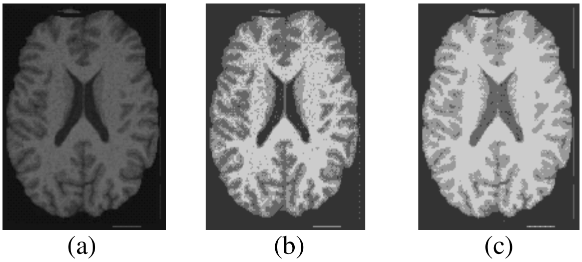


Fig. 3. Segmentation on a real MR image. (a) Original image. (b) FCM segmentation. (c) Our algorithm segmentation.

Fig.4 presents a comparison of segmentation results between the fuzzy c-means, the bias correction fuzzy c-means, and our algorithm. There are many advantages for using digital phantoms rather than real image, where we can include prior knowledge of the true tissue types and control image parameters such as mean intensity values, noise, or intensity inhomogeneity. We also classify the MR image into 3 classes. Fig.4(a) is a T1-weighted MR phantom corrupted with 5% Gaussian noise and 20% intensity non-homogeneity. Fig.4(b), Fig.4(c) and Fig.4(d) shows the segmentation results by using the fuzzy c-means, the bias correction fuzzy c-means, and our segmentation algorithm. From these results, we can see that the traditional fuzzy c-means algorithm provides an inaccurate segmentation because of the intensity non-homogeneity in the image. There are lots of noise in Fig.4(b) and the contour of white matter is unclear. The bias correction fuzzy c-means presents a smooth classification, but the result loses some details for considering the neighborhood. Nevertheless in Fig.4(d), our algorithm provides a better result than other two algorithms.

We should also compare the computational complexities of different algorithms. If the number of iteration is fixed, the execution time of our algorithm is much shorter than that of the bias correction fuzzy c-means. Firstly, from the objective functions of the bias correction fuzzy c-means [1], we will easily find that the bias correction fuzzy c-means algorithm acts on all voxels in a MR image. We know that the number of intensity levels is much less than that of voxels. For 8 bit resolution, there are only 256 intensity levels. Therefore, our algorithm clusters on a very smaller data space than the bias correction fuzzy c-means algorithm does. Secondly, the bias correction fuzzy c-means may misclassify if the initial centroids are not appropriate. Our segmentation algorithm uses the c-means algorithm at first, and makes the results as the initial centroids for further clustering, which can provide good initial centroids and speed up the algorithm. In addition, the bias correction fuzzy c-means uses a regularization term in the objective function to optimize the bias field, which aggravates the computational load.

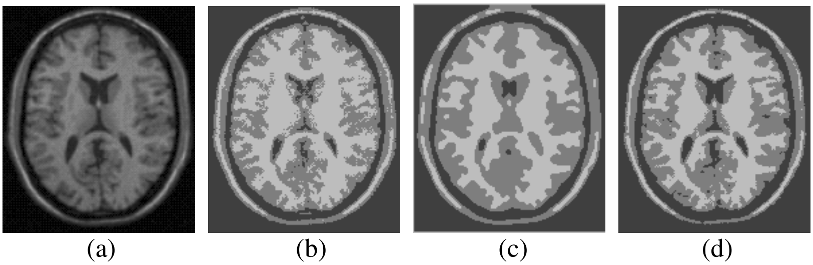


Fig. 4. Segmentation on a noisy MR image. (a) Original image. (b) FCM segmentation. (c) BCFCM segmentation. (d) Our algorithm segmentation.

5 Conclusions

In this paper, we present a new and fast fuzzy segmentation algorithm. Considering two voxels with the same intensity belonging to the same tissue, we use *q* intensity levels instead of *n* intensity values in the objective function of the fuzzy c-means algorithm, which makes the algorithm clusters much faster since *q* is much smaller than *n*. Furthermore, a gain field is incorporate in the objective function to compensate for the inhomogeneity. In each iteration step, we transform the gain field to a gain field image and filter it using an iterative low-pass filter, and then convert the gain field image to a gain field term again for the next iteration. In addition, we use c-means clustering algorithm to initialize the centroids. This can further accelerate the clustering. The test results show that our method reduces a lot of executive time and gives a better segmentation results. The efficiency of the proposed algorithm is demonstrated on different magnetic resonance images.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants No.60503050 and the foundational fund of Beijing Institute of Technology No.200501F4210.

References

1. Mohamed N. Ahmed, Sameh M. Yamany et al: A Modified Fuzzy c-Means Algorithm For Bias Field Estimation and Segmentation of MRI Data. IEEE Transactions on Medical Imaging. 21 (3) (2002) 193–199.
2. Dzung L. Pham, Jerry L. Prince: An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. Pattern Recognition Letters. 20, (1999) 57–68.
3. J. C. Bezdek, L. O. Hall, L. P. Clarke: Review of MR Image Segmentation Techniques Using Pattern Recognition. Med. Phys. 20 (1993) 1033–1048.
4. Weijie Chen, Maryellen L. Giger: A fuzzy c-means (FCM) based algorithm for intenity inhomogeneity correction and segmentation of MR images. IEEE International Symposium on Biomedical Imaging. (2004) 1307–1310.
5. Yonggiang Zhao, Minglu Li: A modified fuzzy c-means algorithm for segmentation of MRI. Fifth International Conference on Computational Intelligence and Multimedia Application. (2003) 391 - 395.
6. Renjie He, Sushmita Datta et al: Adaptive FCM with contextual constrains for segmentation of multi-spectral MRI. Proceeding of the 26th Annual International Conference of the IEEE EMBS. 1 (2004) 1660–1663.
7. Dzung L. Pham, Jerry L. Prince: Adaptive Fuzzy Segmentation of Magnetic Resonance Images. IEEE Transaction on Medical Imaging. 18 (3) (1999) 737–752.
8. Lei Jiang, Wenhui Yang: A Modified Fuzzy C-Means Algorithm for Segmentation of Magnetic Resonance Images. Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.). (2003) 225–231.
9. L. Szilágyi, Z. Benyó, S. M. Szilágyi: Brain Image Segmentation for Virtual Endoscopy. Proceedings of the 26th Annual International Conference of the IEEE EMBS. 1 (2004) 1730 –1732.

Part III

Intelligent Identification and Control

A Fuzzy-Neural Hierarchical Multi-model for Systems Identification and Direct Adaptive Control

Ieroham Baruch, Jose-Luis Olivares G., Carlos-Roman Mariaca-Gaspar,
and Rosalíba Galvan Guerra

CINVESTAV-IPN, Department of Automatic Control, Av. IPN No2508, A.P. 14-740, 07360
Mexico D.F., México
{baruch,lolivares,cmariaca,rgalvan}@ctrl.cinvestav.mx

Abstract. A Recurrent Trainable Neural Network (RTNN) with a two layer canonical architecture and a dynamic Backpropagation learning method are applied for local identification and local control of complex nonlinear plants. The RTNN model is incorporated in Hierarchical Fuzzy-Neural Multi-Model (HFNMM) architecture, combining the fuzzy model flexibility with the learning abilities of the RTNNs. A direct feedback/feedforward HFNMM control scheme using the states issued by the identification FNHMM is proposed. The proposed control scheme is applied for 1-DOF mechanical plant with friction, and the obtained results show that the control using HFNMM outperforms the fuzzy and the single RTNN one.

1 Introduction

In the last decade, the computational intelligence, including artificial Neural Networks (NN) and Fuzzy Systems (FS) became a universal tool for many applications. Because of their approximation and learning capabilities, the NNs have been widely employed to dynamic process modeling, identification, prediction and control, [1]-[4]. Mainly, two types of NN models are used: Feedforward (FFNN) or static and Recurrent (RNN) or dynamic. The first type of NN could be used to resolve dynamic tasks introducing external dynamic feedbacks. The second one possesses its own internal dynamics performed by its internal local feedbacks so to form memory neurons [3], [4]. The application of the FFNN for modeling, identification and control of nonlinear dynamic plants caused some problems due to the lack of universality. The major disadvantage of all this approaches is that the identification NN model applied is a nonparametric one that does not permit them to use the obtained information directly for control systems design objectives. In [5], [6], Baruch and co-authors applied the state-space approach to describe RNN in an universal way, defining a Jordan canonical two- or three-layer RNN model, named Recurrent Trainable Neural Network (RTNN) which has a minimum number of learned parameter weights. This RTNN model is a parametric one, permitting to use of the obtained parameters and states during the learning for control systems design. This model has the advantage to be completely parallel one, so its dynamics depends only on the previous step and not on the other past steps, determined by the systems order which simplifies the computational complexity of the learning algorithm with respect to the sequential RNN model of Frasconi, Gori and Soda (FGS-RNN), [4]. For complex plants identification it is proposed

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 163–172, 2007.
springerlink.com

to use the Takagi-Sugeno (T-S) fuzzy-neural model [7] applying T-S fuzzy rules with a static premise and a dynamic function consequent part, [8]. The [9], [10], [11] proposed as a dynamic function in the consequent part of the T-S rules to use a RNN. The difference between the used in [9] fuzzy neural model and the approach used in [10] is that the first one uses the FGS-RNN model [4], which is sequential one, and the second one uses the RTNN model [5], [6] which is completely parallel one. But it is not still enough because the neural nonlinear dynamic function ought to be learned, and the Backpropagation (BP) learning algorithm is not introduced in the T-S fuzzy rule. So, the present paper proposed to extend the power of the fuzzy rules, using in its consequent part a learning procedure instead of dynamic nonlinear function and to organize the defuzzyfication part as a second RNN hierarchical level incorporated in a new Hierarchical Fuzzy-Neural Multi-Model (HFNMM) architecture. The output of the upper level represents a filtered weighted sum of the outputs of the lower level RTNN models. The HFNMM proposed uses only three membership functions (positive, zero, and negative), which combine the advantages of the RNNs with that of the fuzzy logic, simplifying the structure, augmenting the level of adaptation and decreasing the noise.

2 RTNN Model and Direct Control Scheme Description

The RTNN model is described by the following equations, [5], [6]:

$$X(k+1) = JX(k) + BU(K); J = block - diag(J_{ii}); |J_{ii}| < 1\,. \tag{1}$$

$$Z(k) = \Gamma[X(k)]; Y(k) = \Phi[CZ(k)] \tag{2}$$

Where: *Y, X, U* are, respectively, *l, n, m* - output, state and input vectors; *J* is a (*nxn*)-state block-diagonal weight matrix; J_{ii} is an *i*-th diagonal block of *J* with (*1x1*) dimension; *Γ(.), Φ(.)* are vector-valued activation functions like saturation, sigmoid or hyperbolic tangent, which have compatible dimensions. Equation (1) includes also the local stability conditions, imposed on all blocks of *J*; *B* and *C* are (*nxm*) and (*lxn*)-input and output weight matrices; *k* is a discrete-time variable. The stability of the RTNN model is assured by the activation functions and by the local stability condition (1). The given RTNN model is a completely parallel parametric one, with parameters - the weight matrices *J, B, C,* and the state vector *X*. The RTNN topology has a linear time varying structure properties like: controllability, observability, reachability, and identifiability, which are considered in [6]. The main advantage of this discrete RTNN (which is really a Jordan Canonical RNN model), is of being an universal hybrid neural network model with one or two feedforward layers, and one recurrent hidden layer, where the weight matrix *J* is a block-diagonal one. So, the RTNN posses a minimal number of learning weights and the performance of the RTNN is fully parallel. Another property of the RTNN model is that it is globally nonlinear, but locally linear. Furthermore, the RTNN model is robust, due to the dynamic weight adaptation law, based on the sensitivity model of the RTNN, and the performance index minimization. The general RTNN - BP learning algorithm, is:

$$W_{ij}(k+1) = W_{ij}(k) + \eta\Delta W_{ij}(k) + \alpha\Delta W_{ij}(k-1) \tag{3}$$

Where: W_{ij} is a general weight, denoting each weight matrix element (C_{ij}, A_{ij}, B_{ij}) in the RTNN model, to be updated; ΔW_{ij} (ΔC_{ij}, ΔJ_{ij}, ΔB_{ij}), is the weight correction of W_{ij}; while; η and α are learning rate parameters. The weight updates are as:

$$\Delta C_{ij}(k) = [Y_{d,j}(k)\text{-}Y_j(k)]\Phi_j'[Y_j(k)]Z_i(k)\,, \tag{4}$$

$$\Delta J_{ij}(k) = RX_i(k-1); \Delta B_{ij}(k) = RU_i(k)\,, \tag{5}$$

$$R = C_i(k)[Y_d(k)\text{-}Y(k)]\Gamma_j'[Z_i(k)] \tag{6}$$

Where: ΔJ_{ij}, ΔB_{ij}, ΔC_{ij} are weight corrections of the weights J_{ij}, B_{ij}, C_{ij}, respectively; (Y_d -Y) is an error vector of the output RTNN layer, where Y_d is a desired target vector and Y is a RTNN output vector, both with dimensions l; X_i is an i-th element of the state vector; R is an auxiliary variable; Φ_j', Γ_j' are derivatives of the activation functions. Stability proof of this learning algorithm is given in [6]. The equations (1), (2) together with the equations (3)-(6) forms a BP-learning procedure, where the functional algorithm (1), (2) represented the forward step, executed with constant weights, and the learning algorithm (3)-(6) represented the backward step, executed with constant signal vector variables. This learning procedure is denoted by Π *(L, M, N, Y_d, U, X, J, B, C, E)*. It uses as input data the RTNN model dimensions *l, m, n,* and the learning data vectors Y_d, U, and as output data - the X-state vector, and the matrix weight parameters *J, B, C*.

The block-diagram of the direct adaptive neural control system is given on Fig.1a. The control scheme contains three RTNNs. The RTNN-1 is a plant identifier, learned by the identification error $E_i = Y_d - Y$, which estimates the state vector. The RTNN-2 and RTNN-3 are feedback and feedforward NN controllers, learned by the control error $E_c = R\text{-}\ Y_d$. The control vector is a sum of RTNN functions F_{fb}, F_{ff}, learned by the procedure Π*(L, M, N, Y_d, U, X, J, B, C, E)*, [5], as:

$$U(k) = -U_{fb}(k) + U_{ff}(k) = -F_{fb}[X(k)] + F_{ff}[R(k)] \tag{7}$$

This control system structure is maintained also in the case of the fuzzy-neural system, where the neural identifier is substituted by fuzzy-neural identifier and the neural controller is substituted by fuzzy-neural controller.

3 HFNMM Identifier and HFNMM Controller Description

Let us assume that the unknown system $y = f(x)$ generates the data $y(k)$ and $x(k)$ measured at *k, k-1,..p*, then the aim is to use this data to construct a deterministic function $y=F(x)$ that can serve as a reasonable approximation of $y=f(x)$ in which the function $f(x)$ is unknown. The variables $x = [x_1,...,x_p]' \in \aleph \subset \Re^p$ and $y \in \mathrm{Y} \subset \Re$ are called *regressor* and *regressand*, respectively. The variable $\boldsymbol{x}$ is called an antecedent variable and the variable $\boldsymbol{y}$ is called a consequent variable. The function $F(x)$ is represented as a collection of ***IF-THEN*** fuzzy rules as:

$$\textbf{\textit{IF}}\text{ antecedent proposition }\textbf{\textit{THEN}}\text{ consequent proposition} \tag{8}$$

The linguistic fuzzy model of Zadeh and Mamdani, cited in [8] consists of rules R_i, where both the antecedent and the consequent are fuzzy propositions:

$$R_i\text{: }\textbf{\textit{If}}\text{ x(k) }\textbf{\textit{is}}\text{ A}_i\text{ }\textbf{\textit{then}}\text{ y(k) }\textbf{\textit{is}}\text{ B}_i\text{, i = 1, 2,......, P} \tag{9}$$

Where: A_i and B_i are linguistic terms (labels) defined by fuzzy sets $\mu_{Ai}(x){:}X \rightarrow [0, 1]$ and $\mu_{Bi}(y){:}\ Y\rightarrow[0, 1]$, respectively; $\mu_{Ai}(x),\ \mu_{Bi}(y)$ are membership functions of the correspondent variables; R_i denotes the i-th rule and P is the number of rules in the rule base. The model of Takagi and Sugeno, [7], is a mixture between linguistic and mathematical regression models, where the rule consequent is crisp mathematical function of the inputs. The T-S model has the most general form:

$$R_i\text{: }\textbf{\textit{If}}\text{ x(k) }\textbf{\textit{is}}\text{ A}_i\text{ }\textbf{\textit{then}}\text{ y}_i\text{ (k) = f}_i\text{ [x(k)], i=1,2,..,P} \tag{10}$$

The consequent part of the T-S model (10) could be also a dynamic state space model [8], [9]. So, the T-S model could be rewritten in the form:

$$R_i\text{: }\textbf{\textit{If}}\text{ x(k) }\textbf{\textit{is}}\text{ A}_i\text{ }\textbf{\textit{and}}\text{ u(k) }\textbf{\textit{is}}\text{ B}_i$$

$$\textbf{\textit{then}}\ \begin{cases} \text{x}_i\text{(k+1) = J}_i\text{ x}_i\text{(k) + B}_i\text{ u(k)} \\ \text{y}_i\text{(k)} \quad \text{= C}_i\text{ x(k)} \end{cases} \tag{11}$$

Where: in the antecedent part A_i and B_i are the above mentioned linguistic terms; in the consequent part, $x_i(k)$ is the variable associated with the i-th sub-model state; $y_i(k)$ is the i-th sub-model output; $J_i,\ B_i,\ C_i$ are parameters of this sub-model (J_i is a diagonal matrix). The paper [10] makes a step ahead proposing that the consequent function is a RTNN model (1), (2). So the fuzzy-neural rule obtains the form:

$$R_i\text{:}\textbf{\textit{If}}\text{ x(k) }\textbf{\textit{is}}\text{ J}_i\text{ }\textbf{\textit{and}}\text{ u(k) }\textbf{\textit{is}}\text{ }B_i\text{ }\textbf{\textit{then}}\text{ y}_i\text{(k+1) = N}_i\text{ [x}_i\text{(k),u(k)], i =1,2,..,P} \tag{12}$$

Where: the function $y_i(k+1) = N_i\ [x_i(k),u(k)]$ represents the RTNN**,** given by the equations (1), (2); i - is the number of the function and P is the total number of RTNN approximation functions. The biases, obtained in the process of BP learning of the RTNN model could be used to form the membership functions, as they are natural centers of gravity for each variable, [10]. The number of rules could be optimized using the Mean-Square Error ($MSE\%< 2.5\%$) of RTNN's learning. As the local RTNN model could be learned by the local error of approximation $E_i = Y_{di} - Y_i$, the rule (12) could be extended changing the neural function by the learning procedure $Y = \Pi\ (L, M, N, Y_d, U, X, J, B, C, E)$, given by the equations (1)-(6). In this case the rule (12) could be rewritten as:

$$R_i\text{:}\textbf{\textit{If}}\text{ x(k) }\textbf{\textit{is}}\text{ J}_i\text{ }\textbf{\textit{and}}\text{ u(k) }\textbf{\textit{is}}\text{ B}_i\text{ }\textbf{\textit{then}}\text{ Y}_i\text{ = }\Pi_i\text{ (L,M,N}_i\text{,Y}_{di}\text{,U,X}_i\text{,J}_i\text{,B}_i\text{,C}_i\text{,E}_i\text{),} \\ \text{i=1,2,..,P} \tag{13}$$

The output of the fuzzy neural multi-model system, represented by the upper hierarchical level of defuzzyfication is given by the following equation:

$$Y(k) = \Sigma_i\, w_i\, y_i(k);\; w_i = \mu_i(y) \,/\, [\Sigma_i\, \mu_i(y)] \tag{14}$$

Where w_i are weights, obtained from the membership functions $\mu_i(y)$. As it could be seen from the equation (14), the output of the fuzzy-neural multi-model, approximating the nonlinear plant, is a weighted sum of the outputs of RTNN models, appearing in the consequent part of (13). The weights w_i depend on the form of the membership functions which is difficult to choose. We propose to augment the level of adaptation of the fuzzy-neural multi-model creating an upper hierarchical level of defuzzification which is a RTNN with inputs $y_i(k)$, $i=1,\ldots, P$. So, the equation (14) is represented like this:

$$Y = \Pi\, (L, M, N, Y_d, Yo, X, J, B, C, E) \tag{15}$$

Where: the input vector Yo is formed from the vectors $y_i(k)$, $i = 1,\ldots, P$; $E =. Y_d - Y$ is the error of learning; $\Pi(.)$ is a RTNN learning procedure, given by equations (1)-(6). So, the output of the upper hierarchical defuzzyfication procedure (15) is a filtered weighted sum of the outputs of the T-S rules. As the RTNN is a universal function approximator, the number of rules P could be rather small, e.g. $P = 3$ (negative, zero, and positive) in the case of overlapping membership functions and $P = 2$ (negative and positive), in the case of non-overlapping membership functions. The stability of this HFNMM could be proven via linearization of the activation functions of the RTNN models and application of the methodology, given in [6]. As follows, in both HFNMM identification and control systems proposed, the three fuzzyfication intervals for the reference signal and the plant output are to be the same. A block-diagram of the dynamic system identification, using a HFNMM identifier is given on Fig.1b. The structure of the entire identification system contains a Fuzzyfier, a Fuzzy Rule-Based Inference System (FRBIS), containing up to three T-S rules (20), and a defuzzyfier. The system uses a RTNN model as an adaptive, upper hierarchical level defuzzyfier (15). The local and global errors used to learn the respective RTNNs models are $E_i(k) = Y_{di}(k) - Y_i(k)$; $E(k) = Y_d(k) - Y_i(k)$. The HFNMM identifier has two levels – Lower.

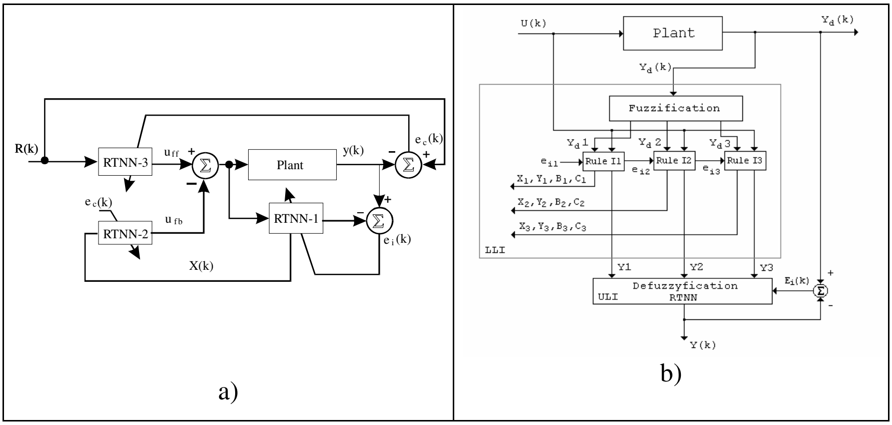


Fig. 1. Block-diagrams; **a)** Block-diagram of the direct adaptive RTNN control system; b) Detailed block-diagram of the HFNMM identifier

Hierarchical Level of Identification (LLI), and Upper Hierarchical Level of Identification (ULI). It is composed of three parts: 1) Fuzzyfication, where the normalized plant output signal $Y_d(k)$ is divided in three intervals (membership functions - μ_i): positive [1, -0.5], negative [-1, 0.5], and zero [-0.5, 0.5]; 2) Lower Level Inference Engine, which contains three T-S fuzzy rules, given by (20), and operating in the three intervals. The consequent part (procedure) of each rule has the L, M, N_i RTNN model dimensions, Y_{di}, U, E_i inputs and Y_i (used as entry of the defuzzyfication level), X_i, J_i, B_i, C_i outputs, used for control. The T-S fuzzy rule is:

$$R_i\text{: } \textbf{\textit{If}} \text{ Yd(k) } \textbf{\textit{is}} \ A_i \ \textbf{\textit{then}} \ \mathrm{Y_i} = \Pi_i \ (\mathrm{L, M, N_i, Y_{di}, U, X_i, J_i, B_i, C_i, E_i}), \ \mathrm{i} = 1{,}2, 3 \qquad (16)$$

3) Upper Level Defuzzyfication, which consists of one RTNN learning procedure, doing a filtered weighted summation of the outputs Y_i of the lower level RTNNs. The defuzzyfication learning procedure (15) has L, M, N RTNN model dimensions, Y_i ($P=3$), E, inputs, and $Y(k)$ - output. The learning and functioning of both levels is independent. The main objective of the HFNMM identifier is to issue states and parameters for the HFNNMM controller when its output follows the output of the plant with a minimum error of approximation. The tracking control problem consists in the design of a controller that asymptotically reduces the error between the plant output and the reference signal. The block diagram of this direct adaptive control is schematically depicted in Fig.2a. The identification part on the right contains three RTNNs, corresponding to the three rules, fired by the fuzzyfied plant output and taking part of the FRBIS HFNMM identifier, and the RTNN DF1 represents the defuzzyfier of the HFNMM identifier (see Fig.1b). The control part on the left contains three double RTNN blocks. The RTNN-Uff block represented the feedforward part of the control, corresponding to the rule fired by the fuzzyfied reference, and the RTNN-Ufb block represented the feedback part rule, fired by the fuzzyfied plant output, and its entries are the corresponding states, issued by the HFNMM identifier. The RTNN DF2 represents the defuzzyfier of the HFNMM controller. The detailed structure of the direct adaptive HFNMM controller is given on Fig.2b. The structure of the entire control system has a Fuzzyfier, a Fuzzy Rule-Based Inference System (FRBIS), containing up to six T-S FF and FB rules, and a defuzzyfier. The system uses a RTNN model as an adaptive, upper hierarchical level defuzzyfier, given by equation (15). The local and global errors used to learn the respective RTNNs models are $E_{ci}(k) = R_i(k) - Y_{di}(k)$; $E(k) = R(k) - Y_d(k)$. The HFNMM controller has two levels – Lower Hierarchical Level of Control (LLC), and Upper Hierarchical Level of Control (ULC). It is composed of three parts: 1) Fuzzyfication, where the normalized reference signal $R(k)$ is divided in three intervals (membership functions - μ_i): positive [1, -0.5], negative [-1, 0.5], and zero [-0.5, 0.5]; 2) Lower Level Inference Engine, which contains six T-S fuzzy rules (three for the feedforward part and three for the feedback part), operating in the corresponding intervals. The consequent part of each feedforward control rule (the consequent learning procedure) has the M, L, N_i RTNN model dimensions, R_i, Y_{di}, E_{ci} inputs and U_{ffi}, outputs used to form the total control. The T-S fuzzy rule has the form:

$$R_i\text{: } \textbf{\textit{If}} \text{ R(k) } \textbf{\textit{is}} \ B_i \ \textbf{\textit{then}} \ \mathrm{Uff_i} = \Pi_i \ (\mathrm{M, L, N_i, R_i, Y_{di}, X_i, J_i, B_i, C_i, E_{ci}}), \ \mathrm{i} = 1{,}2{,}3 \qquad (17)$$

The consequent part of each feedback control rule (the consequent learning procedure) has the M, L, N_i RTNN model dimensions, Y_{di}, X_i, E_{ci} inputs and U_{fbi}, outputs used to form the total control. The T-S fuzzy rule has the form:

$$R_i\text{: } \boldsymbol{If}\ \mathrm{Y_{di}}\ \boldsymbol{is}\ A_i\ \boldsymbol{then}\ \mathrm{Ufb_i = \Pi_i\ (M, L, N_i, Y_{di}, X_i, X_{ci}, J_i, B_i, C_i, E_{ci}),\ i = 1,2,\ 3} \qquad (18)$$

The total control corresponding to each membership function is a sum of its corresponding feedforward and feedback parts:

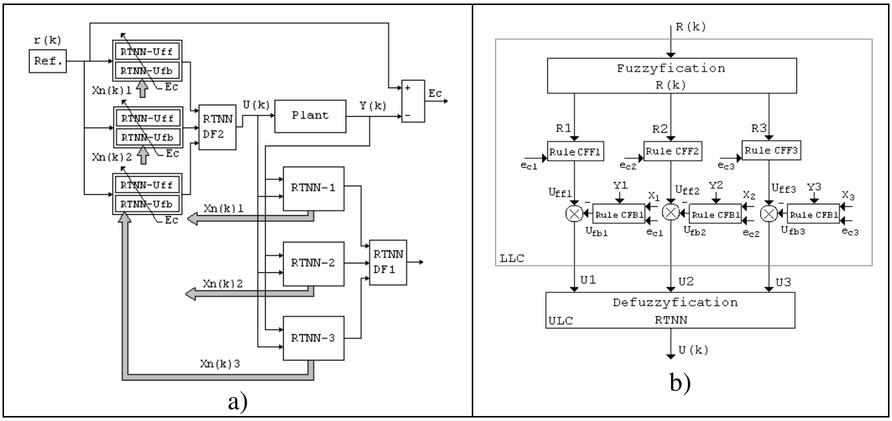


Fig. 2. Block-diagrams; **a)** Block-diagram of the direct adaptive fuzzy-neural multi-model control system; b) Detailed block-diagram of the HFNMM controller

$$U_i\,(k) = -U_{ffi}\,(k) + U_{fbi}\,(k) \qquad (19)$$

3) Upper Level Defuzzyfication which consists of one RTNN learning procedure, doing a filtered weighted summation of the control signals U_i of the lower level RTNNs. The defuzzyfication learning procedure is described by:

$$U = \Pi\,(M, L, N, Y_d, Uo, X, J, B, C, E) \qquad (20)$$

It has *M, L, N* RTNN model dimensions, the vector *Uo* contains U_i (*P=3*), and the control error E_c, is an input. The learning and functioning of both levels is independent. The main objective of the HFNMM controller is to reduce the error of control, so the plant output to track the reference signal.

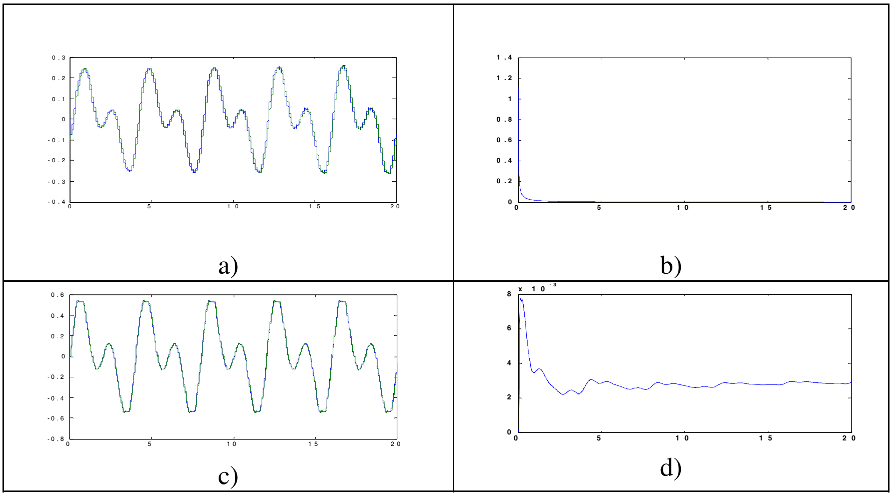
4 Simulation Results

Let us consider a DC-motor - driven nonlinear 1-DOF mechanical system with friction, [12], to have the following friction parameters: $\alpha = 0.001\ m/s$; $F_s^+ = 4.2\ N$; $F_s^- = -\ 4.0\ N$; $\Delta F^+ = 1.8\ N$; $\Delta F^- = -\ 1.7\ N$; $v_{cr} = 0.1\ m/s$; $\beta = 0.5\ Ns/m$. The position and velocity measurements are taken with period of discretization $To = 0.1s$, the system gain $ko = 8$, the mass $m = 1\ kg$, and the load disturbance depends on the position and the velocity, ($d(t) = d_1q(t) + d_2v(t); d_1 = 0.25; d_2 = - 0.7$). The discrete-time model of the 1-DOF mechanical system with friction is given as:

$$x_1(k+1) = x_2(k); x_2(k+1)=-0.025x_1(k)-0.3x_2(k)+0.8u(k)-0.1fr(k); \qquad (21)$$

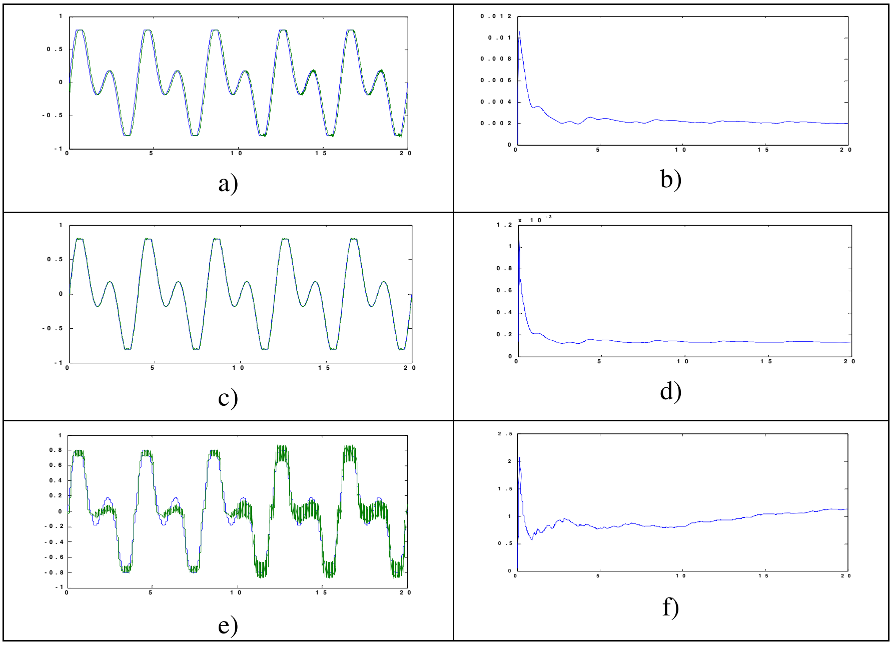
$$v(k) = x_2(k) - x_1(k); y(k) = 0.1\, x_1(k) \qquad (22)$$

Where: $x_1(k), x_2(k)$ are system states; $v(k)$ is shaft velocity; $y(k)$ is shaft position; $fr(k)$ is a friction force, taken from [12]. The topology of the identification and FF control RTNNs is (1, 5, 1), and that – of the FB control RTNN is (5, 5, 1). The topology of the defuzzyfication RTNN is (1, 3, 1). The learning rate parameters are $\eta = 0.01$, $\alpha = 0.9$ and $To = 0.01$ *sec*. The reference signal is $r(k) = sat\ [0.5 \sin(\pi k) + 0.5 \sin(\pi k/2)]$ with a saturation level as ± 0.8. The graphics of the comparative identification simulation results are given on Fig.3a-d. The graphics of the comparative simulation results, obtained for different control systems, are shown on Fig. 4.a-f. The identification results show that the HFNMM identifier outperformed the RTNN identifier (the MSE% is *1.2%* vs. *2.5%*, respectively). The results of control show that the MSE% of control has final values which are: *0.41%* for the direct adaptive HFNMM control; *2.7%* for the control with single RTNNs, and *5.8%* for the fuzzy control. From the graphics of Fig. 4a-d, we could see that the direct adaptive HFNMM control is better than that, using single RTNNs. From Fig. 4e,f we could see that the fuzzy control is worse with respect to the neural control, especially when the friction parameters changed.



a) b) c) d)

Fig. 3. Comparative closed-loop system identification results; a) comparison of the plant output and the output of a single RTNN identifier; b) MSE% of RTNN identification; c) comparison of the plant output and the output of a HFNMM identifier; d) MSE% of HFNMM identification



a)

b)

c)

d)

e)

f)

Fig. 4. Comparative trajectory tracking control results; a) comparison of the reference signal and the output of the plant using RTNN controllers; b) MSE% of RTNN control; c) comparison of the reference signal and the output of the plant using HFNMM controller; d) MSE% of HFNMM control; e) comparison of the reference signal and the output of the plant using fuzzy controller; d) MSE% of fuzzy control

5 Conclusion

The present paper proposed a new identification and direct adaptive control system based on the HFNMM. The HFNMM has three parts: 1) fuzzyfication, where the output of the plant is divided in three intervals μ (positive [1, -0.5], negative [-1, 0.5], and zero [-0.5, 0.5]); 2) inference engine, containing a given number of T-S rules corresponding to given number of RTNN models operating in the given intervals μ; 3) defuzzyfication, which consists of one RTNN doing a filtered weighted summation of the outputs of the lower level RTNNs. The learning and functioning of both hierarchical levels is independent. The HFNMM identifier is incorporated in a direct feedback/feedforward control scheme, using a HFNMM controller. The proposed HFNMM-based control scheme is applied for a 1-DOF mechanical plant with friction control. The comparative simulation results show the superiority of the proposed HFNMM control with respect to the others.

References

[1] K.J. Hunt, D. Sbarbaro, R. Zbikowski, P.J. Gawthrop, Neural network for control systems (A survey), Automatica 28 (1992) 1083-1112.

[2] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks, 1 (1) (1990) 4-27.

[3] P.S. Sastry, G. Santharam, K.P. Unnikrishnan, Memory networks for identification and control of dynamical systems, IEEE Trans. on Neural Networks, 5 (1994) 306-320.

[4] P. Frasconi, M. Gori, G. Soda, Local feedback multilayered networks, Neural Computation, 4 (1992) 120-130.

[5] I. Baruch, J.M. Flores, F. Thomas, R. Garrido, Adaptive neural control of nonlinear systems, Proc. of the Int. Conf on NNs, ICANN 2001, Vienna, Austria, August 2001, G. Dorffner, H. Bischof, K. Hornik, Edrs., Lecture Notes in Computer Science 2130, Springer, Berlin, Heidelberg, N. Y., 2001, 930-936.

[6] I. Baruch, J.M. Flores, F. Nava, I.R. Ramirez, B. Nenkova, An advanced neural network topology and learning applied for identification and control of a D.C. motor, Proc. of the 1-st Int. IEEE Symp. on Intel. Syst., Varna, Bulgaria, Sept., 2002, 289-295.

[7] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems, Man, and Cybernetics, 15 (1985) 116-132.

[8] R. Babuska, Fuzzy Modeling for Control, Norwell, MA, Kluwer, NY, 1998.

[9] P.A. Mastorocostas, J.B. Theocharis, A recurrent fuzzy-neural model for dynamic system identification, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 32 (2002) 176-190.

[10] I. Baruch, J.M. Flores, R. Garrido, A fuzzy-neural recurrent multi-model for systems identification and control, Proc. of the European Control Conference, ECC’01, Porto, Portugal, Sept. 4-7, 2001, 3540-3545.

[11] J.B. Theocharis, A high-order recurrent neuro-fuzzy system with internal dynamics: Application to the adaptive noise cancellation, Fuzzy Sets and Syst., 157 (4) (2006) 471-500.

[12] S.W. Lee, J.H. Kim, Robust adaptive stick-slip friction compensation, IEEE Transaction on Industrial Electronics, 42 (5) (1995) 474-479.

Robust Speed Controller Design Method Based on Fuzzy Control for Torsional Vibration Suppression in Two-Mass System

Eun-Chul Shin[1], Tae-Sik Park[1], Ji-Yoon Yoo[1], and Dong-Sik Kim[2]

[1] Department of Electrical Engineering, Korea University, 1, 5-ga, Anam-dong, Seongbuk-gu, Seoul, Korea
churky@korea.ac.kr, points@naver.com, jyyoo@korea.ac.kr

[2] Division of Information Technology Engineering, Soonchunhyang University, 546, Shinchang-myun, Asan, Chungnam, Korea
dongsik@sch.ac.kr

Abstract. This paper presents a robust speed controller design method based on fuzzy logic control(FLC) for robust torsional vibration suppression control scheme in rolling mill drive system. This method proposes a torsional vibration suppression controller that comprises a reduced-order state feedback controller and a PI controller whose motor speed and observed torsional torque are fed back. By using the mechanical parameters estimated by an off-line recursive least square algorithm, a speed controller for torsional vibration suppression and its gains can be determined by FLC with the Kharitonov's robust control theory. This method can yield a robust stability with a specified stability margin and damping limit. Even if the parameters are varied within some specified range, the proposed control method guarantees a highly efficient vibration suppression. By using a fully digitalized 5.5 kW rolling mill drive system, the effectiveness and usefulness of the proposed scheme are verified and obtained experimental results.

Keywords: fuzzy logic control, reduced-order state feedback, robust stability, rolling mill drive system, torsional vibration suppression.

1 Introduction

Equipment such as flexible robot manipulator, paper and iron manufacture, and steel rolling mill drive system have been employed in major industrial applications. If the motor is linked with a load by a long flexible shaft, the assembled system becomes a mechanical resonance system termed a two-mass system. In industrial applications such as the rolling mill drive system, the mechanical section of the driver has a very low natural resonance frequency (approximately tens of Hz) because of a large roll inertia and long shaft. As a result, the rolling mill drive system generates vibrations in the speed control response. These vibrations result in shaft damage, scratches on the iron surface, and deterioration in the product quality.

The solutions to these problems have been studied using various procedures such as the torsional vibration compensation method, two-degrees-of-freedom (TDOF) method, simulator and model reference following control method, and state feedback

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 173–184, 2007.
springerlink.com

method [1]–[3]; further, several advanced control algorithms such as robust control, adaptive control, and neural network control theory [4][5] have been used. The most promising approach for vibration suppression is the (observer-based) state feedback control. But this approach essentially requires the order of the observer to be greater than three [2][4]. Although the results obtained from the observer-based state feedback controller that uses a pole placement method have been studied, no decisive control methods for achieving controller gain selection have been obtained. This is mainly due to variations in the system parameters during operation. In addition, the procedure for tuning the control parameters is not described in detail in many papers. Conventional algorithms do not yield a highly effective control performance because they provide a partial torsional vibration suppression performance, the algorithm is complex, and the design procedure for mechanical parametric variation are not considered [3].

In this paper, a novel robust torsional vibration suppression controller for a two-mass system such as the rolling mill drive system is proposed. The controller consists of a reduced-order state feedback controller, state observer, and PI speed controller based on FLC. In addition, the aim of the proposed control scheme is the design of the simplest controller along with obtaining the maximum torsional vibration suppression in practical industrial applications using FLC. A closed-loop system is made robust by applying the Kharitonov's robust stability theory to the mechanical parameter variations and their uncertainties.

By using a DSP-based fully digitalized 5.5 kW rolling mill drive system, the effectiveness and usefulness of the proposed scheme are verified on the basis of a simulation and the obtained experimental results

2 Rolling Mill Drive System

Fig. 1 shows the block diagram of a rolling mill drive system [2][3]. On the basis of Fig. 1, the state-space equation of a two-mass system and the rolling mill drive system is expressed as follows.

$$\begin{bmatrix} \dot{\omega}_M \\ \dot{\theta}_{SH} \\ \dot{\omega}_R \end{bmatrix} = \begin{bmatrix} 0 & -K_{SH}/J_M & 0 \\ 1 & 0 & -1 \\ 0 & K_{SH}/J_R & 0 \end{bmatrix} \begin{bmatrix} \omega_M \\ \theta_{SH} \\ \omega_R \end{bmatrix} + \begin{bmatrix} 1/J_M \\ 0 \\ 0 \end{bmatrix} T_M - \begin{bmatrix} 0 \\ 0 \\ 1/J_R \end{bmatrix} T_L \ . \tag{1}$$

Here, J_M and J_R are the moment of inertias of the motor and the load, respectively, and T_M and T_L are the motor torque and disturbance input, respectively. ω_M , ω_L , θ_{SH} and T_{SH} are the motor speed, load speed, torsional angle of the shaft, and shaft stiffness, respectively. The transfer function between the motor speed and the induced torque is expressed as follows.

$$G(s) = \frac{\omega_M}{T_M} = \frac{1}{s} \frac{J_R s^2 + K_{SH}}{J_M J_R s^2 + K_{SH}(J_M + J_R)} \ . \tag{2}$$

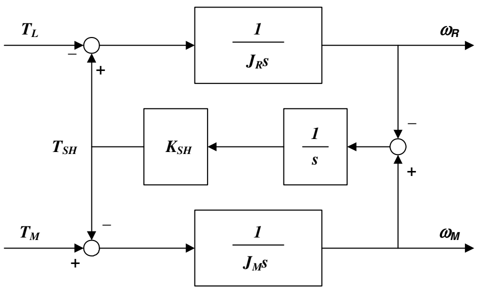


Fig. 1. Block diagram of the rolling mill drive system

From (2), the occurrence of mechanical resonance characteristics is verified. The mechanical resonance frequency and antiresonance frequency are important characteristics of a two-mass system, which are represented as (3) and (4), respectively.

$$\omega_r = \sqrt{\frac{K_{SH}}{J_R}\left(1+\frac{J_R}{J_M}\right)} \;. \tag{3}$$

$$\omega_a = \sqrt{\frac{K_{SH}}{J_R}} \;. \tag{4}$$

3 Proposed Torsional Vibration Suppression Method

Fig. 2 (a) shows the total block diagram of the proposed reduced-order state feedback controller including a first-order observer of torsional torque and a proposed PI speed controller.

The transfer function between the motor speed and induced torque is expressed as follows.

$$\frac{\omega_M}{T_{\text{in}d}} = \frac{(J_R s^2 + K_{SH})}{J_M J_R s^3 - J_R K_{\omega_M} s^2 + \left[\left(J_R + J_M\right) K_{SH} - J_R K_{T_{SH}} K_{SH}\right] s - K_{SH} K_{\omega_M}} \;. \tag{5}$$

Here, $K_{T_{sh}}$ and K_{ω_M} are the feedback gains of the torsional torque and motor speed, respectively. In order to compensate for the errors between the actual speed and the reference speed, the reduced-order state feedback controller, torsional torque observer, and PI controller are used. Fig. 2 (b) shows the block diagram of the torsional torque observer. From the state-space equation, an expression for T_{SH} is obtained as (6); further, the value of T_{SH} is sensitive to noise because it contains the differential term of motor speed.

$$T_{SH} = T_M - J_M \dot{\omega}_M \;. \tag{6}$$

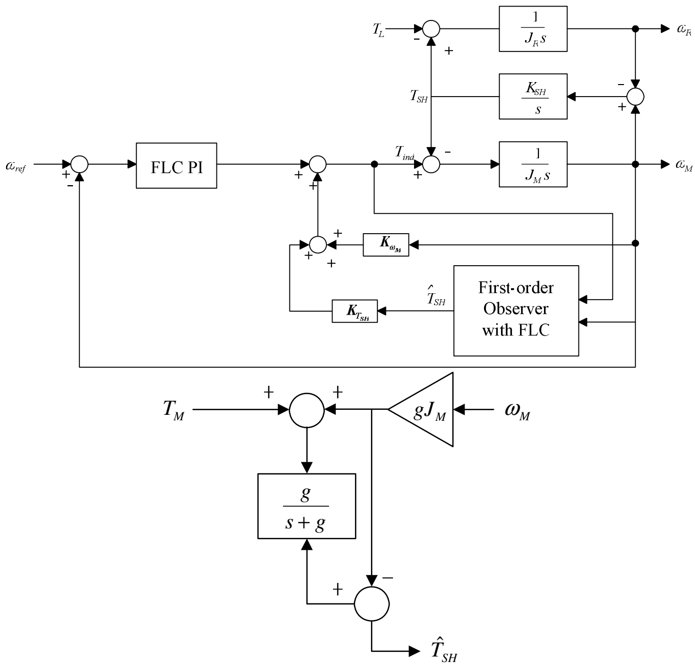


Fig. 2. The proposed reduced-order state feedback and PI controller (a) Block diagram (b) Torsional torque observer using a low-pass filter

The single-state observer obtained using a low-pass filter is employed to obtain the torsional torque information. The proposed observer does not require a pure differentiator because of the low-pass filtering of the observed state output. Therefore, it is more advantageous to apply it to noisy working environments such as those in rolling mill drive systems. The transfer function of the torsional torque observer is expressed as follows, where g denotes the filter gain.

$$\begin{aligned}\hat{T}_{SH} &= (T_M - gJ_M\omega_M)\frac{g}{s+g} - gJ_M\omega_M \ . \\ &= \frac{g}{s+g}(T_M - J_M\dot{\omega}_M)\end{aligned} \tag{7}$$

4 Design Procedure of the Reduced-Order State Feedback Controller and the PI Controller with Robust Stability Based on FLC

4.1 Design of a Reduced-Order State Feedback Controller with δ and Θ Hurwitz Stability

From (5), after the state feedback, the transfer function between the motor torque and speed can be expressed as (8). It is assumed that in the rolling mill drive system, only the roll inertia among all the other mechanical parameters varies during operation. Actually, the shaft stiffness and inertia of the induction motor are estimated from the mechanical parameter estimation, and these parameters seldom vary during motor operation. (9) indicates the variable ranges of roll inertia, mechanical resonance frequency, and antiresonance frequency; (10) represents the maximum and minimum values of the mechanical resonance and antiresonance frequencies.

$$G(s)=\frac{N(s)}{D(s)}=\frac{b_2 s^2+b_0}{a_3 s^3+a_2 s^2+a_1 s+a_0}\ .$$

where $b_2=\frac{1}{J_M}, b_1=0, b_0=\frac{\omega_a{}^2}{J_M}$. (8)

$$a_3=1, a_2=-\frac{K_{\omega_M}}{J_M},\ a_1=\omega_r{}^2-\frac{K_{SH}K_{T_{SH}}}{J_M}, a_0=-\frac{\omega_a{}^2 K_{\omega_M}}{J_M}$$

$$J_L\in[J_{L\min},J_{L\max}],\omega_r{}^2\in[\omega_{r\min}{}^2,\omega_{r\max}{}^2],\omega_a{}^2\in[\omega_{a\min}{}^2,\omega_{a\max}{}^2]\ . \tag{9}$$

$$\omega_{r\min}{}^2=K_{SH}[\frac{1}{J_M}+\frac{1}{J_{L\max}}],\omega_{r\max}{}^2=[\frac{1}{J_M}+\frac{1}{J_{L\min}}],\ \omega_{a\min}{}^2=\frac{K_{SH}}{J_{L\max}},\ \omega_{a\max}{}^2=\frac{K_{SH}}{J_{L\min}}\ \cdot \tag{10}$$

Equation (11) shows the variation in the coefficients of the transfer function due to variations in the roll inertias and the maximum/minimum value of each coefficient. Since variations in the roll inertias do not affect the coefficients of the third- and second-order terms (a_3, a_2) in the polynomial *D(s)* , it influences the coefficients of the first- and zero-order terms (a_1, a_0) shown in (8).

$$\begin{gathered}a_3=1,\quad a_2=-\frac{K_{\omega_M}}{J_M}\ ,(a_1>0)\in\left[a_{1\min},a_{1\max}\right],\quad (a_0>0)\in\left[a_{0\min},a_{0\max}\right]\\ a_{1\min}=\omega_{r\min}{}^2-\frac{K_{SH}K_{T_{SH}}}{J_M},\quad a_{1\max}=\omega_{r\max}{}^2-\frac{K_{SH}K_{T_{SH}}}{J_M},a_{0\min}=-\frac{\omega_{a\min}{}^2K_{\omega_M}}{J_M},\quad a_{0\max}=-\frac{\omega_{a\max}{}^2K_{\omega_M}}{J_M}\end{gathered} \tag{11}$$

According to the Kharitonov's robust stability theory, the system poles get located on the left-half side of the s plane, where the parameters vary in each specific range and the stability margin and system damping are not considered [6]. By extending this method, a robust stability controller can yield the minimum performance with regard to vibration suppression along with the parameter variations. The characteristic equation of the closed-loop system can be modified by including an additional stability margin δ and damping limit θ. By applying the GKT to the modified characteristic equation, the additional stability margin and damping limit can be obtained. That is, the system poles can be located at a certain location such that the stability margin is δ and the damping limit exceeds θ. The (12) shows the characteristic equation of the reduced-order state feedback system including the additional stability margin in the s-plane and each of the coefficients are as (12). The modified characteristic equation of the reduced-order state feedback system that is rotated by Θ is expressed as (13).

$$\begin{aligned} \Delta(s-\delta) &= D(s-\delta) \\ &= c_3 s^3 + c_2 s^2 + c_1 s^1 + c_0 s^0 \\ \text{where} \quad & c_3 = a_3 = 1, c_2 = a_2 - 3a_3\delta \\ & c_1 = -2a_2\delta + 3a_3\delta^2 + a_1, c_0 = -a_1\delta + a_0 + a_2\delta^2 - a_3\delta^3 \end{aligned} \tag{12}$$

$$\begin{aligned} \Delta(s_1) = (c_3 + jd_3)s_1^{\ 3} + (c_2 + jd_2)s_1^{\ 2} + (c_1 + jd_1)s_1^{\ 1} + (c_0 + jd_0) = 0 \\ \text{where} \quad c_3 = a_3\left((\cos\theta)^3 - 3\cos\theta(\sin\theta)^2\right), d_3 = a_3\left(3(\cos\theta)^2\sin\theta - (\sin\theta)^3\right) \\ c_2 = a_2\left((\cos\theta)^2 - (\sin\theta)^2\right), \qquad d_2 = a_2 2\cos\theta\sin\theta \\ c_1 = a_1\cos\theta, \ \ d_1 = a_1\sin\theta, c_0 = a_0, \ \ d_0 = 0 \end{aligned} \tag{13}$$

It is possible to obtain a stability margin of δ in the closed-loop system if the gains of the state feedback controller are selected such that the two extremal polynomials may be stable according to the Routh-Hurwitz stability criterion. Thus, a closed-loop system maintains a stability margin of δ against a specific variation in the roll inertia. Fig. 3(c) shows the common area in the additional stability margin with an appointed ±70% variation in the roll inertia. By applying the Hurwitz stability criterion for complex polynomials, it obtains the gain limits for the reduced-order state feedback controller such that $\Delta(s_1)$is stable. If the feedback gains are selected such that they just

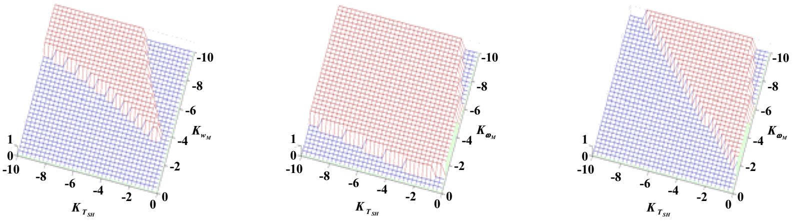


Fig. 3. State feedback gain area that satisfies the stability condition with an additional stability margin of 5 (a) Variation in roll inertia = +70% (b) Variation in roll inertia = –70% (c) Variation in roll inertia = ±70%

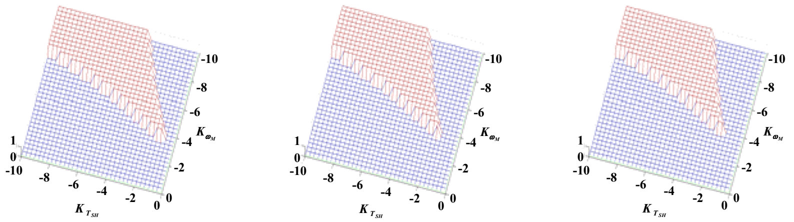


Fig. 4. State feedback gain area that satisfies the stability condition within a specific Θ= ±45° (a) Variation in roll inertia = +70% (b) Variation in roll inertia = –70% (c) Variation in roll inertia = ±70%

satisfy the stability condition, a two-mass system can acquire the required damping property and arbitrarily located system poles within the limited damping area. Fig. 4(c) shows the state feedback gain area that satisfies the stability condition with a specific additional damping of ±45° with respect to the variation in the roll inertia.

Fig. 5 (a) shows the state feedback controller gain areas that possess an additional stability margin of δ and a damping property of Θ using the common areas shown in Fig. 3(c) and Fig. 4(c). The common satisfactory area becomes FLC decision rule for optimal gain selection. Then reduced order state feedback gains are selected by conventional fuzzy set point weighting method. The approach proposed by Visioli [7] consists of fuzzifying the set-point weight, leaving fixed the other parameters. The fuzzy rules are based on the Macvicar-Whelan matrix [8]-[11].

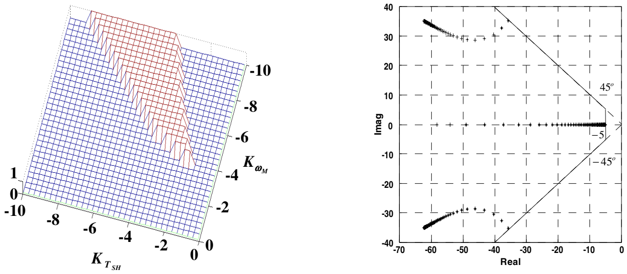


Fig. 5. Common satisfactory area for (a) δ and Θ Hurwitz stability conditions, (b) Pole movement of the characteristic equation according to roll inertia variations ($K_{\omega_M} = -7.5$, $K_{T_{SH}} = -5$)

4.2 PI Controller

A PI controller that is in front of the reduced-order state feedback controller can be expressed as follows.

$$G_c(s) = K_p\left(1 + \frac{\omega_{pi}}{s}\right) \tag{14}$$

(14) is the characteristic equation of the closed-loop system that includes a PI controller and stability gain margin. If GKT is applied to design the PI speed controller such as that employed for the reduced-order state feedback controller, it obtains a stable gain area for the PI controller.

Fig. 6(a) shows a satisfactory common area for PI controller gain areas that satisfy the stability condition with a stability margin of 5. Fig. 6(b) illustrates the pole location of the modified characteristic equation according to the roll inertia variation. Although the roll inertia varies within ±70%, the poles of the PI speed controller guarantee a stability gain margin and minimum speed control performance. Then speed PI controller gains are selected by conventional fuzzy set point weighting method and the Macvicar-Whelan matrix.

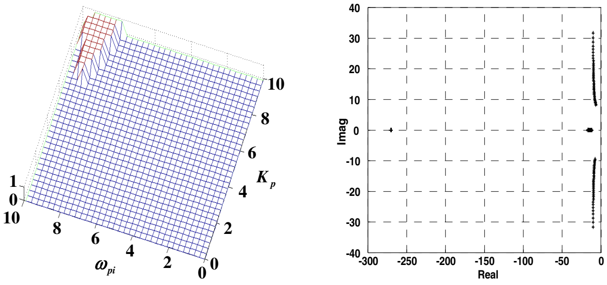


Fig. 6. Common satisfactory area for (a) PI controller gains area that satisfies the stability condition with a stability margin of 5, (b) Pole movement of characteristic equation according to a roll inertia variation of +70% ($k_p = 9$ and $\omega_{pi} = 9$)

5 Construction of the Experiment Set

To verify the effectiveness of the proposed torsional vibration suppression algorithm for a rolling mill drive system, an equivalent experimental set was set up in the laboratory. A photograph of experimental setup of the rolling mill drive system is shown in Fig. 7. The drive system that comprises a PWM IGBT inverter using an intelligent power module and a 5.5 kW induction motor/dc generator was used for the

Table 1. Specifications of the rolling mill drive system experimental setup

Induction Motor		DC Motor	
Power	5.5 kW	Power	5.5 kW
Rated speed	1775 rpm	Rated speed	1800 rpm
Torque	29 Nm	Torque	30 Nm
Inertia	0.04 $kg \cdot m^2$	Inertia	0.04 $kg \cdot m^2$
Torsion Shaft		Inertia Circle Board	
Min. shaft stiffness	44 Nm / rad	Inertia per round disk	0.094 $kg \cdot m^2$

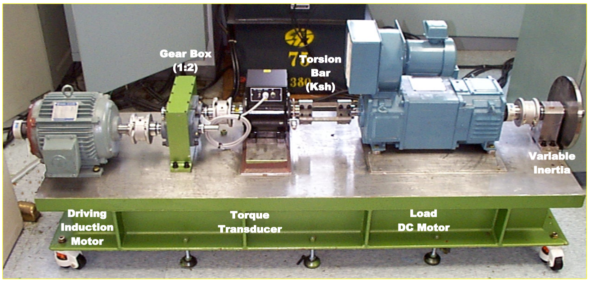


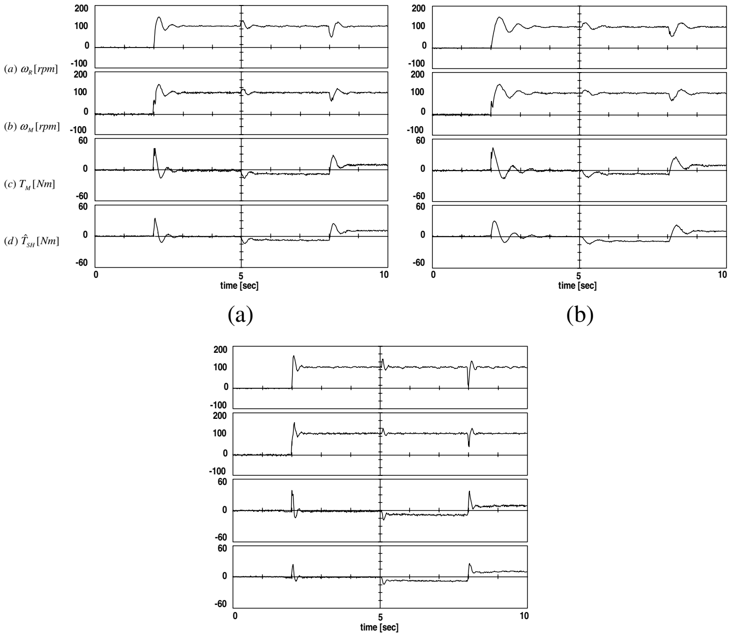
Fig. 7. Experimental setup of the rolling mill drive system

experiments. The proposed control algorithms are implemented by a fully digitalized DSP TMS320C31 controller. Table 1 lists the specifications of the total experimental set for the rolling mill drive system.

6 Experimental Results

The proposed torsional vibration suppression control scheme using FLC presented in the previous section has been experimentally tested. Its performance and properties are verified with the actual rolling mill drive system experimental set. The PWM frequency of the inverter is 8 kHz, the sampling time of the control loop is 125 μs, and the sampling time of speed control is 2 ms. The loads of the experimental equipment consist of a dc generator (0.04 kg•m^2) and an inertia circle board (0.094 kg•m^2). For a reliable evaluation, the proposed torsional vibration suppression control method is compared with a conventional PI controller and state feedback controller method. For an external load test in the steady state, positive (forward) and negative (reverse) loads of 9 *Nm* each are applied to the system during operation at 100 rpm.

Fig. 8 shows the speed and load response characteristics when a conventional PI speed controller is applied to the rolling mill drive system. In designing a PI speed



(a)

(b)

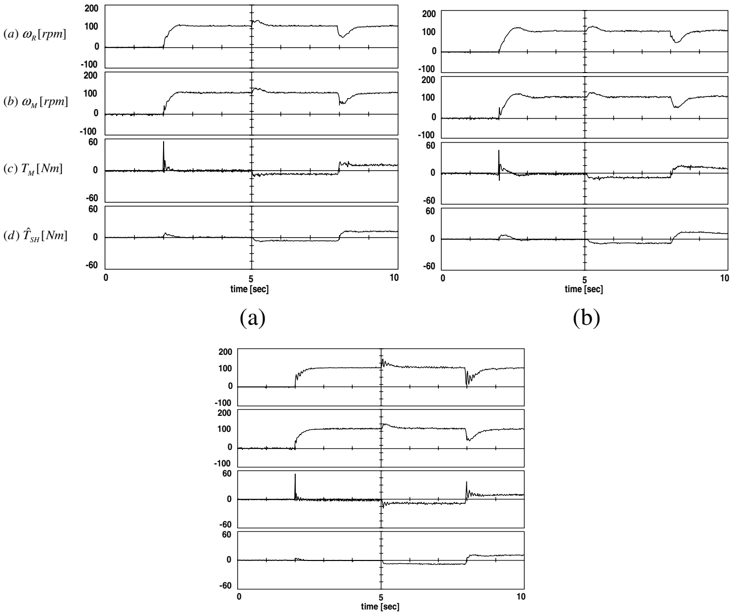
(c)

Fig. 8. Speed and load response characteristics with a conventional PI controller (a) Without variation in roll inertia, (b) Variation in roll inertia: +0.093 kg•m^2, (c) Variation in roll inertia: –0.093 kg•m^2

controller for the rolling mill drive system, the abovementioned combination is considered to be a single inertia system under the assumption that the shaft has an infinite stiffness. As a result, vibration cannot be avoided. Figs. 8(b) and (c) show the characteristics of torsional vibration against the roll inertia variation of ±0.093 kg•m^2 (±70%). If the inertia increases from its rated value to +70%, low frequency vibrations occur and the damping characteristics get changed. As the inertia decreases to –70%, high frequency vibrations occur.

Fig. 9 shows the speed and load response characteristics wherein the state feedback controller is applied to the rolling mill drive system and the gains of the reduced-order state feedback controller that employs all the states such as motor speed, torsional torque, induced torque, and load speed. This system exhibits superior torsional vibration suppression characteristics than a conventional PI controller. However, a state feedback controller exhibits a slower response to the PI controller in the transient region. For states with higher values of roll inertias, low frequency torsional vibrations occur. High-frequency torsional vibrations occur for states with lower values of roll inertias. Remarkably, the conventional PI controller and state feedback controller exhibit deterioration in the control characteristics of vibration suppression when the inertia is varied.

Fig. 10 shows the performance of the proposed control scheme. It shows the speed and load response characteristics. By applying the proposed design scheme to the rolling mill drive system, a highly effective performance with regard to vibration



(a) (b)

(c)

Fig. 9. Speed and load response characteristics with a conventional state feedback controller without FLC (a) Without variation in roll inertia, (b) Variation in roll inertia: +0.093 kg•m^2, (c) Variation in roll inertia: –0.093 kg•m^2

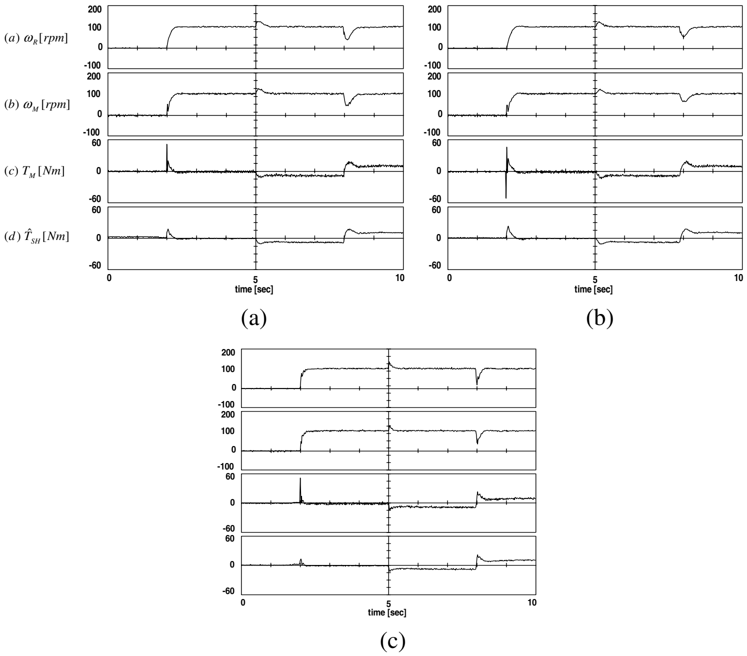


Fig. 10. Speed and load response characteristics with the proposed reduced-order state feedback and PI controller with FLC. (a) Without variation in roll inertia, (b) Variation in roll inertia: +0.093 kg•m^2 (c) Variation in roll inertia: –0.093 kg•m^2

suppression and fast response speed in the transient region can be acquired. Figs. 9(b) and (c) show the speed and load response characteristics when the roll inertia changes by ±0.093 kg•m^2. The speed and load response characteristics of the closed-loop system vary with the load condition; however, the torsional vibration suppression characteristics are more superior to PI controller and state feedback controller. In particular, although the roll inertia varies from its rated value to +70% and –70%, the characteristics of variation suppression are more stable than any other method.

7 Conclusion

In this paper, a novel torsional vibration suppression control method for a rolling mill drive system with robust stability has been proposed. This controller consists of a reduced-order state feedback controller, LPF-based state observer, and PI speed controller based on robust stability. Further, when compared with PI controllers and full-order state feedback control algorithms, this method requires simple implementation, small amount of computation, and yields a higher performance. Moreover, the system convergence property was obtained even if the mechanical parameters vary within the specified region. By applying the off-line RLS algorithm and a robust stability theory to the selection of controller gains, the stability margin for vibration suppression characteristics and system stability can be guaranteed. By using a fully digitalized 5.5 kW rolling mill drive system, the effectiveness and usefulness of the proposed scheme are verified on the basis of a simulation and the obtained experimental results.

References

1. Rahed Dhaouadi, Kenji Kubo, M. Tobise, "Two-Degree-of-Freedom Robust Speed Controller for High performance Rolling Mill Drives", IEEE Trans. on Ind. App., Vol. 29, No. 5, pp. 919, 1993
2. Yoichi Hori, Hiroyuki Iseki, and Koji Sugiura, " Basic consideration of vibration suppression and disturbance rejection control of multi-inertia system using SFLAC (state feedback and load acceleration control)", IEEE Transactions on Industry Applications, Vol.30, No.4, pp:889-896, 1994
3. Koji Sugiura and Yoichi Hory, "Vibration suppression in 2- and 3-mass system based on the feedback of imperfect derivative of the estimated torsional torque", IEEE Trans. on Ind. Elec., Vol.43, No.1, pp.56, 1996
4. Rahed Dhaouadi, Kenji Kubo, "Analysis and Design of a Nonlinear Control Concept Applied to Two-Mass-Model Elastic Drive Systems," AMSE Journal, Advances in Modeling and Analysis, Series C, Vol.53, No.2, pp.27-44, 1999
5. Karsten Peter, Ingo Schöling, and Bernd Orlik, “Robust Output-Feedback H Control with a Nonlinear Observer for a Two-Mass System”, IEEE Trans. on Ind. Appl., Vol.39, No.3, 2003
6. S. P. Bhattacharyya, H. Chapellat and L. H. Keel, Robust Control The Parametric Approach, Prentice-Hall, 1995
7. Visioli, A.: ‘Fuzzy logic based set-point weighting for PID controllers’,IEEE Trans. Syst. Man, Cybern. - Pt. A, 1999, 29, pp. 587-592
8. MACVICAR-WHELAN, P.J.: ‘Fuzzy sets for man-machine interaction’,Int. 1 Man-Mach. Stud., 1976, 8, PP. 687-697
9. J. Carvajal, G. Chen, and H. Ogmen, “Fuzzy PID controller: Design, performance evaluation, and stability analysis,” Inform. Sci., vol. 123,pp. 249–270, 2000.
10. G. Chen and T. T. Pham, Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems. Boca Raton, FL: CRC Press, 2000.
11. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Program, 3rd ed. New York: Springer-Verlag, 1996.

Self-organizing Fuzzy Controller Based on Fuzzy Neural Network

Seongwon Cho[1], Jaemin Kim[1], and Sun-Tae Chung[2]

[1] School of Electronic and Electrical Engineering, Hongik University, 72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea
swcho@hongik.ac.kr

[2] School of Electronic Engineering, Soongsil University, 511 Sangdo-dong, Dongjak-gu, Seoul 156-743, Korea
cst@ssu.ac.kr

Abstract. Fuzzy logic has been successfully used for nonlinear control systems. However, when the plant is complex or expert knowledge is not available, it is difficult to construct the rule bases of fuzzy systems. In this paper, we propose a new method of how to construct automatically the rule bases using fuzzy neural network. Whereas the conventional methods need the training data representing input-output relationship, the proposed algorithm utilizes the gradient of the performance index for the construction of fuzzy rules and the tuning of membership functions. Experimental results with the inverted pendulum show the superiority of the proposed method in comparison to the conventional fuzzy controller.

1 Introduction

Fuzzy logic control(FLC) has been widely applied to industries [1][2][3][4][5] [6][7][8]. These various applications require convenient method that construct fuzzy controller. Thus, the automatic construction of rules has been an important research problem[9][10][11][12][13][14]. Expert knowledge is used to construct knowledge base for rule-based system. In other words, in the knowledge acquisition step, an expert express his knowledge in IF-THEN rules. When an expert is not available, rules can be inferred from input-output relation data.

Differently from non-fuzzy system, fuzzy systems need fuzzy sets used in them. To develop such systems, the membership functions for fuzzy sets as well as fuzzy rules should be defined. Fuzzy rule-based control system has an advantage that knowledge is represented in the form of IF-THEN rules, so knowledge can be formulated as linguistic form in comparison to neural network.

In this paper, we present a new method for the automatic construction of rule-base needed for fuzzy control system based on fuzzy neural network and object function when input-output relation data are not available. Also, we propose a neural network and its learning algorithm to fine-tune the parameters used in the control system.

2 Fuzzy Neural Network

The proposed neural fuzzy network needs two conditions for training. First, an object function for control aim should be selected pertinently. Generally an object function is

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 185–194, 2007.
springerlink.com

not unique. It is important to select the pertinent object function. Let us introduce a general object function that minimizes the sum of squared errors. The object function can be expressed as follows:

$$J = \frac{1}{2}\left\{\alpha_1 (e_1)^2 + \alpha_2 (e_2)^2 + \cdots + \alpha_n (e_n)^2\right\} \tag{1}$$

where α_i denotes the weight of I-th error term and e_i denotes I-th error term.

Secondly, fuzzy sets of input linguistic variable should be determined. In this procedure, we must consider the number of fuzzy sets. The number of fuzzy control rules is determined as the multiplication of the number of fuzzy sets defined in each input linguistic variable. In this paper, a bell-shaped function is used. We must determine the center and the width of the bell-shaped function. If the number of fuzzy sets is n, the universe of discourse is divided into n-1 equal parts and the center is disposed at each partition point. The widths are determined so that membership functions are fully overlapping.

The fuzzy neural network has three modes: the construction of fuzzy control rule, the tuning of parameter, and fuzzy reasoning.3 Iris Feature Extraction.

2.1 Structure

Fig. 1 shows the structure of the proposed fuzzy neural network. The system has four layers. Nodes in Layer 1 are input nodes which represent input linguistic variables. Layer 2 acts as membership functions to represent the terms of the representative linguistic variable. Each node at Layer 3 is a rule node which represents one fuzzy logic rule. Layer 4 is the output layer.

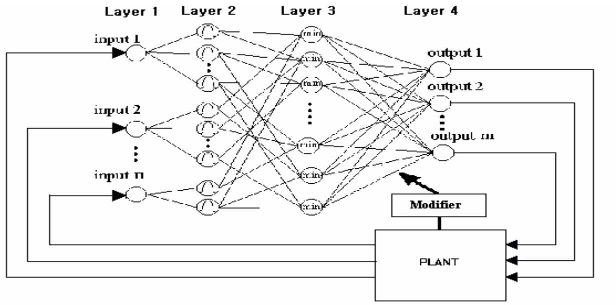


Fig. 1. Block diagram of the proposed system

The following describes the functions of the nodes in each of the four layers of the proposed neural model.

Layer 1: The nodes in this layer transmit input value directly to the next layer.

Layer 2: Each node in this layer perform a bell-shaped membership function. This function is given by

$$f_{ij} = {}^{-\frac{(x_i - m_{ij})^2}{2\sigma_{ij}^2}} \tag{2}$$

where m_{ij} and σ_{ij} are the center and the width of bell-shaped function of the j-th term of the i-th input linguistic variable, respectively.

Layer 3: The rule nodes should perform T-norm operation. T-norm has many version. In this paper, we use Minimum operator as T-norm operation. Minimum operator is given by

$$T(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \tag{3}$$

T : T-norm operator

$\mu_A(x) and \mu_B(x)$ represent membership of x in fuzzy set A and B, respectively.

Layer 4: The node in this layer transmit the decision signal out of network. These nodes act as the defuzzyfier. Yager's level set method is used[15].

Modifier: This part measures control performance with the object function and modifies the fuzzy neural network.

2.2 Fuzzy Reasoning Mode

The fuzzy neural network acts as a fuzzy reasoning system. When the fuzzy network receives the input value, nodes in Layer 1 transmit the input value to next layer. Each node in Layer 2 calculates the membership of the received value. Layer 3 performs T-norm operation and saves the calculated value. Layer 4 defuzzifies using the weight value in layer four and the value saved in layer three, then transmits the output value to the plant.

2.3 Fuzzy Rule Learning Mode

In this section, we explain the algorithm for constructing fuzzy control rule . Fuzzy control rule construction is the adjustment of fuzzy singletons in Layer 4. To adjust this value, we use a particular learning algorithm which is similar to competitive learning algorithm.

The present input is transmitted to Layer 1 and Layer 2 calculates the membership values. Each rule node performs T-norm operation using the membership values and save the result. The winner node in Layer 3 is determined based on the saved values. The winner is a node which saved the largest value in the Layer 4. The weight of the winner node is updated. In this procedure, the output value is the weight of the winner node. Thus, the change of the weight of the winner is the change of the output value.

The gradient G of the object function is given by

$$G(t) = \frac{\partial J}{\partial U(t)} = \frac{\partial J}{\partial u(t)_{i^*}} \tag{4}$$

where J is the object function, U(t) is a network output value, $u(t)_i$ is the weight in layer three, i^* represents the winner.

Approximated gradient is given by

$$G(t) \approx \frac{\Delta J}{\Delta U(t)} = \frac{\Delta J}{\Delta u(t)_{i^*}} \tag{5}$$

We update the weight using the gradient. Learning rule is given by

$$u(t+1)_{i^*} = u(t)_{i^*} - \eta(t) M(i^*) G(t) \tag{6}$$

where η(t) is the learning rate and M(i) is value saved in layer four.

2.4 Parameter Learning

After the fuzzy logic rule has been determined, the whole network parameter is established. Then, the network enters the second learning phase to adjust the parameters of the membership function optimally.

For a adjustable parameter in a node, its learning rule is as follows

$$G(t) = \frac{\partial J}{\partial \omega(T)} \tag{7}$$

$$\omega(t+1) = \omega(t) - \eta(t) \cdot G(t) \tag{8}$$

Equation to adjust center mij of the j-th fuzzy set of the i-th linguistic variable is derived as

$$G(t) = \frac{\partial J}{\partial m_{ij}(t)} \approx \frac{\Delta J}{\Delta m_{ij}(t)} \tag{9}$$

$$m_{ij}(t+1) = m_{ij}(t) - \eta(t) \cdot M(i,j) \cdot G(t) \tag{10}$$

where η is the learning late and M(i,j) is the membership function of the j-th fuzzy set of the i-th linguistic variable.

Similarly, the adjustment rule of the width is as follows:

$$G(t) = \frac{\partial J}{\partial \sigma_{ij}} \approx \frac{\Delta J}{\Delta \sigma_{ij}(t)} \tag{11}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta(t) \cdot M(i,j) \cdot G(t) \tag{12}$$

3 Simulation Results

The proposed fuzzy network has been simulated for the inverted pendulum balancing problem[16]. This problem is openly used as an example of inherently unstable and dynamic system.

As shown in Fig. 6, the inverted pendulum balancing problem is the problem of learning how to balance an upright pole. The inverted pendulum is modeled by

$$d\dot{\theta} = \frac{g\sin\theta - \cos\theta\left(\dfrac{F + m_p\dot{\theta}^2\sin\theta}{m_c + m_p}\right)}{\dfrac{4}{3}l - \dfrac{m_p l\cos^2\theta}{m_c + m_p}} \qquad (13)$$

where

F : force(output)

g : gravity

m_c : 1kg, mass of the cart

m_p : 0.1kg, mass of the pole

l : 0.5m, half-pole length

θ : angle of the pole

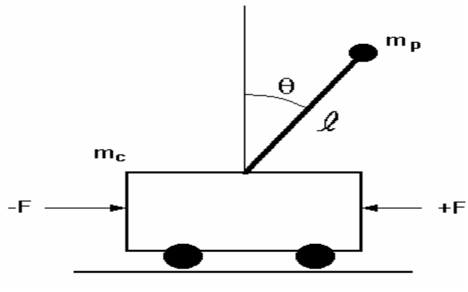


Fig. 2. Inverted pendulum

The constrains on the variable are

θ : -30°~ +30°

dθ : -240 ~ +240 (rpm)

F : -24 ~ +24.

The proposed fuzzy neural network is compared to the conventional fuzzy system. Fig. 3 and Fig. 4 show fuzzy set of the conventional fuzzy system. Table 1 shows the fuzzy rule of the conventional fuzzy system.

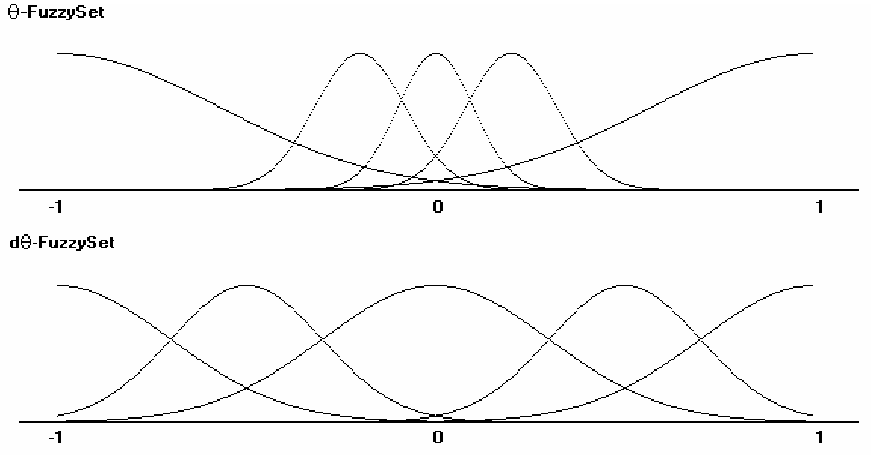


Fig. 3. Fuzzy sets of the input variable in the conventional fuzzy system

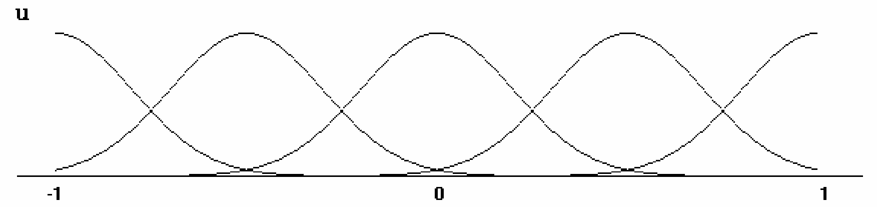


Fig. 4. Fuzzy sets of the output variable in the conventional fuzzy system

Table 1. Fuzzy rules in the conventional fuzzy system

θ dθ	NM	NS	Z	PS	PM
NM	PM	PM	PM	PS	Z
NS	PM	PM	PS	Z	NS
Z	PM	PS	Z	NS	NM
PS	PS	Z	NS	NM	NM
PM	Z	NS	NM	NM	NM

The fuzzy neural network uses the initial fuzzy set shown in Fig. 5.

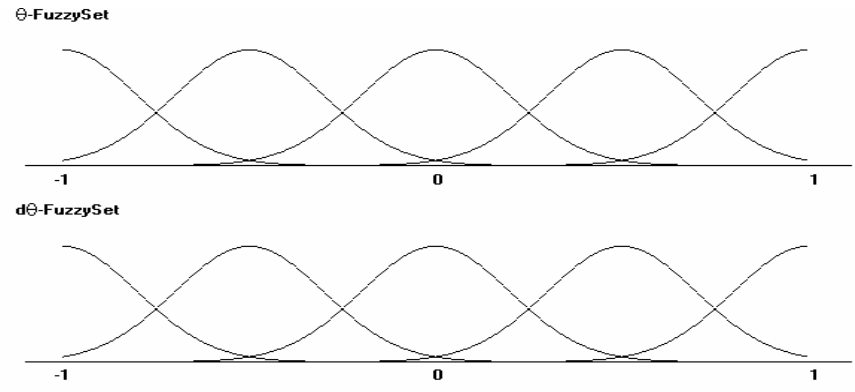


Fig. 5. Initial fuzzy sets of the input variables in the proposed fuzzy network

The O=object function is given by

$$J = \frac{1}{2}\left[30 * (X)^2 + \left(\dot{X} \right)^2 \right] \tag{14}$$

The learning rate in construct fuzzy rule is given by

$$\frac{9}{t+10} + 0.1 \tag{15}$$

Total epoch for constructing fuzzy rules is 100.
Table 2 shows the constructed fuzzy rules.

Table 2. Fuzzy rules generated by the proposed fuzzy network

θ dθ	NM	NS	Z	PS	PM
NM	1	1	0.999	0.774	0.105
NS	1	1	0.999	0.122	-0.565
Z	1	1	0.000	-1	-1
PS	0.558	-0.125	-1	-1	-1
PM	-0.018	-0.701	-1	-1	-1

The learning rate in learning parameter is given by

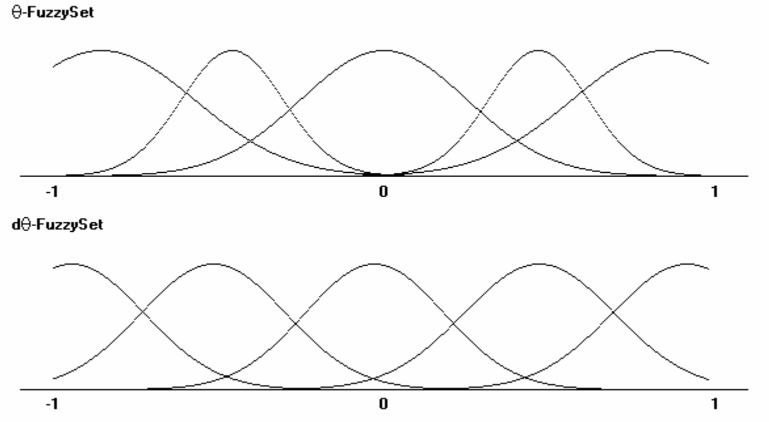


Fig. 6. Fuzzy sets of the input variables generated by the proposed fuzzy network

$$\frac{9}{t+10}+0.1 \tag{16}$$

Total epoch for constructing fuzzy rules is 100.

Fig. 6 shows the tuned fuzzy set.

In this simulation, we can find that the proposed fuzzy neural network is superior to the conventional fuzzy system. The fuzzy neural network is better about 40% in the settling time and about 20% in the overshoot than the conventional system.

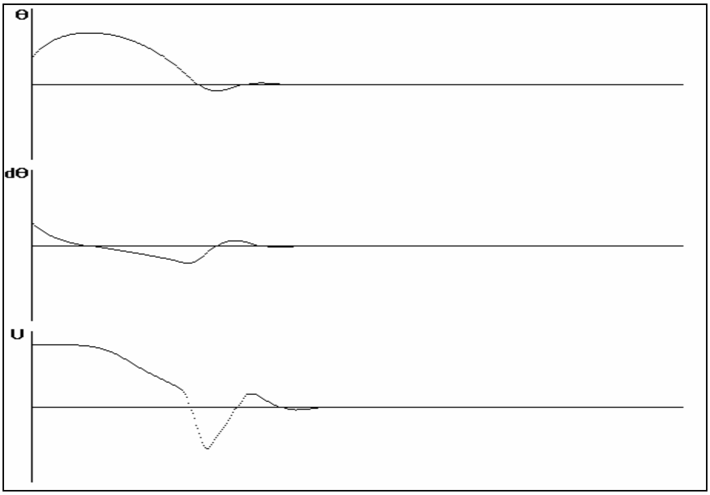


Fig. 7. Experimental results of the conventional fuzzy system (θ=1, dθ=1)

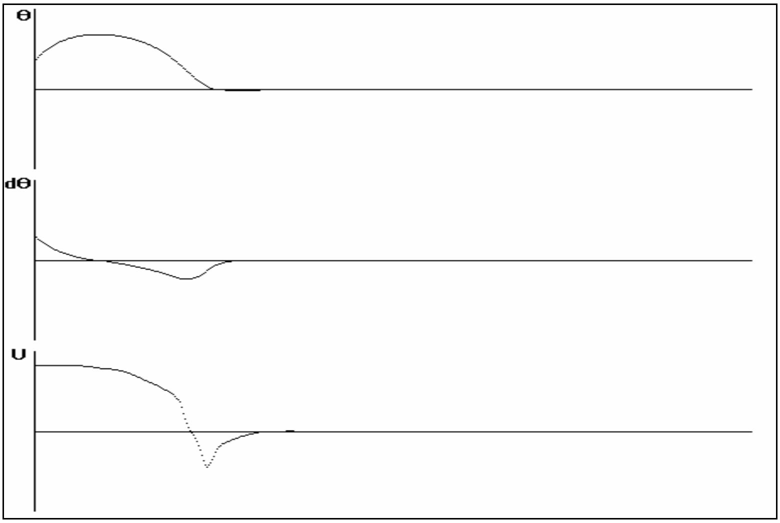


Fig. 8. Experimental results of the proposed fuzzy system (θ=1, dθ=1)

4 Conclusion

This paper described the development of fuzzy network. The proposed fuzzy neural network has three modes. The first one is fuzzy reasoning mode, the second one is fuzzy rule constructing mode, and the third mode is parameter tuning. Using the proposed algorithm, we can construct fuzzy control systems easily. This algorithm needs no input-output relation data for training but needs only the object function. Computer simulation results of the cart-pole balancing problem show that the proposed algorithm is superior to the conventional fuzzy system.

Acknowledgement

This work was supported by 2006 Hongik University Research Fund, the Soongsil University Research Fund, and BK21.

References

1. Mamdani,E.H.: Apllication of Fuzzy Algorithm for Control of Simple Dynamic Plant. Inst. Electri. Eng. **121** (1974) 1569-1588
2. Li ,Y.F. and Lau,C.C.: Developement of fuzzy algorithms for servo systems. IEEE Control Syst. Mag. (1989) 65-72
3. Bernard,J.A.: Use of a rule-based system for process control. IEEE Control Syst. Mag. (1988) 3-13
4. Lo Schiavo,A.,Luciano,A.M.: Powerful and flexible fuzzy algorithm for nonlinear dynamic system identification. IEEE Transactions on Fuzzy Systems. **9**(6) (2001) 828 - 835

5. Ozdemir,D.,Akarun,L.: Fuzzy algorithms for combined quantization and dithering. IEEE Transactions on Image Processingg. **10**(6) (2001) 923 - 931
6. Mousavi,P.,Ward,R.K.,Fels,S.S.,Sameti,M.,Lansdorp,P.M.: Feature analysis and centromere segmentation of human chromosome images using an iterative fuzzy algorithm. IEEE Transactions on Biomedical Engineering. **49**(4) (2002) 363 – 371
7. Roizman,O.,Davydov,V.: Neuro-fuzzy algorithms for power transformers diagnostics. Proceedings of International Conference on Power System Technology **1** (2000) 253 - 258 vol.1
8. Mumolo,E.,Nolich,M.,Menegatti,E.: A genetic-fuzzy algorithm for the articulatory imitation of facial movements during vocalization of a humanoid robot. 2005 5th IEEE-RAS International Conference on Humanoid Robots (2005) 436 - 441
9. Matsushita,S.,Yukita,K.,Goto,Y.,Ichiyanagi,K.,Aoki,H.,Mizutani, Y.: Automatic generation control using GA considering distributed generation. Asia Pacific. IEEE/PES Transmission and Distribution Conference and Exhibition 2002 **3** (2002) 1579 - 1583
10. Rahmoun,A.,Berrani,S.: A genetic-based neuro-fuzzy generator: NEFGEN. ACS/IEEE International Conference on Computer Systems and Applications (2001) 18 - 23
11. Fonseca,E.T.,Vellasco,P.C.Gd.S.,Vellasco,M.M.B.R.,de Andrade,S.A.L.: A neuro-fuzzy system for steel beams patch load prediction. 2005 Fifth International Conference on Hybrid Intelligent Systems (2005)
12. Fan Hehong,Sun Xiaohan,Zhang Mingde: A novel fuzzy evaluation method to evaluate the reliability of FIN. The 8th International Conference on Communication Systems **2** (2002) 1247 - 1251
13. Dmitry,K.,Dmitry,V.: An algorithm for rule generation in fuzzy expert systems. Proceedings of the 17th International Conference on Pattern Recognition **1** (2004) 212 - 215
14. Chong,Gedeon,T.D.,Kovacs,S.,Koczy,L.T.: Sparse fuzzy systems generation and fuzzy rule interpolation: a practical approach. The 12th IEEE International Conference on Fuzzy Systems **1** (2003) 494 - 499
15. Figueiredo,M., Gomide ,F., Rocha,A. and Yager,R.: Comparison of Yager's Level Set Method for Fuzzy Logic Control with Mamdani's and Rasen's Method. IEEE Trans. Fuzzy System **1**(2) (1993)
16. Inoue,H.,Matsuo,K.,Hatase,K.,Kamei,K.,Tsukamoto,M.,Miyasaka,K.: A fuzzy classifier system using hyper-cone membership functions and its application to inverted pendulum contol. 2002 IEEE International Conference on Systems, Man and Cybernetics **6** (2002) 6~9

Decision Making Strategies for Real-Time Train Dispatch and Control

Alexandre Tazoniero[1], Rodrigo Gonçalves[2], and Fernando Gomide[1]

[1] Faculty of Electrical and Computer Engineering, State University of Campinas, Campinas, SP, Brazil
[2] Cflex Ltda, Av. José Bonifácio, 187, Campinas, SP, Brazil
alextaz@dca.fee.unicamp.br, rodrigo@cflex.com.br,
gomide@dca.fee.unicamp.br

Abstract. This paper discusses a decision support system based on fuzzy set techniques and tree search for movement planning in freight rail networks. The aim is to provide a decision making toot to help train dispatchers to plan and control rail traffic in real time. The system uses fuzzy rule-based decision procedures to drive train circulation as closed as possible to reference trajectories. A tree search algorithm is used to improve solution quality. A major characteristic of the system concerns its ability to provide feasible movement plans in real time. Incorporation of practical knowledge and real time response are essential requirements in real world freight railroads traffic management and control.

Keywords: Movement planning, freight train dispatch, fuzzy control, search.

1 Introduction

Routine tasks in freight railroad traffic control require train dispatchers to control train movement over a line, plan the meeting and passing of trains on single-track sections, align switches to control each train movement, perform information gathering and communication with train crew and station or yard workers. Currently, centralized and distributed computer systems are essential to perform both, routine and critical decision-making tasks. The computer system architecture depends on the size and rail network complexity. Computerized dispatching systems evolved from off-line computer aided dispatch systems through on-line integrated information and control systems for dispatch operation management, traffic optimization, and train control. The first solutions for train movement planning systems emerged in the beginning of the seventies, with the development of operations research models [12,11,7], especially linear, mixed programming models and network flow formulations [4,13]. Heuristic schemes such as tabu search, constraints satisfaction, simulated annealing [6], tree search [1], knowledge-based systems [2], have also been developed as an attempt to solve movement planning problems. At the same time, discrete event dynamic systems were introduced as an alternative modeling approach [9,10]. Currently, fuzzy systems, neural networks, genetic algorithms [7] and hybrids [13] are on the research agenda. Basically, a train movement planning and control system is an instance of an information system with local and remote data entry concerning train characteristics, location and status, and record keeping to prepare train sheets and

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 195–204, 2007.
springerlink.com

train performance reports. In addition to information processing tasks, computerized dispatch systems perform the clearing of signals ahead of trains, the clearing of trains out of sidings. Some systems offer tools to suggest meeting and passing decisions. Train dispatchers may accept or override decisions, and align switches. In current systems, the computer can select the meeting and passing plan, control switches and signals under the supervision of a train dispatcher who overrides only in particular circumstances. Integrated train traffic planning, dispatch and train control systems comprise the most complex dispatch systems and, in addition of the previous tasks, perform continuous train operation decisions. Reactive train scheduling consistent with system state, constraints and crew shifts are also part of integrated train control system nowadays.

Currently, most railways worldwide operate under centralized traffic control systems, often with distributed computer architectures, with radio, satellite, and local area networks communication capabilities [14]. Multi-agent systems solutions are under development as an alternative approach for advanced movement plan and train control systems.

This paper addresses a knowledge-based system developed upon fuzzy set techniques [3,15] for train movement planning and control in freight railroads. The system is a decision support tool to help train dispatchers in circulation planning and real time control. The decision-making kernel of the system uses fuzzy a rule-based decision-making procedure to drive train circulation as closed as possible to desired trajectories. The decision procedure uses the rail line state to determine train priority and speed to solve meet and pass conflicts. The approach allows inclusion of operational knowledge and dispatcher experience as a part of the decision-making strategy. Knowledge and real time response are essential for traffic management and control in practice. A decision-making strategy based on a tree search algorithm [8] is also presented. The tree search algorithm gives the system a global view of the problem during decision-making process. The paper discusses train movement planning, dispatch, and control system structure and decision procedures. Comparison with optimal circulation decisions provided by a mixed linear programming model and a genetic algorithm is included. A case study conducted for a high traffic corridor of a major Brazilian railway is also reported. The paper concludes suggesting items for further development.

2 Dispatch Planning System Structure

Several modules compose the train movement planning, dispatch, and control system. The main module uses a discrete event model and rail network simulator [10] to emulate train movements considering operational and practical constraints, line blockage prevention, and transportation goals. Examples of operational constraints include: trains can not move in the same single track sections at the same time; trains must be re-fuelled at specific fueling stations; train meeting/ passing must occur at sidings or yards; time windows for rail maintenance and crew work must be given to maintenance crew. System structure with its main modules is shown in Fig. 1. The man machine interface allows dispatchers to modify and custom movement plans considering all information, including those not automatically available. The global optimization module uses a mixed linear programming model [13] and genetic fuzzy algorithms [7] to develop *optimal* trajectory references. During simulation, conflicting situations in which trains compete to use the same track at the same time often occur. In these cases the system must decide which train has the preference to occupy the track and which must stop at a siding. At this point the simulator kernel calls the decision-making strategy.

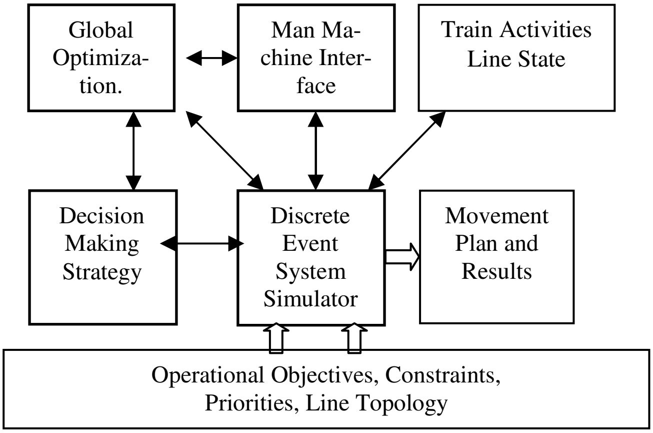

Global Optimization.
Man Machine Interface
Train Activities Line State
Decision Making Strategy
Discrete Event System Simulator
Movement Plan and Results
Operational Objectives, Constraints, Priorities, Line Topology

Fig. 1. System structure

An important characteristic of rail systems is the frequent and unexpected occurrence of events such as train faults, line outages, crew problems, load and unload delays. This means that in actual circumstances the concept of optimal decisions should be viewed carefully. Whenever an *optimal* solution is found, the state of the railroad system often is different from the one originally assumed and the current solution may be no longer valid. For this reason, real time, knowledge based solutions are often preferred in practice because it responds faster and represents realistically trade-offs train dispatchers must do during train movement.

3 Fuzzy Decision Making Strategy

The decision-making strategy heuristics is encoded in a fuzzy rule base to classify train states and movements and produce the most appropriate decision. Decision rules are includes the ones used by train dispatchers in practice. A typical rule to decide train priority in conflicting situations is: if a train with a large delay meets another train that is ahead of its desired time reference, then the delayed train has higher preference to occupy the track. Fuzzy set techniques are primarily used to analyze and decide train movement situations. Train delay and advance are defined assuming a reference timeline with lower and upper boundary time lines as depicted in Fig. 2a. The reference timeline may be seen as a reference for the trajectory of a train along the railroad section in a time horizon. The reference trajectory can either be constructed by the dispatcher or obtained by a global optimization models. We note that, as shown in Fig. 2a, whenever a train is within Area 1 it is advanced and when within Area 2 it is delayed. Delays and advances are granulated using fuzzy sets defined in normalized universes as in Fig. 2b where ND = no delay, SD = small delay, BD = big delay, and SA = small advance, BA = big advance. Whenever two trains compete for a track, their current states are classified using the fuzzy sets and corresponding rule bases to get a decision. During each decision-making decision cycle, trains delays or advance are qualified using linguistic labels shown in Fig. 2b.

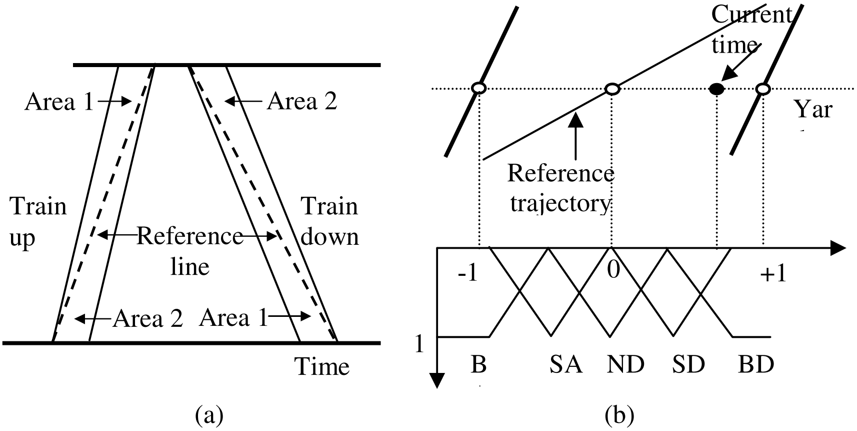


Fig. 2. (a) Reference trajectory and bounds. (b) Rule base.

A decision table mapping the fuzzy rule base, Table 1, is used to determine the relative priorities between trains. In the cases which competing trains are at equivalent states or situations the final decision depend on others factors such as train regularity, number of stops, default priorities, etc. When trains remain in similar states, the default is to assign the track to the train that traverse the conflicting yard first.

In a constant speed model, whenever a train is delayed from its reference trajectory there is no chance to reduce this delay to get closer to the reference. Variable speed models allow trains to reach references increasing or reducing their velocities.

Practical rules can also be used to determine the speed: if a train with a large delay has the preference to occupy a track, then it can increase its speed. The speed is chosen within maximum and minimum bounds. Speed is granulated using fuzzy sets defined in normalized universes as in Fig. 3 where VL = very low speed, LS = low speed, MS = medium speed, and HS = high speed, VH = very high speed.

Table 1. Relative preference decision table

Train A / Train B	BA	SA	ND	SD	BD
BA	A	A	A	A	A
SA	B	B	A	A	A
ND	B	B	A	A	A
SD	B	B	B	B	A
BD	B	B	B	B	A

Based on characterization of train delay using fuzzy sets, Fig. 2b, and assuming train speeds with linguistic values as shown in Fig 3, a rule base is constructed to determine the speed with which a train should move in a track section. The rule base is summarized in Table 2.

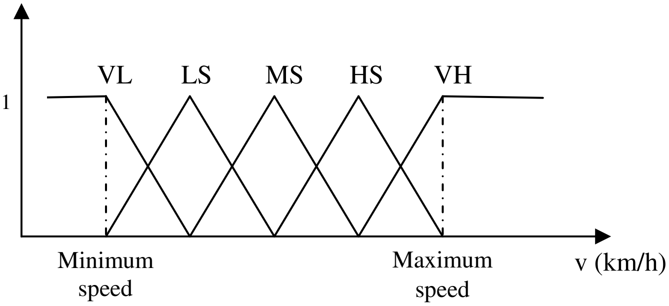


Fig. 3. Characterization of train speed

When two or more trains compete to use the same track at the same time, the preferred train to occupy the track is determined using the rule base of Table 1 and the speed with which the preferred train should cross the track is found using the rule base of Table 2.

It is important to note that the fuzzy decision-making procedure uses local information only, but references provided by optimization algorithms or train dispatchers give a global view to the local decision-making procedure. This feature allows fast response and considers information on state of the system as a whole. An algorithm that develops a global decision is presented in next section.

Table 2. Rule base for train speed

Train temporal state	Train speed	Rule
BA	VL	If train is BA then speed is VL
SA	LS	If train is SA then speed is LS
ND	MS	If train is ND then speed is MS
SD	HS	If train is SD then speed is HS
BD	VH	If train is BD then speed is VH

4 Tree Search Decision-Making Strategy

Combinatorial and discrete optimization problems, such as train movement planning and dispatch problems, consider a set of decision variables that must be assigned to a value selected from a set of discrete alternatives [8]. A decision problem can be represented as a tree: each leaf is a solution, each node is a decision variable, and each branch a value assigned to a variable.

In the train movement planning and dispatch problem, a node is an occupancy conflicting situation and the decision variable concerns which train is the preferred train to occupy the track. Thus, each branch represents a decision to select a train among the competing trains. Fig. 4 shows an example of a small instance of a rail line with two trains, train 1 and train 2, three single line tracks, and 4 sidings. The root node represents the time instant when train 1, at the siding s1, competes with train 2, at the siding s3, to occupy track s2. There are two alternatives to resolve this conflict. The first gives preference to train 1, train 1 moves to s2, and train 2 remains stopped at s3. This decision generates node 1. The second alternative assigns preference to train 2 which moves to s2 while train 1 remains stopped. This decision generates node 2. In this simple example, node 1 and node 2 are leafs of the tree, and are candidate solutions.

Search starts at the root node, Fig. 4, and proceeds choosing the next node. One way to generate both nodes and choose the node which best fits the decision objectives of the problem. This is simple for small problem instances, but infeasible in practice because the number of conflicts can be very high. An alternative is a look-ahead strategy to analyze the impact of the actual conflict in the future train meetings and passing in the planning horizon. Due to the time restrictions of real world environments, it is not possible to expand all nodes and choose the best node. Therefore a mechanism to estimate the cost of the future meetings and passing simulating trains movements using a discrete event simulator. Simulation allows the tree search strategy to make a decision based on information of what has happened up to the conflict and on what will happen for each decision alternative. Clearly, during simulation new conflicts will occur and they must also be solved. A fast way to solve conflicts is to use a greedy decision strategy, such as choosing among the conflicting trains the one that traverses the conflicting track first.

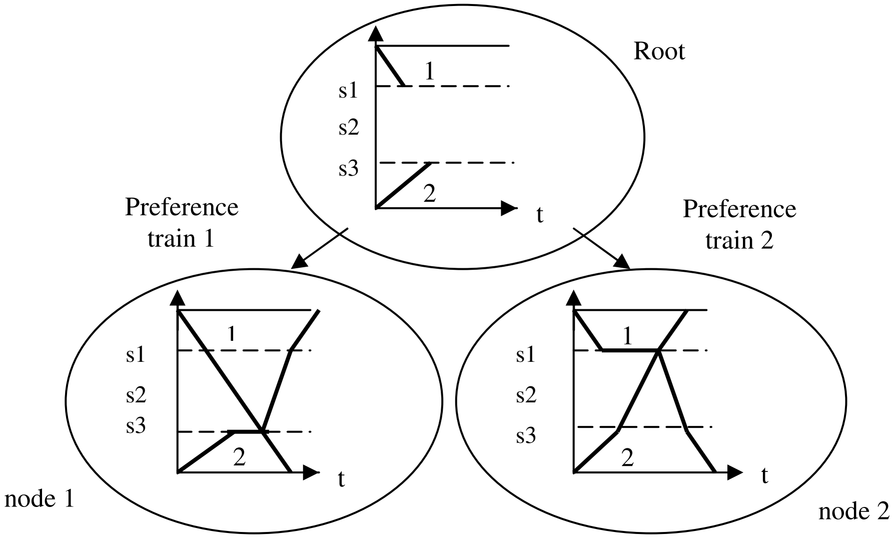


Fig. 4. Search tree representation of the train movement problem

Using traditional notation of tree search methods [9], we may set the value of a node *n* using a node evaluation function f(*n*) of the form:

$$f(n) = g(n) + h(n) \tag{1}$$

In (1) g(*n*) is the value from the initial node to node *n* and h(*n*) is the estimated value from node *n* to a solution node of the problem. In our case, h(*n*) is the value given by simulation of node *n* for the planning horizon. Depending on the horizon considered, simulation may find a solution or not. Whenever a node (trains conflict) is found, all successors nodes are generated and f(.) is calculated for each of them. The node selected (preferred train) is the one with the lowest f(.) value. The cost of a node must be calculated considering the transportation aims of the problem. In the application example and scenarios addressed in next section, the aim is to reduce the total delay due to crossing and passing, that is, the sum of the delays of all trains within the planning horizon.

5 Simulation Results

In this section we summarize the results obtained when using the algorithms of the previous sections and compare their solutions with the ones provided by a mixed linear programming model and a fuzzy genetic algorithm described in [13] and [7], respectively. Due the considerable processing time needed to obtain exact solutions using the mixed linear programming models, only small size problems are considered for comparison purposes. As it is well know, train dispatch problems of the sort addressed in this paper are np-hard. We consider two rail lines examples. Rail line 1, has five single-track segments and four double track segments for crossing and passing. Rail line 2 has eight single-track segments and seven double track segments for crossing and passing.

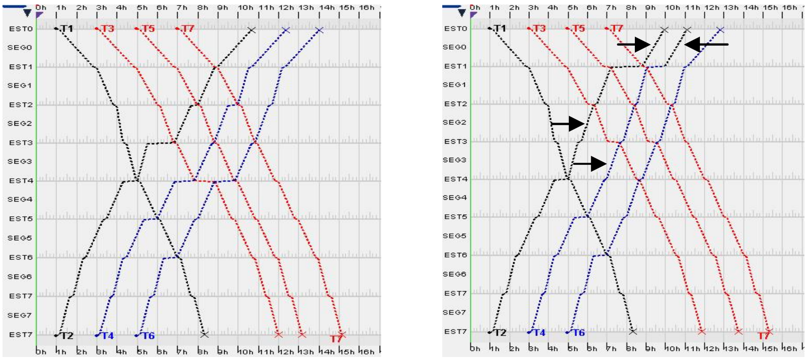
Using the two rail lines above, 30 instances were considered. The instances were generated varying the number of trains from three to seven, and varying the time interval between departures from two to four hours. For each case both, the solution provided by the mixed linear mathematical model (MPM) and the solution obtained by the genetic fuzzy algorithm (GFA), are adopted as the reference trajectory used by the fuzzy decision-making algorithm. Self-centered references, considering the individual train activities programs (TAP), were also adopted as reference trajectories. Fuzzy decision-making considers speed limits ± 30% of the train average speed. The tree search algorithm (TSDM) considers a time horizon of six hours to estimate the cost function, h(*n*). In all simulations we assume that the main objective is to reduce the total train delay. Table 3 summarizes simulation results using rail line 1 with two hours interval between train departures and a total of seven trains. Its important to note that the MPM, the GFA, and the TSDM assumes that the time required for a train to cross each segment is the average, whereas the FDM strategy can vary the speed of a train.

Table 3. Simulation Results

	MPM	GFA	FDM_{MPM}	FDM_{FGA}	FDM_{TAP}	TSDM
Delay (min)	498.23	518.17	498.23	518.17	580.3	498.23
Processing time (ms)	9849672	36450	47	94	78	1100

GFA: Genetic Fuzzy Algorithm, MPM: Mathematic Programming Model, FDM_{GFA}: Fuzzy Decision Making with GFA reference, FDM_{MPM}: Fuzzy Decision Making with MPM reference, FDM_{TAP}: Fuzzy decision Making with TAP (Trains Activities Program) reference TSDM: Tree Search Decision Making.

Fig. 5a shows the optimal solution provided by MPM and by the TSDM. As it can be seen in Table 3, the solution developed by the FDM strategy using the solution provided by the mathematical model is the same as the reference solution. With the GFA reference the solution was also the same as the FGA solution. Using the TAP as reference produces worse solution because the TAP reference usually is not feasible once it does not reflect trains conflicts. Fig. 5b shows the results when TAP is set as a reference for the fuzzy decision-making algorithm. The arrows indicate where trains speed increases. The MPM and GFA solutions are feasible and the FDM strategy can follow them identically. The TSDM does not use any reference, but obtains the optimal solution as well. Looking at the processing times we notice that the MPM spent above three hours to find the optimal solution while the GFA needed 36 seconds only. In contrast, the FDM spent less than 100ms to find a solution while TSDM needed about a second, a value considered satisfactory for real-time time movement planning.



(a) (b)

Fig. 5. (a) Optimal solution given by the mathematical programming model. (b) Fuzzy decision-making solution using TAP solution as reference.

Table 4 summarizes the optimality gaps considering the total delays of 30 problem instances. The fuzzy decision making strategy follows the reference identically when it is provided by the mathematical model or by the fuzzy genetic algorithm. With TAP as reference the total delay is 23% higher than the optimal solution. The TSDM achieves a total mean delay 5.3% above the optimal.

Table 4. Optimality gap of the total delay

	FDM_{PLMI}	GFA	FDM_{GFA}	FDM_{TAP}	TSDM
Mean (%)	0	3.5	3.5	23	5.3
Worst case (%)	0	40	40	253	43

An instance of train circulation planning using actual data from a major Brazilian railway corridor with 50 trains and 40 passing yards was also considered as a test case. It is impractical to get exact solutions within acceptable time limits using mixed linear mathematical programming model to solve this instance. Table 5 summarizes the results. The genetic algorithm required 12 minutes to solve it. The knowledge-based system, with the fuzzy decision-making algorithm described in this paper, needed 453 ms to produce an acceptable schedule. The best total delay result was obtained by the tree search strategy within 42 seconds.

Table 5. Simulation results for an actual railroad corridor instance

	GFA	FDM_{GFA}	FDM_{TAP}	TSDM
Delay (min)	3798	3798	4802	3365
Processing time (ms)	720000	422	453	42515

6 Conclusion

The knowledge-based strategies suggested in this paper have shown to provide an effective mechanism to develop satisfactory solutions in real time train movement planning, and traffic management. This feature is of utmost importance in train dispatch and control systems, where the unpredictably and intractability of the problem makes conventional solutions ineffective. Moreover, since the system uses a detailed discrete event model of the rail network, the solutions obtained are often closer to practice than conventional mathematical programming models.

Future work will focus on the fuzzy decision-algorithm to improve performance when using self-centered references. Construction of estimation functions h(n) for the tree search strategy is also under investigation.

References

1. Cherniavsky AL (1972) A Program for Timetable Compilation by a Look-Ahead Method. Artificial Intelligence 3: 61-67
2. Chiang H et al. (1998) Knowledge-based system for railway scheduling. Data & Knowledge Engineering 27: 289-312
3. Gomide F, Pedrycz W (1998) An Introduction to Fuzzy Sets: Analysis and Design. MIT Press, Cambridge, USA.

4. Higgins A et al. (1996) Optimal Scheduling of Trains on a Single Line Track. Transportation Research 30: 147-161
5. Higgins A et al. (1997) Heuristic Techniques for Single Line Train Scheduling. Journal of Heuristics 3: 43-62
6. Issai TM, Singh, GM (2001) Hybrid Applications of Constraint Satisfaction and Meta-Heuristics to Railway Timetabling: A Comparative Study. IEEE Transactions on System, Man, and Cybernetics 31: 87–95
7. Mota F et al. (2005) Genetic Algorithm, Fuzzy Clustering and Discrete Event Systems: An Application in Scheduling. In: Proceedings of the I Workshop on Genetic Fuzzy Systems, Granada, Spain, pp 83-88.
8. Pearl J (1984) Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley Publishing Company, Los Angeles.
9. Petersen E, Taylor A (1982) A Structured Model for Rail Line Simulation and Optimization. Transportation Science 16: 192-205.
10. Rondon M (2000) Modelagem Estruturada e Sistemas Inteligentes: Uma Aplicação a Sistemas de Transporte Ferroviário. MSc. thesis, Universidade Estadual de Campinas
11. Sauder RL, Easterman M (1983) Computer-Aided Train Dispatching Decision Support through Optimization. Interfaces 13: 24-37.
12. Szpigel B (1972) Optimal Train Scheduling on Single Track Railway. Operations Research 20: 343-352.
13. Vale A et al. (2005) Fuzzy Optimization Model for Train Dispatch Systems. In: Eleventh IFSA World Congress. Beijing, China, pp 1788-1793.
14. Vernazza G, Zunino R (1990) A distributed intelligence methodology for railway traffic control. IEEE Transactions on Vehicular Technology 39:263-270.
15. Yager R, Zadeh L (1992) An Introduction to Fuzzy Logic Applications in Intelligent Systems. Kluwer Academic Publishers, Boston, USA.

Soft Margin Training for Associative Memories Implemented by Recurrent Neural Networks

Jose A. Ruz-Hernandez[1], Edgar N. Sanchez[2], and Dionisio A. Suarez[3]

[1] Universidad Autonoma del Carmen, Av. 56 # 4 X Av. Concordia, Col. Aviación, C.P. 24180,
Cd. del Carmen, Campeche, Mexico
jruz@pampano.unacar.mx

[2] CINVESTAV, Unidad Guadalajara, Apartado Postal 31-430, Plaza La Luna, C.P. 45091,
Guadalajara, Jalisco, MEXICO, on sabbatical leave at CUCEI, Universidad de Guadalajara
sanchez@gdl.cinvestav.mx

[3] Instituto de Investigaciones Electricas, Calle Reforma # 113, Col. Palmira, C.P. 62490,
Cuernavaca, Morelos, Mexico
suarez@iie.org.mx

Abstract. In this paper, the authors discuss a new synthesis approach to train associative memories, based on recurrent neural networks (RNNs). They propose to use soft margin training for associative memories, which is efficient when training patterns are not all linearly separable. On the basis of the soft margin algorithm used to train support vector machines (SVMs), the new algorithm is developed in order to improve the obtained results via optimal training algorithm also innovated by the authors, which are not fully satisfactory due to that some times the training patterns are not all linearly separable. This new algorithm is used for the synthesis of an associative memory considering the design based on a RNN with the connection matrix having upper bounds on the diagonal elements to reduce the total number of spurious memory. The scheme is evaluated via a full scale simulator to diagnose the main faults occurred in fossil electric power plants.

1 Introduction

The implementation of associative memories via RNNs is discussed in [1], where a synthesis approach is developed based on the perceptron training algorithm. The goal of associative memories is to store a set of desired patterns as stable memories such that a stored pattern can be retrieved when the input pattern (or the initial pattern) contains sufficient information about that stored pattern. In practice the desired memory patterns are usually represented by bipolar vectors (or binary vectors). There are several well-known methods available in the literature, which solve the synthesis problem of RNNs for associative memories, including the *outer product method*, the *projection learning rule* and the *eigenstructure method*, [2].

Due to their high generalization performance, SVMs have attracted great attention for pattern recognition, machine learning, neural networks and so on, [3]. Learning of a SVM leads to a quadratic programming (QP) problem, which can be solved by many techniques [4]. Furthermore, the relation existing between SVMs and the design of associative memories based on the generalized brain-state-in-a-box (GSB) neural model is formulated in [5].

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 205–214, 2007.
springerlink.com

On the basis of the optimal hyperplane algorithm used to train SVMs, an optimal training algorithm for associative memories has been developed and applied by the authors in [6] and [7]. The proof related to the convergence properties for this algorithm when patterns are linearly separable, and the corresponding proof for the design of an associative memory via RNNs using constraints in the diagonal elements of connection matrix can be reviewed in [6] and [8].

This paper propose a new soft margin training for associative memories implemented by RNNs. On the basis of the soft margin algorithm used to train SVMs as described in [3], the new algorithm is developed in order to improve the obtained results via optimal training algorithm when the training patterns are not all linearly separable.

2 Preliminaries

This section introduces useful preliminaries about associative memories implemented by RNNs, the perceptron training algorithm and a synthesis for RNNs based on this algorithm, which is proposed in [1]. The class of RNNs considered is described by equations of the form

$$\begin{aligned} \frac{dx}{dt} &= -Ax + T\mathrm{sat}(x) + I \\ y &= \mathrm{sat}(x) \end{aligned} , \tag{1}$$

where x is the state vector, $y \in D^n = \{x \in R^n \mid -1 \le x_i \le 1\}$ is the output vector, $A = \mathrm{diag}\,[a_1, a_2, \ldots, a_n]$ with $a_i > 0$ for $i = 1, 2, ..., n$, $T = [T_{ij}] \in R^{nxn}$ is the connection matrix, $I = [I_1, I_2, \ldots, I_\mathrm{n}]^T$ is a bias vector, and $\mathrm{sat}(x) = [\mathrm{sat}(x_1), \ldots, \mathrm{sat}(x_n)]^T$ represents the activation function, where

$$\mathrm{sat}(x_i) = \begin{cases} 1 & x_i > 1 \\ x_i & -1 \le x_i \le 1 \\ -1 & x_i < -1 \end{cases} . \tag{2}$$

It is assumed that the initial states of (1) satisfy $| x_i(0) | \le 1$ for $i = 1, 2, ..., n$. System (1) is a variant of the analog Hopfield model with activation function sat(•).

2.1 Synthesis Problem for Associative Memories Implemented by RNNs

For the sake of completeness, the following results are taken from [1] and included in this section. A vector α will be called a (*stable*) *memory vector* (or simply, *a memory*) of system (1) if $\alpha = \mathrm{sat}(\beta)$ and if β is an *asymptotically stable* equilibrium point of system (1). In the following lemma, B^n is defined as a set of n-dimensional *bipolar vectors* $B^n = \{x \in R^n \mid x_i = 1 \text{ o } -1, i = 1, 2, ..., n\}$. For $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T \in B^n$ define $\mathrm{C}(\alpha) = \{x \in R^n \mid x_i\alpha_i > 1, i = 1, 2, ..., n\}$.

Lemma 2.1 If $\alpha \in B^\mathrm{n}$ and if

$$\beta = A^{-1}(T\alpha + I) \in C(\alpha) , \tag{3}$$

then (α, β) is a pair of stable memory vector and an asymptotically stable equilibrium point of (1). The proof of this result can be reviewed in [1].

The following synthesis problem concerns the design of (1) for associative memories.

Synthesis Problem: Given m vectors $\alpha^1, \alpha^2, ..., \alpha^m$ in the set of n-dimensional bipolar vectors, B^n, choose $\{A, T, I\}$ in such a manner that:

1. $\alpha^1, \alpha^2, ..., \alpha^m$ are stable memory vectors of system (1);
2. the total number of spurious memory vectors (i.e., memory vectors which are not desired) is as small as possible, and the domain (or basin) of attraction of each desired memory vectors is as large as possible.

Item 1) of the synthesis problem can be guaranteed by choosing the $\{A, T, I\}$ such that every α^i satisfies condition 3 of Lemma 2.1. Item 2) can be partly ensured by constraining the diagonal elements of the connection matrix.

In order to solve the synthesis problem, one needs to determine A, T and I from (3) with $\alpha = \alpha^k$, $k = 1, 2,..., m$.

Condition given in (3) can be equivalently written as

$$\begin{cases} T_i\alpha^k + I_i > a_i & \text{if } \alpha_i^k = 1 \\ T_i\alpha^k + I_i < -a_i & \text{if } \alpha_i^k = -1 \end{cases}, \tag{4}$$

for $k = 1, 2,..., m$ and $i = 1, 2,..., n$ where T_i represents the ith row of T, I_i denotes the ith element of I, a_i is the i-th diagonal element of A, and α_i^k is the i-th entry of α^k.

3 New Approach: Soft Margin Training for Associative Memories

This section contains our principal contribution. First, we describe soft margin algorithm used for SVMs when training patterns are not all linearly separable. We propose a new soft margin training for associative memories implemented by RNN (1).

3.1 Soft Margin Algorithm

Consider the case where the training data can not be separated without error by an SVM. In this case one may want to separate the training set with a minimal number of errors. To express this formally let us introduce some non-negative variables $\xi_i \geq 0$, $i = 1, \ldots, l$.

We can now minimize the functional

$$\phi(\xi) = \sum_{i=1}^{l} \xi_i^{\sigma} \tag{5}$$

for small $\sigma > 0$, subject to the constraints

$$y_i(W \bullet X_i + b) \geq 1 - \xi_i, \qquad i = 1, ..., l. \tag{6}$$

$$\xi_i \geq 0, \qquad i = 1,...,l. \tag{7}$$

For sufficiently small σ the functional (5) describes the number of the trainin errors. Minimizing (3) one finds some minimal subset of training errors

$$(y_{i1}, X_{i1}), \ldots, (y_{ik}, X_{ik}). \tag{8}$$

If these data are excluded from the training set one can separate the remaining part of the training set without errors. To separate the remaining part of the training data one can construct an optimal separating hyperplane. To construct a soft margin hyperplane we maximize the functional

$$\Phi = \frac{1}{2} W \bullet W + C(\sum_{i=1}^{l} \xi_i^{\sigma}) \tag{9}$$

subject to constraints (6) and (7), where C is a constant. The solution to the optimization problem under the constraint (6) is given by the saddle point of the Lagrangian, as obtained in [3]

$$L(W, \xi, b, \Lambda, R) = \frac{1}{2} W \bullet W + C(\sum_{i=1}^{l} \xi_i^{\xi}) - \sum_{i=1}^{l} \alpha_i [y_i (x_i \bullet W + b) - 1 + \xi_i^{\xi}] - \sum_{i=1}^{l} r_i \xi_i^{\xi} \tag{10}$$

where R^T=[r_1, $r_{2,}$ $r_{3,}$ $r_{4,}$ r_5] enforces the constraint (6). In [3] is described that vector W can be written as a linear combination of support vectors when optimal hyperplane algorithm is used for training. To find the vector Λ^T=[λ_1, …, λ_l] one has to solve the dual quadratic programming problem of maximizing

$$W(\Lambda, \delta) = \Lambda^T Q - \frac{1}{2}[\Lambda^T D \Lambda + \frac{\delta^2}{C}] \tag{11}$$

subject to the constraints

$$\Lambda^T Y = 0 \tag{12}$$

$$\delta \geq 0 \tag{13}$$

$$0 \leq \Lambda \leq \delta Q \tag{14}$$

where Q is an l-dimensional unit vector, Λ=[λ_1, λ_2, …, λ_l] contains Lagrange multipliers, Y contains the entries, and D is a symmetric matrix are the same elements as used in the optimization problem for constructing an optimal hyperplane, δ is a scalar, and (14) describes coordinate-wise inequalities.

Note that (14) implies that the smallest admissible value δ in the functional (11) is

$$\delta = \lambda_{\max} = \max\ [\lambda_1, \lambda_{2,} \ldots, \lambda_l] \tag{15}$$

Therefore to find a soft margin classifier one has to find a vector Λ that maximizes

$$W(\Lambda) = \Lambda^T Q - \frac{1}{2}[\Lambda^T D A + \frac{\lambda_{\max}^2}{C}] \tag{16}$$

under the constraints Λ ≥ 0 and (14). This problem differs from the problem of constructing an optimal margin classifier only by the additional term with $\lambda_{\max}$ in the functional (11). Due to this term the solution to the problem of constructing the soft margin classifier is unique and exist for any data set.

3.2 New Synthesis Approach

Considering the soft margin algorithm, we propose the soft margin training for associative memories implemented by RNNs which is described as follows.

Synthesis Algorithm 3.1: Given *m* training patterns α^k, $k = 1, 2,\ldots, m$ which are known to belong to class X_1 (corresponding to $Z = 1$) or X_2 (corresponding to $Z = -1$), the weight vector *W* can be determined by means of the following algorithm.

1. Start out by solving the quadratic programming problem given by

$$F(\Lambda^{\mathrm{i}}) = (\Lambda^{\mathrm{i}})^T Q - \frac{1}{2}[(\Lambda^{\mathrm{i}})^T D\Lambda^{\mathrm{i}} + \frac{\lambda_{\max}^2}{C}], \tag{17}$$

under the constraints

$$(\Lambda^{\mathrm{i}})^T Y^i = 0, \text{ where } Y^i = \begin{bmatrix}\alpha_i^1 & \alpha_i^2 & \cdots & \alpha_i^m\end{bmatrix},$$

$$\delta \geq 0,$$

$$0 \leq \Lambda \leq \delta Q$$

to obtain $\Lambda^{\mathrm{i}} = \left[\lambda_1^i, \lambda_2^i, \ldots, \lambda_n^i\right]$.

2. Compute the weight vector

$$W^i = \sum_{j=1}^{m} \lambda_j \alpha_j^k \alpha^{-k} = \left[w_1^i, w_2^i, \ldots, w_n^i, w_{n+1}^i\right], \tag{18}$$

$i = 1, 2, \ldots, n$, such that

$$\begin{aligned} W^i \bullet \alpha^{-k} + b \geq 1 - \xi_i \quad \text{if} \quad \alpha_i^k = 1, \\ W^i \bullet \alpha^{-k} + b \leq -1 + \xi_i \quad \text{if} \quad \alpha_i^k = -1, \end{aligned} \tag{19}$$

and for $k = 1, 2, \ldots, m$ where

$$\alpha^{-k} = \begin{bmatrix}\alpha^k \\ \ldots \\ 1\end{bmatrix}. \tag{20}$$

Choose $A = \operatorname{diag}[a_1, a_2, \ldots, a_n]$ with $a_i > 0$. For $i, j = 1,2,\ldots,n$ choose $T_{ij} = w_j^i$ if $i \neq j$, $T_{ij} = w_j^i + a_i\mu_i$ -1 if i=j, with $\mu_i > 1$ e $I_i = w_{n+1}^i + b_i$.

4 Application to Fault Diagnosis in Fossil Electric Power Plants

Due to space limitations, in this section we only describe some results for fault diagnosis in fossil electric power plants. A full description about this application can be reviewed in [7] and [8]. Fig. 1 illustrates the scheme for fault diagnosis and its components are explained below.

The first component is based on comparison between the measurements coming from the plant and the predicted values generated by a neural network predictor. The predictor is based on neural network models, which are trained using healthy data from the plant. The differences between these two values, named as residuals, constitute a good indicator for fault detection. These residuals are calculated as

$$r_i(k) = x_i(k) - \hat{x}_i(k)\ ,\ i = 1, 2, \ldots, n. \tag{21}$$

where $x_i(k)$ are the plant measures and $\hat{x}_i(k)$ are the predictions. In absence of faults, the residuals are only due to noise and disturbance. When a fault occurs in the system, the residuals deviate from zero in characteristic ways.

The scheme is applied to six faults: *waterwall tube breaking*, *superheating tube breaking*, *dirty regenerative preheater*, *superheated steam temperature control fault*, *velocity variator of feedwater pumps operating at maximum and blocked fossil oil valve* named as *fault* 1 to *fault* 6, respectively. However, we only present the results for *fault* 1 due to space limitations. Fault data base is adquired with a full scale simulator for 75% of initial load power (225 MW), 15 % of severity fault, 2 minutes for inception and 8 minutes of simulation time. Fig. 2 illustrates that the residuals are closed to zero before fault inception; after this interval, residuals deviate from zero in different ways.

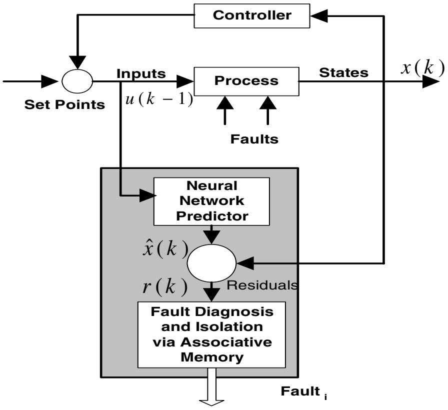


Fig. 1. Scheme for Fault Diagnosis

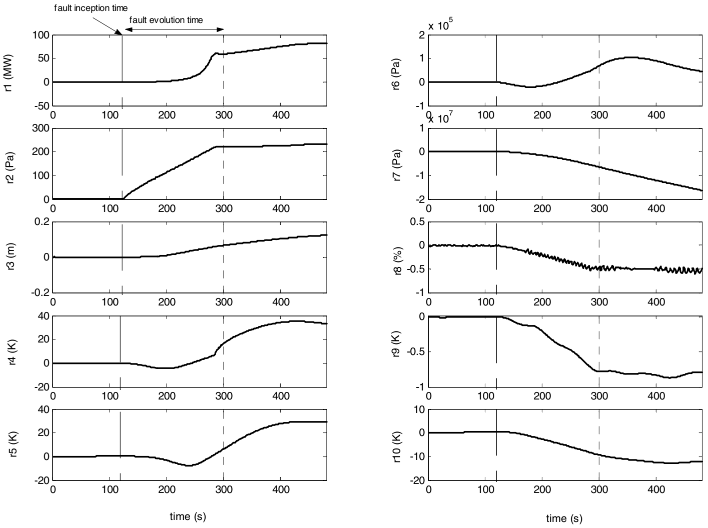


Fig. 2. Residuals for *fault* 1

For the second component, residuals are encoded in bipolar or binary vectors using thresholds to obtain fault patterns to determine the fault. The previous stage generates a residual vector with ten elements. Each residual is evaluated by a detection threshold (τ_i). Detection thresholds are displayed in Table 1. This evaluation provides a set of encoded residuals as bipolar vectors [s$_1$(*k*), s$_2$(*k*), …, s$_{10}$(*k*)]^T where

$$s_i(k) = \begin{cases} -1 \text{ if } |r_i| < \tau_i \\ 1 \text{ if } |r_i| \geq \tau_i \end{cases}, \; i = 1, 2, ..., 10. \tag{22}$$

Table 1. Detection thresholds

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
τ_i	25 MW	30 Pa	0.022 m	4 k	10 K	20000 Pa	42000 Pa	0.85 %	4 K	10 K

Fault patterns for all the six faults previously mentioned are contained in Table 2, where *fault* 0 pattern is included to denote a normal operating condition.

The soft margin training algorithm is encoded in MATLAB. The number of neurons is *n*=10 (fault pattern length) and the patterns are *m*=7 (number of fault patterns). The Lagrange multipliers matrix *LM*= [Λ^1, Λ^2 …,Λ^n], the weight matrix *WM*=[W^1, W^2, …,W^{n+1}] and the bias vector *BV*=[b^1, b^2, …, b^n] are obtained as in (23), (24) and (25). The matrices *A*, *T* and *I* are calculated as in (26), (27) and (28). It is worth to mention that all diagonal elements in matrix *T* satisfies the optimal constrainst $T_{ii} \leq a_i$ in order to reduce the total number of spurious memories as described in [1] and [6]. New soft margin training algorithm improves the obtained results via optimal training developed by the authors where $T_{88} = 2.59 > a_8$ and $w_{88} = 1.59 > 1$ does not satisfy the corresponding optimal constraint as obtained in [7 p. 60]. Using soft margin training, $T_{88} = a_8 = 1$ and $w_{88} = 0.7 < 1$ are obtained. The associative memory is evaluated with these matrices fixed using encoded residuals as input bipolar vectors. Fig. 3 illustrates how α^1 pattern, is retrieved when *fault* 1 is occurring.

Table 2. Fault patterns to store in associative memory

α^0	α^1	α^2	α^3	α^4	α^5	α^6
-1	1	-1	-1	-1	-1	1
-1	1	-1	-1	-1	-1	-1
-1	1	1	1	1	1	1
-1	1	-1	1	-1	-1	-1
-1	1	1	1	-1	-1	1
-1	1	-1	1	-1	1	1
-1	1	1	1	1	-1	1
-1	-1	-1	-1	-1	-1	1
-1	1	-1	-1	-1	1	1
-1	1	-1	-1	-1	-1	1

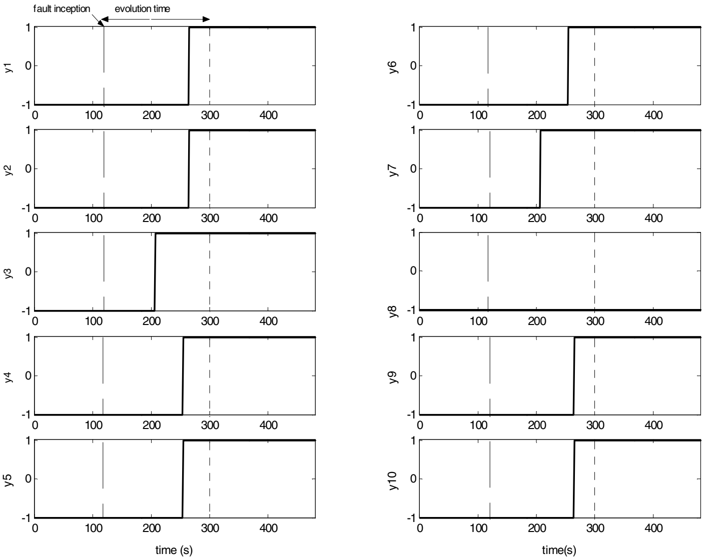


Fig. 3. Retrieved fault pattern via associative memory, *fault* 1

5 Conclusions

The obtained results illustrate that soft margin training proposed in this work is adequated to train associative memories based on RNNs. By means of this new approach, an associative memory is designed and applied to fault diagnosis in fossil electric power plants. Using soft margin training, all diagonal elements on connection matrix T are equals to diagonal elements in matrix A. This fact indicates that the total number of spurious memories is reduced. As a future work, it is necessary to analyze convergence properties for this new algorithm and it is necessary to establish the corresponding properties on connection matrix *T*.

$$LM = \begin{bmatrix} 0.00 & 0.00 & 0.10 & 0.00 & 0.02 & 0.05 & 0.10 & 0.00 & 0.05 & 0.00 \\ 0.08 & 0.12 & 0.00 & 0.10 & 0.00 & 0.00 & 0.00 & 0.06 & 0.06 & 0.08 \\ 0.00 & 0.00 & 0.00 & 0.08 & 0.10 & 0.10 & 0.03 & 0.02 & 0.00 & 0.00 \\ 0.09 & 0.04 & 0.00 & 0.10 & 0.08 & 0.10 & 0.03 & 0.00 & 0.10 & 0.09 \\ 0.00 & 0.00 & 0.06 & 0.00 & 0.10 & 0.08 & 0.10 & 0.00 & 0.04 & 0.00 \\ 0.05 & 0.00 & 0.03 & 0.03 & 0.10 & 0.10 & 0.10 & 0.02 & 0.10 & 0.05 \\ 0.07 & 0.06 & 0.00 & 0.08 & 0.04 & 0.03 & 0.03 & 0.10 & 0.02 & 0.07 \end{bmatrix} \tag{23}$$

$$WM = \begin{bmatrix} 0.51 & 0.37 & -0.07 & 0.22 & 0.22 & 0.22 & 0.07 & 0.37 & 0.37 & 0.51 & -0.22 \\ 0.30 & 0.42 & -0.18 & 0.68 & 0.30 & 0.06 & -0.06 & 0.18 & 0.18 & 0.30 & -0.30 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.26 & 0.43 & -0.08 & 0.64 & 0.61 & 0.26 & 0.08 & 0.08 & 0.08 & 0.26 & -0.26 \\ 0.13 & 0.04 & 0.13 & 0.00 & 0.13 & 0.13 & 0.22 & 0.04 & 0.04 & 0.13 & 0.04 \\ 0.10 & 0.03 & 0.10 & 0.05 & 0.10 & 0.23 & 0.03 & 0.03 & 0.16 & 0.10 & 0.03 \\ 0.03 & -0.03 & 0.16 & 0.03 & 0.03 & 0.03 & 0.23 & -0.03 & -0.03 & 0.03 & 0.10 \\ 0.50 & 0.30 & -0.30 & 0.22 & 0.10 & 0.10 & -0.10 & 0.70 & 0.30 & 0.50 & -0.50 \\ 0.14 & 0.08 & 0.02 & 0.00 & 0.02 & 0.14 & -0.02 & 0.08 & 0.20 & 0.14 & -0.02 \\ 0.51 & 0.37 & -0.07 & 0.22 & 0.22 & 0.22 & 0.07 & 0.37 & 0.37 & 0.51 & -0.22 \end{bmatrix}, \tag{24}$$

$$BV = [-0.07 \quad -0.80 \quad 0.80 \quad -0.52 \quad 0.16 \quad 0.20 \quad 0.26 \quad -0.88 \quad 0.05 \quad -0.07]^T, \tag{25}$$

$$A = \textit{Identity matrix} \text{ of } 10 \text{ X } 10, \tag{26}$$

$$T = \begin{bmatrix} 1.00 & 0.37 & -0.07 & 0.22 & 0.22 & 0.22 & 0.07 & 0.37 & 0.37 & 0.51 \\ 0.30 & 1.00 & -0.18 & 0.30 & 0.06 & 0.06 & -0.06 & 0.18 & 0.18 & 0.30 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.26 & 0.43 & -0.08 & 1.00 & 0.26 & 0.26 & 0.08 & 0.08 & 0.08 & 0.26 \\ 0.13 & 0.04 & 0.13 & 0.13 & 1.00 & 0.13 & 0.22 & 0.04 & 0.04 & 0.13 \\ 0.10 & 0.03 & 0.10 & 0.10 & 0.00 & 1.00 & 0.03 & 0.03 & 0.16 & 0.10 \\ 0.03 & -0.03 & 0.16 & 0.03 & 0.10 & 0.03 & 1.00 & -0.03 & -0.03 & 0.03 \\ 0.50 & 0.30 & -0.30 & 0.10 & 0.16 & 0.10 & -0.10 & 1.00 & 0.30 & 0.50 \\ 0.14 & 0.08 & 0.02 & 0.02 & 0.10 & 0.14 & -0.02 & 0.08 & 1.00 & 0.14 \\ 0.51 & 0.37 & -0.07 & 0.22 & 0.02 & 0.22 & 0.07 & 0.37 & 0.37 & 1.00 \end{bmatrix}, \tag{27}$$

$$I = [-0.29 \quad -1.10 \quad 0.80 \quad -0.78 \quad 0.21 \quad 0.23 \quad 0.36 \quad -1.38 \quad 0.02 \quad -0.29]^T. \tag{28}$$

Acknowledgment

The authors thank support of the CONACYT, Mexico on project 39866Y. Besides, authors also thank to the Process Supervision Department of IIE, Mexico, for allowing us to use its full scale simulator. The first author thanks support of UNACAR, Mexico on project PR/59/2006.

References

1. Liu, D., Lu, Z.: A new synthesis approach for feedback neural networks based on the perceptron training algorithm. IEEE Trans. Neural Networks, Vol. 8 (1997) 1468-1482.
2. Michel, A. N., Farrel, J. A.: Associative memories via artificial neural networks, IEEE Contr. Syst. Mag., Vol. 10 (1990) 6-17.
3. Cortes, C., Vapnik, V. N.: Support Vector Networks. Machine Learning, Vol. 20 (1995) 273-297.
4. Luemberger, D.: Linear and Non Linear Programming. Addison Wesley Publishing Company, USA, (1984).
5. Casali D., Constantini G., Perfetti R. and Ricci E.: Associative Memory Design Using Support Vector Machines, IEEE Transactions on Neural Networks, Vol. 17 (2006) 1165-1174.
6. Ruz-Hernandez, J. A., Sanchez, E. N. and Suarez, D. A.: Designing and associative memory via optimal training for fault diagnosis, Proceedings of International Joint Conference on Neural Networks, Vancouver, B. C., Canada, (2006), 8771-8778.
7. Ruz-Hernandez, J. A., Sanchez, E. N. and Suarez D. A.: Optimal training for associative memories: application to fault diagnosis in fossil electric power plants, Book Chapter of Hybrid Intelligent Systems Analysis and Design, International Series on Studies in Fuzzyness and Soft Computing, Edited by Castillo, O., P., Melin, J., Kacprzyc, and W., Pedrycz, Vol. 208, ISBN: 3-540-37419-1, (2007) 329-356.
8. Ruz-Hernandez, J. A. Development and application of a neural network-based scheme for fault diagnosis in fossil electric power plants (In Spanish). Ph. D. Thesis, CINVESTAV, Guadalajara Campus, (2006).

Part IV

Time Series Prediction

Modular Neural Networks with Fuzzy Integration Applied for Time Series Forecasting

Patricia Melin, Alejandra Mancilla, Miguel Lopez, Wendy Trujillo, Jose Cota, and Salvador Gonzalez

Division of Graduate Studies, Tijuana Institute of Technology, Mexico

Abstract. We describe in this paper the application of several neural network architectures to the problem of simulating and predicting the dynamic behavior of complex economic time series. We use several neural network models and training algorithms to compare the results and decide at the end, which one is best for this application. We also compare the simulation results with the traditional approach of using a statistical model. In this case, we use real time series of prices of consumer goods to test our models. Real prices of tomato and green onion in the U.S. show complex fluctuations in time and are very complicated to predict with traditional statistical approaches.

1 Introduction

Forecasting refers to a process by which the future behavior of a dynamical system is estimated based on our understanding and characterization of the system. If the dynamical system is not stable, the initial conditions become one of the most important parameters of the time series response, i.e. small differences in the start position can lead to a completely different time evolution. This is what is called sensitive dependence on initial conditions, and is associated with chaotic behavior [2, 16] for the dynamical system.

The financial markets are well known for wide variations in prices over short and long terms. These fluctuations are due to a large number of deals produced by agents that act independently from each other. However, even in the middle of the apparently chaotic world, there are opportunities for making good predictions [4,5]. Traditionally, brokers have relied on technical analysis, based mainly on looking at trends, moving averages, and certain graphical patterns, for performing predictions and subsequently making deals. Most of these linear approaches, such as the well-known Box-Jenkins method, have disadvantages [9].

More recently, soft computing [10] methodologies, such as neural networks, fuzzy logic, and genetic algorithms, have been applied to the problem of forecasting complex time series. These methods have shown clear advantages over the traditional statistical ones [12]. The main advantage of soft computing methodologies is that, we do not need to specify the structure of a model a-priori, which is clearly needed in the classical regression analysis [3]. Also, soft computing models are non-linear in nature and they can approximate more easily complex dynamical systems, than simple linear statistical models. Of course, there are also disadvantages in using soft computing models instead of statistical ones. In classical regression models, we can use the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 217–225, 2007.
springerlink.com

information given by the parameters to understand the process, i.e. the coefficients of the model can represent the elasticity of price for a certain good in the market. However, if the main objective if to forecast as closely as possible the time series, then the use of soft computing methodologies for prediction is clearly justified.

2 Monolithic Neural Network Models

A neural network model takes an input vector X and produces and output vector Y. The relationship between X and Y is determined by the network architecture. There are many forms of network architecture (inspired by the neural architecture of the brain). The neural network generally consists of at least three layers: one input layer, one output layer, and one or more hidden layers. Figure 1 illustrates a neural network with p neurons in the input layer, one hidden layer with q neurons, and one output layer with one neuron.

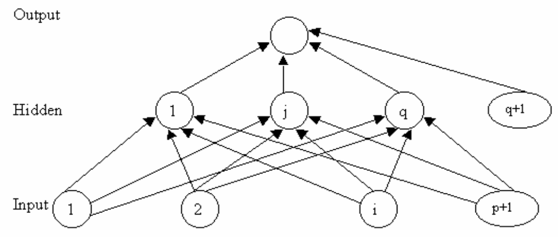


Fig. 1. Single hidden layer feedforward network

In the neural network we will be using, the input layer with p+1 processing elements, i.e., one for each predictor variable plus a processing element for the bias. The bias element always has an input of one, $X_{p+1}=1$. Each processing element in the input layer sends signals Xi (i=1,…,p+1) to each of the q processing elements in the hidden layer. The q processing elements in the hidden layer (indexed by j=1,…,q) produce an "activation" $a_j=F(\Sigma w_{ij}X_i)$ where w_{ij} are the weights associated with the connections between the p+1 processing elements of the input layer and the jth processing element of the hidden layer. Once again, processing element q+1 of the hidden layer is a bias element and always has an activation of one, i.e. $a_{q+1}=1$. Assuming that the processing element in the output layer is linear, the network model will be

$$Y_t = \sum_{j=1}^{p+1} \pi_\iota\, x_{it} + \sum_{j=1}^{p+1} \theta_j\, F\,(\,\sum_{i=1}^{p+1} w_{ij}\, x_{it}\,) \tag{1}$$

Here π_ι are the weights for the connections between the input layer and the output layer, and θ_j are the weights for the connections between the hidden layer and the output layer. The main requirement to be satisfied by the activation function F(·) is that it be nonlinear and differentiable. Typical functions used are the sigmoid, hyperbolic tangent, and the sine functions.

The weights in the neural network can be adjusted to minimize some criterion such as the sum of squared error (SSE) function:

$$E_l = ½ \sum_{l=1}^{n} (d_l - y_l)^2 \tag{2}$$

Thus, the weights in the neural network are similar to the regression coefficients in a linear regression model. In fact, if the hidden layer is eliminated, (1) reduces to the well-known linear regression function. It has been shown [22] that, given sufficiently many hidden units, (1) is capable of approximating any measurable function to any accuracy. In fact F(.) can be an arbitrary sigmoid function without any loss of flexibility.

The most popular algorithm for training feedforward neural networks is the backpropagation algorithm [14,18]. As the name suggests, the error computed from the output layer is backpropagated through the network, and the weights are modified according to their contribution to the error function. Essentially, backpropagation performs a local gradient search, and hence its implementation does not guarantee reaching a global minimum. A number of heuristics are available to partly address this problem, some of which are presented below. Instead of distinguishing between the weights of the different layers as in Equation (1), we refer to them generically as w_{ij} in the following.

After some mathematical simplification the weight change equation suggested by backpropagation can be expressed as follows:

$$w_{ij} = -\eta \frac{\partial E_l}{\partial w_{ij}} + \theta \Delta wij \tag{3}$$

Here, η is the learning coefficient and θ is the momentum term. One heuristic that is used to prevent the neural network from getting stuck at a local minimum is the random presentation of the training data.

3 Modular Neural Networks

There exists a lot of neural network architectures in the literature that work well when the number of inputs is relatively small, but when the complexity of the problem grows or the number of inputs increases, their performance decreases very quickly. For this reason, there has also been research work in compensating in some way the problems in learning of a single neural network over high dimensional spaces.

In the work of Sharkey [20], the use of multiple neural systems (Multi-Nets) is described. It is claimed that multi-nets have better performance or even solve problems that monolithic neural networks are not able to solve. It is also claimed that multi-nets or modular systems have also the advantage of being easier to understand or modify, if necessary.

In the literature there is also mention of the terms “ensemble” and “modular” for this type of neural network. The term “ensemble” is used when a redundant set of neural networks is utilized, as described in Hansen and Salamon [8]. In this case, each of the neural networks is redundant because it is providing a solution for the same task, as it is shown in Figure 2.

On the other hand, in the modular approach, one task or problem is decompose in subtasks, and the complete solution requires the contribution of all the modules, as it is shown in Figure 3.

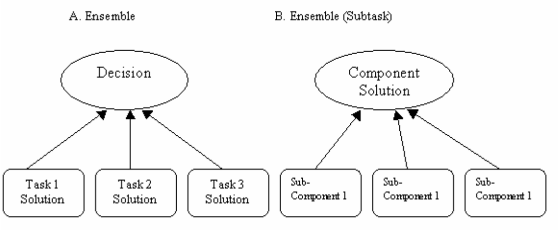


Fig. 2. Ensembles for one task and subtask

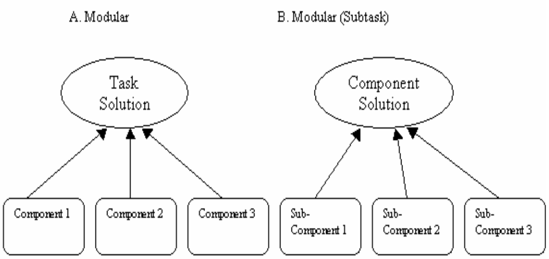


Fig. 3. Modular approach for task and subtask

4 Methods for Response Integration

In the literature we can find several methods for response integration, that have been researched extensively, which in many cases are based on statistical decision methods. We will mention briefly some of these methods of response integration, in particular the ones based on fuzzy logic. The idea of using these types of methods, is that the final decision takes into account all of the different kinds of information available about the time series. In particular, we consider aggregation operators, and the fuzzy Sugeno integral [21].

Yager [23] mentions in his work, that fuzzy measures for the aggregation criteria of two important classes of problems. In the first type of problems, we have a set $Z=\{z_1,z_2,\ldots,z_n\}$ of objects, and it is desired to select one or more of these objects based on the satisfaction of certain criteria. In this case, for each $z_i \in Z$, it is evaluated $D(z_i)=G(A_i(z_i),\ldots,A_j(z_i))$, and then an object or objects are selected based on the value of G. The problems that fall within this structure are the multi-criteria decision problems, search in databases and retrieving of documents.

In the second type of problems, we have a set $G=\{G_1,G_2,\ldots,G_q\}$ of aggregation functions and object z. Here, each G_k corresponds to different possible identifications of object z, and our goal is to find out the correct identification of z. For achieving this, for each aggregation function G, we obtain a result for each z, $D_k(z)=G_k(A_1(z), A_2(z), \dots ,A_n(z))$. Then we associate to z the identification corresponding to the larger value of the aggregation function.

A typical example of this type of problems is pattern recognition. Where A_j corresponds to the attributes and $A_j(z)$ measures the compatibility of z with the attribute. Medical applications and fault diagnosis fall into this type of problems. In diagnostic problems, the A_j corresponds to symptoms associated with a particular fault, and G_k captures the relations between these faults.

Fuzzy integrals can be viewed as non-linear functions defined with respect to fuzzy measures. In particular, the "gλ-fuzzy measure" introduced by Sugeno [21] can be used to define fuzzy integrals. The ability of fuzzy integrals to combine the results of multiple information sources has been mentioned in previous works.

Definition 1. A function of sets $g:2^x\text{-}(0.1)$ is called a fuzzy measure if:

1. $g(0)=0 \quad g(x)=1$
2. $g(A)\leq g(B)$ if $A\subset B$
3. if $\{A_i\}i^{\alpha}=1$ is a sequence of increments of the measurable set then

$$\lim_{i\to\infty} g(A_i) = g\,(\lim_{i\to\infty} A_i) \tag{4}$$

From the above it can be deduced that g is not necessarily additive, this property is replaced by the additive property of the conventional measure.

From the general definition of the fuzzy measure, Sugeno introduced what is called "gλ-fuzzy measure", which satisfies the following additive property: For every A, B $\subset$ X and $A \cap B = \theta$,

$$g(A \cup B) = g(A) + g(B) + \lambda\, g(A)g(B), \tag{5}$$

for some value of $\lambda>-1$.

This property says that the measure of the union of two disjunct sets can be obtained directly from the individual measures. Using the concept of fuzzy measures, Sugeno [21] developed the concept of fuzzy integrals, which are non-linear functions defined with respect to fuzzy measures like the gλ-fuzzy measure.

Definition 2. let X be a finite set and $h:X\to[0,1]$ be a fuzzy subset of X, the fuzzy integral over X of function h with respect to the fuzzy measure g is defined in the following way,

$$\begin{aligned} h(x) \circ g\,(x) &= \max_{E\subseteq X} [\, \min (\, \min_{x\in E} h(x), g(E))] \\ &= \sup_{\alpha\in[0,1]} [\min(\,\alpha\,, g(h_\alpha\,))] \end{aligned} \tag{6}$$

where h_α is the level set α of h,

$$h_\alpha = \{\, x \mid h(x) \geq \alpha \,\}. \tag{7}$$

We will explain in more detail the above definition: h(x) measures the degree to which concept h is satisfied by x. The term min(h_x) measures the degree to which concept h is satisfied by all the elements in E. The value g(E) is the degree to which the subset of objects E satifies the concept measure by g. As a consequence, the obtained value of comparing these two quantities in terms of operator min indicates the degree to which E satifies both criteria g and min(h_x). Finally, operator max takes the greatest of these terms.

5 Simulation and Forecasting Prices in the U.S. Market

We will consider the problem forecasting the prices of tomato in the U.S. market. The time series for the prices of this consumer good shows very complicated dynamic behavior, and for this reason it is interesting to analyze and predict the future prices for this good. We show in Figure 4 the time series of monthly tomato prices in the period of 1960 to 1999, to give an idea of the complex dynamic behavior of this time series.

We will apply both the modular and monolithic neural network approach and also the linear regression method to the problem of forecasting the time series of tomato prices. Then, we will compare the results of these approaches to select the best one for forecasting.

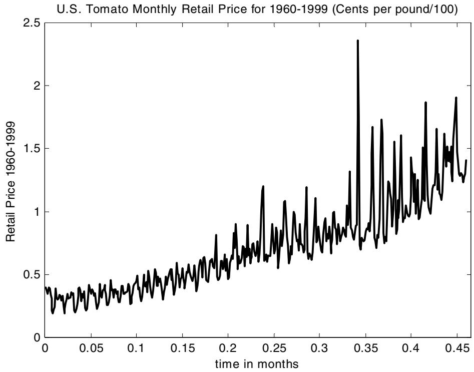


Fig. 4. Prices in US Dollars of tomato from January 1960 to December 1999

6 Experimental Results

We describe, in this section, the experimental results obtained by using neural networks to the problem of forecasting tomato prices in the U.S. Market. We show results of the application of several architectures and different learning algorithms to decide on the best one for this problem. We also compare at the end the results of the neural network approach with the results of linear regression models, to measure the difference in forecasting power of both methodologies.

First, we will describe the results of applying modular neural networks to the time series of tomato prices. We used the monthly data from 1960 to 1999 for training a

Modular Neural Network with four Modules, each of the modules with 80 neurons and one hidden layer. We show in Figure 5 the result of training the modular neural network with this data. In Figure 5, we can appreciate how the modular neural network approximates very well the real time series of tomato prices over the relevant period of time.

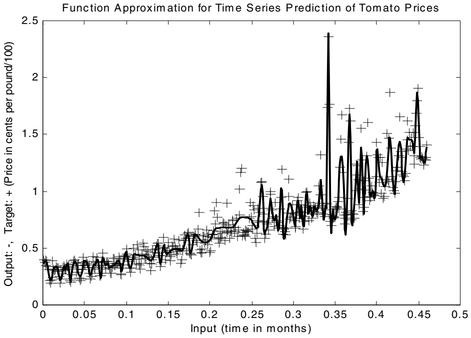


Fig. 5. Modular network for tomato prices with Levenberg-Marquardt algorithm

We have to mention that the results shown in Figure 5 are for the best modular neural network that we were able to find for this problem. We show in Figure 6 the comparison between several of the modular neural networks that we tried in our experiments. From Figure 6 we can appreciate that the modular neural network with one time delay and Leverberg-Marquardt (LM) training algorithm is the one that fits best the data and for this reason is the one selected.

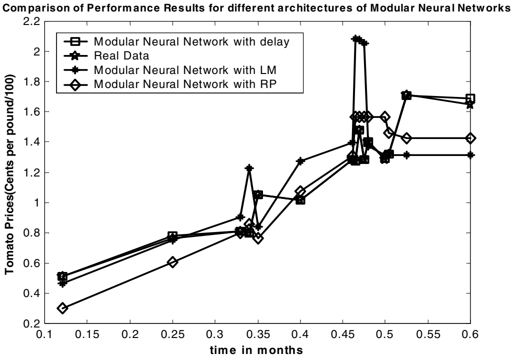


Fig. 6. Comparison of performance results for several modular neural networks

We show in Figure 7 the comparison of the best monolithic network against the best modular neural network. The modular network clearly fits better the real data of the problem.

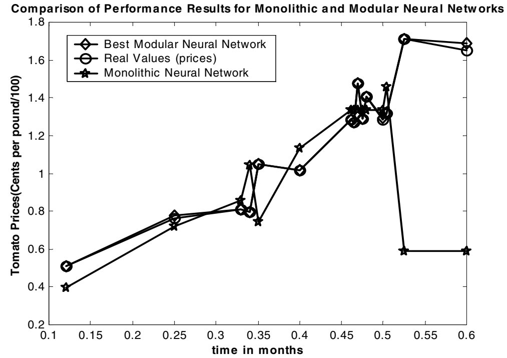


Fig. 7. Comparison of monolithic and modular neural networks

7 Conclusions

We described in this paper the use of modular neural networks for simulation and forecasting time series of consumer goods in the U.S. Market. We have considered a real case to test our approach, which is the problem of time series prediction of tomato prices in the U.S. market. We have applied monolithic and modular neural networks with different training algorithms to compare the results and decide which is the best option. The Levenberg-Marquardt learning algorithm gave the best results. The performance of the modular neural networks was also compared with monolithic neural networks. The forecasting ability of modular networks was clearly superior.

References

[1] Boers, E. and Kuiper, H. (1992) Biological Metaphors and the Design of Modular Artificial Neural Networks. Departments of Computer Science and Experimental and Theoretical Psychology at Leiden University, the Netherlands.

[2] Brock, W.A., Hsieh, D.A., and LeBaron, B. (1991). "Nonlinear Dynamics, Chaos and Instability", MIT Press, Cambridge, MA, USA.

[3] Castillo, O. and Melin, P. (1996). "Automated Mathematical Modelling for Financial Time Series Prediction using Fuzzy Logic, Dynamical System Theory and Fractal Theory", Proceedings of CIFEr'96, IEEE Press, New York, NY, USA, pp. 120-126.

[4] Castillo, O. and Melin P. (1998). "A New Fuzzy-Genetic Approach for the Simulation and Forecasting of International Trade Non-Linear Dynamics", Proceedings of CIFEr'98, IEEE Press, New York, USA, pp. 189-196.

[5] Castillo, O. and Melin, P. (1999). "Automated Mathematical Modelling for Financial Time Series Prediction Combining Fuzzy Logic and Fractal Theory", Edited Book "Soft Computing for Financial Engineering", Springer-Verlag, Germany, pp. 93-106.

[6] O. Castillo and P. Melin, Soft Computing and Fractal Theory for Intelligent Manufacturing. Springer-Verlag, Heidelberg, Germany, 2003.

[7] Fu, H.-C., Lee, Y.-P., Chiang, C.-C., and Pao, H.-T. (2001). Divide-and-Conquer Learning and Modular Perceptron Networks in IEEE Transaction on Neural Networks, vol. 12, No. 2, pp. 250-263.

[8] Hansen, L. K. and Salamon P. (1990). Neural Network Ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, pp. 993-1001.

[9] Haykin, S. (1996). “Adaptive Filter Theory”, Third Edition, Prentice Hall.

[10] Jang, J.-S. R., Sun, C.-T., and Mizutani, E. (1997). “Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, Prentice Hall.

[11] Lu, B. and Ito, M. (1998). Task Decomposition and module combination based on class relations: modular neural network for pattern classification. Technical Report, Nagoya Japan, 1998.

[12] Maddala, G.S. (1996). “Introduction to Econometrics”, Prentice Hall.

[13] Murray-Smith, R. and Johansen, T. A. (1997). Multiple Model Approaches to Modeling and Control. Taylor and Francis, UK.

[14] Parker, D.B. (1982). “ Learning Logic”, Invention Report 581-64, Stanford University.

[15] Quezada, A. (2004). Reconocimiento de Huellas Digitales Utilizando Redes Neuronales Modulares y Algoritmos Geneticos. Thesis of Computer Science, Tijuana Institute of Technology, Mexico.

[16] Rasband, S.N. (1990). “Chaotic Dynamics of Non-Linear Systems”, Wiley.

[17] Ronco, E. and Gawthrop, P. J. (1995). Modular neural networks: A State of the Art. Technical Report, Center for System and Control. University of Glasgow, Glasgow, UK, 1995.

[18] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). “Learning Internal Representations by Error Propagation”, in “Parallel Distributed Processing: Explorations in the Microstructures of Cognition”, MIT Press, Cambridge, MA, USA, Vol. 1, pp. 318-362.

[19] Schdmit, A. and Bandar, Z. (1997). A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors, Proceedings of ICANNGA'97, Norwich, England.

[20] Sharkey, A. (1999). Combining Artificial Neural Nets: Ensemble and Modular Multi-Nets Systems, Ed. Springer-Verlag, London, England.

[21] Sugeno, M. (1974). Theory of fuzzy integrals and its application, Doctoral Thesis, Tokyo Institute of Technology, Japan.

[22] White, H. (1989). “An Additional Hidden Unit Test for Neglected Non-linearity in Multilayer Feedforward Networks”, Proceedings of IJCNN’89, Washington, D.C., IEEE Press, pp. 451-455.

[23] Yager, R. R. (1999). Criteria Aggregations Functions Using Fuzzy Measures and the Choquet Integral, International Journal of Fuzzy Systems, Vol. 1, No. 2.

Predicting Job Completion Time in a Wafer Fab with a Recurrent Hybrid Neural Network*

Toly Chen

Department of Industrial Engineering and Systems Management, Feng Chia University, 100,
Wenhwa Road, Seatwen, Taichung City, Taiwan
tolychen@ms37.hinet.net
http://www.geocities.com/tinchihchen/

Abstract. Predicting the completion time of a job is a critical task to a wafer fabrication plant (wafer fab). Many recent studies have shown that pre-classifying a job before predicting the completion time was beneficial to prediction accuracy. However, most classification approaches applied in this field could not absolutely classify jobs. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For tackling these problems, a recurrent hybrid neural network is proposed in this study, in which a job is pre-classified into one category with the k-means (kM) classifier, and then the back propagation network (BPN) tailored to the category is applied to predict the completion time of the job. After that, the prediction error is fed back to the kM classifier to adjust the classification result, and then the completion time of the job is predicted again. After some replications, the prediction accuracy of the hybrid kM-BPN system will be significantly improved.

1 Introduction

Predicting the completion time for every job in a semiconductor fabrication plant is a critical task not only to the plant itself, but also to its customers. After the completion time of each job in a semiconductor fabrication plant is accurately predicted, several managerial goals can be simultaneously achieved [5]. Predicting the completion time of a job is equivalent to estimating the cycle (flow) time of the job, because the former can be easily derived by adding the release time (a constant) to the latter. There are six major approaches commonly applied to predicting the completion/cycle time of a job in a semiconductor fabrication plant: multiple-factor linear combination (MFLC), production simulation (PS), back propagation networks (BPN), case based reasoning (CBR), fuzzy modeling methods, and hybrid approaches. Among the six approaches, MFLC is the easiest, quickest, and most prevalent in practical applications. The major disadvantage of MFLC is the lack of forecasting accuracy [5]. Conversely, huge amount of data and lengthy simulation time are two disadvantages of PS. Nevertheless, PS is the most accurate completion time prediction approach if the related databases are continuously updated to maintain enough validity, and often serves as a benchmark for evaluating the effectiveness of another method. PS also

* This work was support by the National Science Council, R.O.C.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 226–235, 2007.
springerlink.com

tends to be preferred because it allows for computational experiments and subsequent analyses without any actual execution [3]. Considering both effectiveness and efficiency, Chang et al. [4] and Chang and Hsieh [2] both forecasted the completion/cycle time of a job in a semiconductor fabrication plant with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy measured with root mean squared error (RMSE) was considerably improved with these BPNs. On the other hand, much less time and fewer data are required to generate an completion time forecast with a BPN than with PS. Chang et al. [3] proposed a k-nearest-neighbors based case-based reasoning (CBR) approach which outperformed the BPN approach in forecasting accuracy. Chang et al. [4] modified the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the fuzzy modeling method proposed by Wang and Mendel [22], called the WM method, with a simple genetic algorithm (GA) and proposed the evolving fuzzy rule (EFR) approach to predict the cycle time of a job in a semiconductor fabrication plant. Their EFR approach outperformed CBR and BPN in prediction accuracy. Chen [5] constructed a fuzzy BPN (FBPN) that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN especially in the efficiency respect. Chen's FBPN was applied to evaluate the achievability of a completion time forecast in Chen [6]. Recently, Chen [9] constructed a look-ahead FBPN for the same purpose, and showed that taking the future release plan into consideration did improve the performance. Chen and Lin [12] proposed a hybrid k-means (kM) and BPN approach for estimating job completion time in a semiconductor fabrication plant. Subsequently, Chen [10] constructed a kM-FBPN system for the same purpose. Similarly, Chen [8] combined SOM and BPN, in which a job was classified using SOM before predicting the completion time of the job with BPN. Chen [7] proposed a hybrid look-ahead SOM-FBPN and FIR system for job completion time prediction and achievability evaluation. The hybrid system outperformed many existing hybrid approaches with more accurate completion time estimation. Subsequently, Chen et al. [11] added a selective allowance to the completion time predicted using Chen's approach to determine the internal due date. The results of these studies showed that pre-classifying jobs was beneficial to prediction accuracy.

However, most classification approaches applied in this field could not absolutely classify jobs. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For tackling these problems, a recurrent hybrid neural network is proposed in this study, in which a job is pre-classified into one category with the kM classifier, and then the BPN tailored to the category is applied to predict the completion time of the job. After that, the prediction error is fed back to the kM classifier to adjust the classification result, and then the completion time of the job is predicted again. After some replications, the prediction accuracy of the hybrid kM-BPN system will be significantly improved. The methodology architecture is shown in Fig. 1. To evaluate the effectiveness of the proposed methodology, production simulation is applied to generate test data.

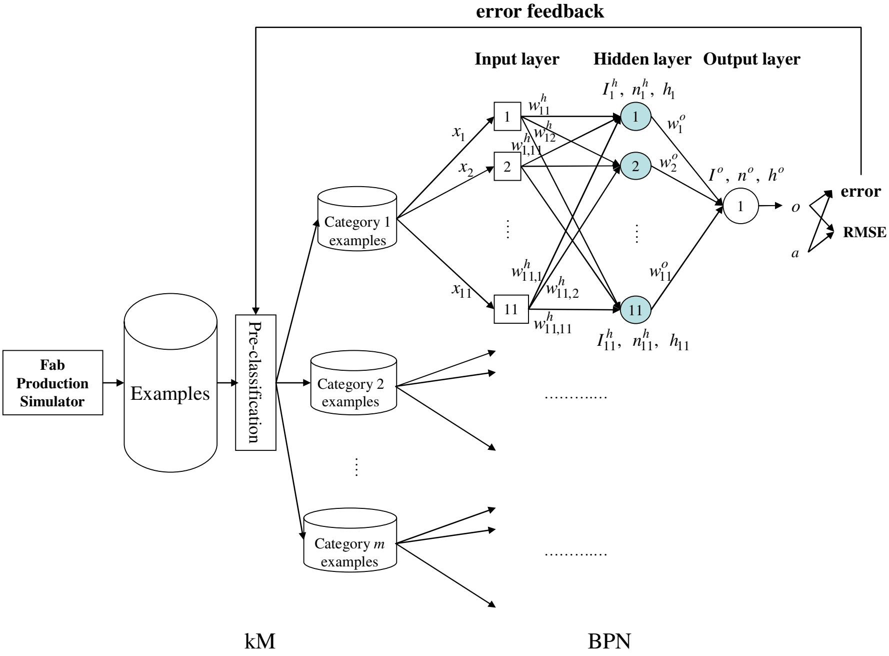


Fig. 1. The methodology architecture

2 Methodology

The parameters used in this study are defined:

1. R_n: the release time of the n-th example/job.
2. U_n: the average fab utilization at R_n.
3. Q_n: the total queue length on the job's processing route at R_n.
4. BQ_n: the total queue length before bottlenecks at R_n.
5. FQ_n: the total queue length in the whole fab at R_n. Obviously, $FQ_n \geq Q_n \geq BQ_n$.
6. WIP_n: the fab work-in-process (WIP) at R_n.
7. $D_n^{(i)}$: the latenesses of the i-th most recently completed jobs.
8. $FDW_n^{(f)}$: the f-th nearest future discounted workload of job n.
9. E_n: the prediction error of job n, which is set to be zero in the beginning.

The proposed methodology is composed of two steps. Firstly, a FBPN is constructed to predict the completion time of a job.

2.1 Job Classification with kM

The procedure of applying kM in forming inputs to the BPN is now detailed. Every job fed into the BPN is called an example. Examples are pre-classified to m categories before they are fed into the BPN according to their Euclidean distances to the category centers:

$$d(n,\ j) = \sqrt{\begin{array}{l} (U_n - U_{(j)})^2 + (Q_n - Q_{(j)})^2 + (BQ_n - BQ_{(j)})^2 + (FQ_n - FQ_{(j)})^2 \\ + (WIP_n - WIP_{(j)})^2 + (D_n^{(1)} - D_{(j)}^{(1)})^2 + (D_n^{(2)} - D_{(j)}^{(2)})^2 + (D_n^{(3)} - D_{(j)}^{(3)})^2 \\ + (FDW_n^{(1)} - FDW_{(j)}^{(1)})^2 + (FDW_n^{(2)} - FDW_{(j)}^{(2)})^2 + (FDW_n^{(3)} - FDW_{(j)}^{(3)} \\ + (E_n - E_{(j)})^2 \end{array}} \tag{1}$$

where parameters $U_{(j)}$ ~ $E_{(j)}$ denote the parameters of the *j*-th category center, which is arbitrarily chosen from those of all examples in the beginning. In this way, job *n* is classified to category *j* with the smallest $d(n, j)$. Each time after all examples are classified, the parameter sets of all category centers are recalculated by averaging those of the examples clustered in the same categories.

Example classification is continued until the sum of the average Euclidean distances (SADs) from examples to their category centers in all categories converges to a minimal value:

$$SAD = \sum_{j=1}^{m} \frac{\sum\limits_{\text{all job } n \text{ in category } j} d(n,\ j)}{\text{number of jobs in category } j}. \tag{2}$$

Examples of different categories are then learned with different BPNs but with the same topology. The procedure for determining the parameters is described in the next section.

2.2 Job Completion Time Prediction Within Each Category Using a BPN

The configuration of the BPN is established as follows:

1. Inputs: eleven parameters associated with the *n*-th example/job including U_n, Q_n, BQ_n, FQ_n, WIP_n, $D_n^{(i)}$ (i = 1~3), and $FDW_n^{(f)}$ (f = 1~3). These parameters have to be normalized so that their values fall within [0, 1].
2. Single hidden layer: Generally one or two hidden layers are more beneficial for the convergence property of the network.
3. Number of neurons in the hidden layer: the same as that in the input layer. Such a treatment has been adopted by many studies (e.g. [2, 5]).
4. Output: the (normalized) cycle time forecast of the example.
5. Network learning rule: Delta rule.
6. Transformation function: Sigmoid function,

$$f(x) = 1/(1 + e^{-x}). \tag{3}$$

7. Learning rate (λ): 0.01~1.0.
8. Batch learning.

The procedure for determining the parameter values is now described. A portion of the examples is fed as "training examples" into the BPN to determine the parameter values. Two phases are involved at the training stage. At first, in the forward phase, inputs are multiplied with weights, summated, and transferred to the hidden layer. Then activated signals are outputted from the hidden layer as:

$$h_l = 1/(1 + e^{-n_l^h}), \tag{4}$$

where

$$n_l^h = I_l^h - \theta_l^h, \tag{5}$$

$$I_l^h = \sum_{i=1}^{11} w_{il}^h x_i. \tag{6}$$

h_l's are also transferred to the output layer with the same procedure. Finally, the output of the BPN is generated as:

$$o = 1/(1 + e^{-n^o}), \tag{7}$$

where

$$n^o = I^o - \theta^o, \tag{8}$$

$$I^o = \sum_{l=1}^{11} w_l^o h_l. \tag{9}$$

Then the output o is compared with the normalized actual cycle time a, for which prediction error E and RMSE are calculated:

$$E = |o - a|. \tag{10}$$

$$RMSE = \sqrt{\sum_{\text{all examples}} (o - a)^2 / \text{number of examples}}. \tag{11}$$

Subsequently in the backward phase, the deviation between o and a is propagated backward, and the error terms of neurons in the output and hidden layers can be calculated respectively as:

$$\delta^o = o(1 - o)(a - o), \tag{12}$$

$$\delta_l^h = h_l(1 - h_l) w_l^o \delta^o. \tag{13}$$

Based on them, adjustments that should be made to the connection weights and thresholds can be obtained as

$$\Delta w_l^o = \eta \delta^o h_l, \tag{14}$$

$$\Delta w_{il}^{h}=\eta \delta_{l}^{h} x_{i}, \tag{15}$$

$$\Delta \theta^{o}=-\eta \delta^{o}, \tag{16}$$

$$\Delta \theta_{l}^{h}=-\eta \delta_{l}^{h}. \tag{17}$$

To accelerate convergence, a momentum can be added to the learning expressions. For example,

$$\Delta w_{l}^{o}=\eta \delta^{o} h_{l}+\alpha(w_{l}^{o}(t)-w_{l}^{o}(t-1)). \tag{18}$$

Theoretically, network-learning stops when RMSE falls below a pre-specified level, or the improvement in RMSE becomes negligible with more epochs, or a large number of epochs have already been run. Then the remaining portion of the adopted examples in each category is used as "test examples" and fed into the BPN to evaluate the accuracy of the network that is also measured with RMSE.

2.3 Recurrent – Feeding Back Prediction Error to Adjust Classification

After training and testing, the BPN of every category is applied to predicting the cycle times of jobs belonging to the category. Then prediction error *E* is calculated for every job according to equation (10), and is fed back to the kM classifier. Subsequently, classification is done again. If the classification result is the same as the previous one, then stop; otherwise, the BPN of every category has to be reconstructed again with the same procedure mentioned above. Finally, these BPNs can be applied to predicting the flow time of a new job.

When a new job is released into the fab, the eleven parameters associated with the new job are recorded and the membership of belonging to each category is calculated. Then the BPN of the category to which the membership of belonging is the highest will be applied to predicting the cycle time of the new job.

3 A Demonstrative Example from a Simulated Wafer Fab

In practical situations, the history data of each job is only partially available in the fab. Further, some information of the previous jobs such as Q_n, BQ_n, and FQ_n is not easy to collect on the shop floor. Therefore, a simulation model is often built to simulate the manufacturing process of a real wafer fab [1-14]. Then, such information can be derived from the shop floor status collected from the simulation model [3]. To generate a demonstrative example, a simulation program coded using Microsoft Visual Basic .NET is constructed to simulate a wafer fab with the following assumptions:

1. Jobs are uniformly distributed into the fab.
2. The distributions of the interarrival times of machine downs are exponential.
3. The distribution of the time required to repair a machine is uniform.

4. The percentages of jobs with different product types in the fab are predetermined. As a result, this study is only focused on fixed-product-mix cases. However, the product mix in the simulated fab does fluctuate and is only approximately fixed in the long term.
5. The percentages of jobs with different priorities released into the fab are controlled.
6. The priority of a job cannot be changed during fabrication.
7. Jobs are sequenced on each machine first by their priorities, then by the first-in-first-out (FIFO) policy. Such a sequencing policy is a common practice in many foundry fabs.
8. A job has equal chances to be processed on each alternative machine/head available at a step.
9. A job cannot proceed to the next step until the fabrication on its every wafer has been finished. No preemption is allowed.

The basic configuration of the simulated wafer fab is the same as a real-world semiconductor fabrication plant which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. A trace report was generated every simulation run for verifying the simulation model. The simulated average cycle times have also been compared with the actual values to validate the simulation model. Assumptions (1)~(3), and (7)~(9) are commonly adopted in related studies (e.g. [2-5]), while assumptions (4)~(6) are made to simplify the situation. There are five products (labeled as A~E) in the simulated plant. A fixed product mix is assumed. The percentages of these products in the plant's product mix are assumed to be 35%, 24%, 17%, 15%, and 9%, respectively. The simulated plant has a monthly capacity of 20,000 pieces of wafers and is expected to be fully utilized (utilization = 100%). Jobs are of a standard size of 24 wafers per job. Jobs are released one by one every 0.85 hours. Three types of priorities (normal, hot, and super hot) are randomly assigned to jobs. The percentages of jobs with these priorities released into the fab are restricted to be approximately 60%, 30%, and 10%, respectively. Each product has 150~200 steps and 6~9 reentrances to the most bottleneck machine. The singular production characteristic "reentry" of the semiconductor industry is clearly reflected in the example. It also shows the difficulty for the production planning and scheduling people to provide an accurate due-date for the product with such a complicated routing. Totally 102 machines (including alternative machines) are provided to process single-wafer or batch operations in the fab. Thirty replicates of the simulation are successively run. The time required for each simulation replication is about 12 minute on a PC with 512MB RAM and Athlon™ 64 Processor 3000+ CPU. A horizon of 24 months is simulated. The maximal cycle time is less than 3 months. Therefore, four months and an initial WIP status (obtained from a pilot simulation run) seemed to be sufficient to drive the simulation into a steady state. The statistical data were collected starting at the end of the fourth month. For each replicate, data of 30 jobs are collected and classified by their product types and priorities. Totally, data of 900 jobs can be collected as training and testing examples. Among them, 2/3 (600 jobs, including all product types and priorities) are used to train the network, and the other 1/3 (300 jobs) are reserved for testing.

3.1 Results and Discussions

To evaluate the effectiveness of the proposed methodology and to make comparisons with four approaches – MFLC, BPN, CBR, and kM-BPN, all the five methods were applied to five test cases containing the data of full-size (24 wafers per job) jobs with different product types and priorities. The convergence condition was established as either the improvement in RMSE becomes less than 0.001 with one more epoch, or 1000 epochs have already been run. The minimal RMSEs achieved by applying the five approaches to different cases were recorded and compared in Table 1. RMSE is adopted instead of mean absolute error (MAE) because the same measure has been adopted in the previous studies in this field (e.g. [2-5]), namely, to facilitate comparison. As noted in Chang and Liao [5], the *k*-nearest-neighbors based CBR approach should be fairly compared with a BPN trained with only randomly chosen *k* cases. MFLC was adopted as the comparison basis, and the percentage of improvement on the minimal RMSE by applying another approach is enclosed in parentheses following the performance measure. The optimal value of parameter *k* in the CBR approach was equal to the value that minimized RMSE [5].

Table 1. Comparisons of the RMSEs of various approaches

RMSE	A(normal)	A(hot)	A(super hot)	B(normal)	B(hot)
MFLC	185.1	106.01	12.81	302.86	79.94
BPN	177.1(-4%)	102.27(-4%)	12.23(-5%)	286.93(-5%)	75.98(-5%)
CBR	172.44(-7%)	86.66(-18%)	11.59(-10%)	295.51(-2%)	78.85(-1%)
kM-BPN	163.06(-12%)	67.7(-36%)	10.37(-19%)	197.89(-35%)	41.85(-48%)
The proposed methodology	151.02(-18%)	61.64(-42%)	10.35(-19%)	179.22(-41%)	40.94(-49%)

According to experimental results, the following discussions are made:

1. From the effectiveness viewpoint, the prediction accuracy (measured with RMSE) of the proposed methodology was significantly better than those of the other approaches by achieving a 18%~49% (and an average of 34%) reduction in RMSE over the comparison basis – MFLC. The average advantage over CBR was 23%.
2. The effect of pre-classification is revealed by the fact that the prediction accuracy of kM-BPN was considerably better than that of BPN without pre-classification in all cases with an average advantage of 25%.
3. The effect of feeding back prediction error to adjust classification, i.e. the recurrent effect, was revealed with the fact that the proposed methodology outperformed kM-BPN without the recurrent property in all cases with an average advantage of 4%.

4 Conclusions and Directions for Future Research

Most classification approaches applied to job completion time prediction in a wafer fab could not absolutely classify jobs. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For these reasons, to further enhance the effectiveness of job completion time prediction in a wafer fab, a recurrent hybrid neural network is proposed in this study, in which a job is pre-classified into one category with the kM classifier, and then the BPN tailored to the category is applied to predict the completion time of the job. After that, the prediction error is fed back to the kM classifier to adjust the classification result, and then the completion time of the job is predicted again. After some replications, the prediction accuracy of the hybrid kM-BPN system will be significantly improved. For evaluating the effectiveness of the proposed methodology and to make comparisons with some existing approaches, PS is applied in this study to generate test data. According to experimental results, the prediction accuracy (measured with RMSE) of the proposed methodology was significantly better than those of the other approaches in most cases by achieving a 18%~49% (and an average of 34%) reduction in RMSE over the comparison basis – MFLC. The recurrent effect was also obvious.

However, to further evaluate the advantages and disadvantages of the proposed methodology, it has to be applied to fab models of different scales, especially a full-scale actual wafer fab. In addition, the proposed methodology can also be applied to cases with changing product mixes or loosely controlled priority combinations, under which the cycle time variation is often very large. These constitute some directions for future research.

References

1. Barman, S.: The Impact of Priority Rule Combinations on Lateness and Tardiness. IIE Transactions 30 (1998) 495-504
2. Chang, P.-C., Hsieh, J.-C.: A Neural Networks Approach for Due-date Assignment in a Wafer Fabrication Factory. International Journal of Industrial Engineering 10(1) (2003) 55-61
3. Chang, P.-C., Hsieh, J.-C., Liao, T. W.: A Case-based Reasoning Approach for Due Date Assignment in a Wafer Fabrication Factory. In: Proceedings of the International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, British Columbia, Canada (2001)
4. Chang, P.-C., Hsieh, J.-C., Liao, T. W.: Evolving Fuzzy Rules for Due-date Assignment Problem in Semiconductor Manufacturing Factory. Journal of Intelligent Manufacturing 16 (2005) 549-557
5. Chen, T.: A Fuzzy Back Propagation Network for Output Time Prediction in a Wafer Fab. Applied Soft Computing 2/3F (2003) 211-222
6. Chen, T.: A Fuzzy Set Approach for Evaluating the Achievability of an Output Time Forecast in a Wafer Fabrication Plant. Lecture Notes in Artificial Intelligence 4293 (2006) 483-493
7. Chen, T.: A Hybrid Look-ahead SOM-FBPN and FIR System for Wafer-lot-output Time Prediction and Achievability Evaluation. International Journal of Advanced Manufacturing Technology (2007) DOI 10.1007/s00170-006-0741-x
8. Chen, T.: A Hybrid SOM-BPN Approach to Lot Output Time Prediction in a Wafer Fab. Neural Processing Letters 24(3) (2006) 271-288

9. Chen, T.: A Look-ahead Fuzzy Back Propagation Network for Lot Output Time Series Prediction in a Wafer Fab. Lecture Notes in Computer Science 4234 (2006) 974-982
10. Chen, T.: Applying an Intelligent Neural System to Predicting Lot Output Time in a Semiconductor Fabrication Factory. Lecture Notes in Computer Science 4234 (2006) 581-588
11. Chen, T., Jeang, A., Wang, Y.-C.: A Hybrid Neural Network and Selective Allowance Approach for Internal Due Date Assignment in a Wafer Fabrication Plant. International Journal of Advanced Manufacturing Technology (2007) DOI 10.1007/s00170-006-0869-8
12. Chen, T., Lin, Y.-C.: A Hybrid and Intelligent System for Predicting Lot Output Time in a Semiconductor Fabrication Factory. Lecture Notes in Artificial Intelligence 4259 (2006) 757-766
13. Chung, S.-H., Yang, M.-H., Cheng, C.-M.: The Design of Due Date Assignment Model and the Determination of Flow Time Control Parameters for the Wafer Fabrication Factories. IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part C 20(4) (1997) 278-287
14. Foster, W. R., Gollopy, F., Ungar, L. H.: Neural Network Forecasting of Short, Noisy Time Series. Computers in Chemical Engineering 16(4) (1992) 293-297
15. Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
16. Hung, Y.-F., Chang, C.-B.: Dispatching Rules Using Flow Time Predictions for Semiconductor Wafer Fabrications. In: Proceedings of the 5th Annual International Conference on Industrial Engineering Theory, Applications and Practice, Taiwan (2001)
17. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed Representation of Fuzzy Rules and Its Application to Pattern Classification. Fuzzy Sets and Systems 52(1) (1992) 21-32
18. Lin, C.-Y.: Shop Floor Scheduling of Semiconductor Wafer Fabrication Using Real-time Feedback Control and Prediction. Ph.D. Dissertation, Engineering-Industrial Engineering and Operations Research, University of California at Berkeley (1996)
19. Piramuthu, S.: Theory and Methodology – Financial Credit-risk Evaluation with Neural and Neuralfuzzy Systems. European Journal of Operational Research 112 (1991) 310-321
20. Ragatz, G. L., Mabert, V. A.: A Simulation Analysis of Due Date Assignment. Journal of Operations Management 5 (1984) 27-39
21. Vig, M. M., Dooley, K. J.: Dynamic Rules for Due-date Assignment. International Journal of Production Research 29(7) (1991) 1361-1377
22. Wang, L.-X., Mendel, J. M.: Generating Fuzzy Rules by Learning from Examples. IEEE Transactions on Systems, Man, and Cybernetics 22(6) (1992) 1414-1427
23. Weeks, J. K.: A Simulation Study of Predictable Due-dates. Management Science 25 (1979) 363–373

A Hybrid ANN-FIR System for Lot Output Time Prediction and Achievability Evaluation in a Wafer Fab

Toly Chen

Department of Industrial Engineering and Systems Management, Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung City, Taiwan
tolychen@ms37.hinet.net
http://www.geocities.com/tinchihchen/

Abstract. A hybrid artificial neural network (ANN)-fuzzy inference rules (FIR) system is constructed in this study for lot output time prediction and achievability evaluation in a fabrication plant (wafer fab), which are critical tasks to the wafer fab. At first, a hybrid and recurrent ANN, i.e. self-organization map (SOM) and fuzzy back propagation network (FBPN), is proposed to predict the output time of a wafer lot. According to experimental results, the prediction accuracy of the hybrid ANN was significantly better than those of some existing approaches. Subsequently, a set of fuzzy inference rules is established to evaluate the achievability of an output time forecast.

1 Introduction

Predicting the output time for every lot in a wafer fab is a critical task not only to the fab itself, but also to its customers. After the output time of each lot in a wafer fab is accurately predicted, several managerial goals can be simultaneously achieved [5]. Predicting the output time of a wafer lot is equivalent to estimating the cycle time of the lot, because the former can be easily derived by adding the release time (a constant) to the latter.

There are six major approaches commonly applied to predicting the output/cycle time of a lot in a wafer fab: multiple-factor linear combination (MFLC), production simulation (PS), back propagation networks (BPN), case based reasoning (CBR), fuzzy modeling methods, and hybrid approaches. Among the six approaches, MFLC is the easiest, quickest, and most prevalent in practical applications. The major disadvantage of MFLC is the lack of forecasting accuracy [5]. Conversely, huge amount of data and lengthy simulation time are two disadvantages of PS. Nevertheless, PS is the most accurate output time prediction approach if the related databases are continuously updated to maintain enough validity, and often serves as a benchmark for evaluating the effectiveness of another method. PS also tends to be preferred because it allows for computational experiments and subsequent analyses without any actual execution [3]. Considering both effectiveness and efficiency, Chang et al. [4] and Chang and Hsieh [2] both forecasted the output/cycle time of a lot in a wafer fab with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy measured with root mean squared error (RMSE) was considerably improved with these BPNs. On the other hand, much less time and fewer data are

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 236–245, 2007.
springerlink.com

required to generate an output/cycle time forecast with a BPN than with PS. Chang et al. [3] proposed a k-nearest-neighbors based case-based reasoning (CBR) approach which outperformed the BPN approach in forecasting accuracy. Chang et al. [4] modified the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the fuzzy modeling method proposed by Wang and Mendel [22], called the WM method, with a simple genetic algorithm (GA) and proposed the evolving fuzzy rule (EFR) approach to predict the cycle time of a lot in a wafer fab. Their EFR approach outperformed CBR and BPN in prediction accuracy. Chen [5] constructed a fuzzy BPN (FBPN) that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN especially in the efficiency respect. Chen's FBPN was applied to evaluate the achievability of an output time forecast in Chen [6]. Recently, Chen [9] constructed a look-ahead FBPN for the same purpose, and showed that taking the future release plan into consideration did improve the performance. Chen and Lin [12] proposed a hybrid k-means (kM) and BPN approach for estimating lot output time in a wafer fab. Subsequently, Chen [10] constructed a kM-FBPN system for the same purpose. Similarly, Chen [8] combined SOM and BPN, in which a lot was classified using SOM before predicting the output/cycle time of the lot with BPN. Chen [7] proposed a hybrid look-ahead SOM-FBPN and FIR system for lot output/cycle time prediction and achievability evaluation. The hybrid system outperformed many existing hybrid approaches with more accurate output/cycle time estimation. Subsequently, Chen et al. [11] added a selective allowance to the output time predicted using Chen's approach to determine the internal due date. The results of these studies showed that pre-classifying lots was beneficial to prediction accuracy.

A hybrid artificial neural network (ANN)-fuzzy inference rules (FIR) system is constructed in this study for lot output time prediction and achievability evaluation in a wafer fab. At first, a hybrid and recurrent ANN (SOM and FBPN) is proposed to predict the output time of a wafer lot. The hybrid ANN is recurrent, which is different from the existing approaches in this field. PS is also applied in this study to generate test examples. According to experimental results, the prediction accuracy of the hybrid ANN was significantly better than those of some existing approaches. Subsequently, a set of FIRs are established to evaluate the achievability of an output time forecast. With the proposed methodology, both output time prediction and achievability evaluation can be concurrently accomplished.

2 Methodology

In this paper, a hybrid ANN-FIR system is constructed for lot output time prediction and achievability evaluation in a wafer fab. The proposed methodology is composed of two parts. In the first part, a hybrid and recurrent ANN (SOM and FBPN) is proposed to predict the output time of a wafer lot.

2.1 Output Time Prediction with a Hybrid ANN (SOM-FBPN)

The parameters used in the following are defined:

1. U_n: the average fab utilization at the release time of the n-th example/lot.
2. Q_n: the total queue length on the lot's processing route at the release time.

3. BQ_n: the total queue length before bottlenecks at the release time.
4. FQ_n: the total queue length in the whole fab at the release time.
5. WIP_n: fab WIP at the release time.
6. $D_n^{(i)}$: the delay of the i-th recently completed lots, i = 1~3.
7. E_n: the prediction error of lot n, which is set to be zero in the beginning.

2.1.1 Lot Classification with SOM

Every lot fed into the FBPN is called an example. Examples are pre-classified into different categories before they are fed into the FBPN with SOM. Let $X=\{x_1, x_2, \ldots, x_n\}$ denote the set of feature vectors corresponding to examples. Each item x_i is a nine-dimensional feature vector whose elements are the U_n, Q_n, BQ_n, FQ_n, WIP_n, $D_n^{(i)}$ (i = 1~3), and E_n of the corresponding example. These feature vectors are fed into a SOM network. After the training is accomplished, input vectors that are topologically close are mapped to the same category, which means the input space is divided into k categories, and each example is associated with a certain category. Then, the classification result is post-processed, including eliminating isolated examples, merging small blocks, etc. Finally, the classification is finished.

After classification, for k categories, k different FBPNs with the same topology will be used.

2.1.2 Output Time Prediction Within Each Category with FBPN

The configuration of the FBPN is established as follows:

1. Inputs: eight parameters associated with the n-th example/lot including U_n, Q_n, BQ_n, FQ_n, WIP_n, and $D_n^{(i)}$ (i = 1~3). These parameters have to be normalized so that their values fall within [0, 1]. Then some production execution/control experts are requested to express their beliefs (in linguistic terms) about the importance of each input parameter in predicting the cycle (output) time of a wafer lot. Linguistic assessments for an input parameter are converted into several pre-specified fuzzy numbers. The subjective importance of an input parameter is then obtained by averaging the corresponding fuzzy numbers of the linguistic replies for the input parameter by all experts. The subjective importance obtained for an input parameter is multiplied to the normalized value of the input parameter. After such a treatment, all inputs to the FBPN become triangular fuzzy numbers, and the fuzzy arithmetic for triangular fuzzy numbers is applied to deal with all calculations involved in training the FBPN.
2. Single hidden layer: Generally one or two hidden layers are more beneficial for the convergence property of the network.
3. Number of neurons in the hidden layer: the same as that in the input layer. Such a treatment has been adopted by many studies (e.g. [2, 5]).
4. Output: the (normalized) cycle time forecast of the example. The output is chosen according to the purpose. Besides, the same output has been adopted in many previous studies (e.g. [2-5]).
5. Transfer function: Delta rule.
6. Network learning rule: Back propagation (BP) rule.
7. Transformation function: Sigmoid function,

$$f(x) = 1/(1+e^{-x}). \tag{1}$$

8. Learning rate (η): 0.01~1.0.
9. Batch learning.

The procedure for determining the parameter values is now described. A portion of the examples is fed as "training examples" into the FBPN to determine the parameter values. Two phases are involved at the training stage. At first, in the forward phase, inputs are multiplied with weights, summed, and transferred to the hidden layer. Then activated signals are output from the hidden layer as:

$$\tilde{h}_j = (h_{j1},\ h_{j2},\ h_{j3}) = 1/1+e^{-\tilde{n}_j^h}\ , \tag{2}$$

where

$$\tilde{n}_j^h = (n_{j1}^h,\ n_{j2}^h,\ n_{j3}^h) = \tilde{I}_j^h(-)\tilde{\theta}_j^h\ , \tag{3}$$

$$\tilde{I}_j^h = (I_{j1}^h,\ I_{j2}^h,\ I_{j3}^h) = \sum_{all\ i} \tilde{w}_{ij}^h(\times)\tilde{x}_{(i)}\ , \tag{4}$$

where (−) and (×) denote fuzzy subtraction and multiplication, respectively; $\tilde{h}_j$'s are also transferred to the output layer with the same procedure. Finally, the output of the FBPN is generated as:

$$\tilde{o} = (o_1,\ o_2,\ o_3) = 1/1+e^{-\tilde{n}^o}\ , \tag{5}$$

where

$$\tilde{n}^o = (n_1^o,\ n_2^o,\ n_3^o) = \tilde{I}^o(-)\tilde{\theta}^o\ , \tag{6}$$

$$\tilde{I}^o = (I_1^o, I_2^o, I_3^o) = \sum_{all\ j} \tilde{w}_j^o(\times)\tilde{h}_j\ . \tag{7}$$

To improve the practical applicability of the FBPN and to facilitate the comparisons with conventional techniques, the fuzzy-valued output $\tilde{o}$ is defuzzified according to the centroid-of-area (COA) formula:

$$o = \text{COA}(\tilde{o}) = (o_1 + 2o_2 + o_3)/4. \tag{8}$$

Then the defuzzified output o is applied to predict the actual cycle time a, for which prediction error E and RMSE is calculated:

$$E = |o - a|. \tag{9}$$

$$RMSE = \sqrt{\sum (o-a)^2 / \text{number of examples}}. \tag{10}$$

Note that the prediction error is fed back to the SOM classifier to adjust the classification result.

Subsequently in the backward phase, the deviation between o and a is propagated backward, and the error terms of neurons in the output and hidden layers can be calculated, respectively, as

$$\delta^o = o(1-o)(a-o)\,, \tag{11}$$

$$\tilde{\delta}_j^h = (\delta_{j1}^h,\ \delta_{j2}^h,\ \delta_{j3}^h) = \tilde{h}_j(\times)(1-\tilde{h}_j)(\times)\tilde{w}_j^o\delta^o\,. \tag{12}$$

Based on them, adjustments that should be made to the connection weights and thresholds can be obtained as

$$\Delta\tilde{w}_j^o = (\Delta w_{j1}^o,\ \Delta w_{j2}^o,\ \Delta w_{j3}^o) = \eta\delta^o\tilde{h}_j\,, \tag{13}$$

$$\Delta\tilde{w}_{ij}^h = (\Delta w_{ij1}^h,\ \Delta w_{ij2}^h,\ \Delta w_{ij3}^h) = \eta\tilde{\delta}_j^h(\times)\tilde{x}_i\,, \tag{14}$$

$$\Delta\theta^o = -\eta\delta^o\,, \tag{15}$$

$$\Delta\tilde{\theta}_j^h = (\Delta\theta_{j1}^h, \Delta\theta_{j2}^h, \Delta\theta_{j3}^h) = -\eta\tilde{\delta}_j^h\,. \tag{16}$$

Theoretically, network-learning stops when RMSE falls below a pre-specified level, or the improvement in RMSE becomes negligible with more epochs, or a large number of epochs have already been run. In addition, to avoid the accumulation of fuzziness during the training process, the lower and upper bounds of all fuzzy numbers in the FBPN will no longer be modified if Chen's index [5] converges to a minimal value. Then test examples are fed into the FBPN to evaluate the accuracy of the network that is also measured with the RMSE. Finally, the FBPN can be applied to predicting the cycle time of a new lot. When a new lot is released into the fab, the eight parameters associated with the new lot are recorded and fed as inputs to the FBPN. After propagation, the network output determines the output time forecast of the new lot.

2.2 Achievability Evaluation with Fuzzy Inference Rules

The "achievability" of an output time forecast is defined as the probability that the fabrication on the wafer lot can be finished in time before the output time forecast [7]. Theoretically, if a probability distribution can be obtained for the output time forecast, then achievability can be evaluated with the cumulative probability of the probability distribution before the given date. However, there are many managerial actions (e.g. elevating the priority of the wafer lot, lowering the priority of another wafer lot, inserting emergency lots, adding allowance, etc.) that are more influential to achievability. Considering the effects of these actions, the evaluation of achievability is decomposed into the following two assessments: the possible forwardness of the output time forecast if the priority is elevated, and the easiness of priority elevation. For combining the two assessments, the fuzzy AND operator is applied:

1. Vary the priority of the wafer lot (from the current level to every higher level), and then predict the output time of the wafer lot again with the hybrid ANN.
2. Calculate the forwardness (represented with *FWD*) of the output time after priority elevation, and then classify the result into one of the following five categories: "Insignificant (I)", "Somewhat Insignificant (SI)", "Moderate (M)", "Somewhat Significant (SS)", and "Significant (S)". Apply Ishibuchi's simple fuzzy partition [17] in forming the five categories.
3. Request experts to evaluate the easiness of priority elevation (represented with *EAS*), and then classify the result into one of the following five categories: "Very Easy (VE)", "Easy (E)", "Moderate (M)", "Difficult (D)", and "Very Difficult (VD)". Usually the percentages of lots with various priorities in a wafer fab are controlled. The easiness of priority elevation is determined against such targets.
4. Apply the fuzzy AND operator to combine the two assessments, according to the fuzzy inference rules proposed by Chen [7], so as to determine the achievability of the original output time forecast which is represented with linguistic terms including "Very Low (VL)", "Low (L)", "Medium (M)", "High (H)", and "Very High (VH)".

3 Simulation Experiments

In practical situations, the history data of each lot is only partially available in the fab. Further, some information of the previous lots such as Q_n, BQ_n, and FQ_n is not easy to collect on the shop floor. Therefore, a simulation model is often built to simulate the manufacturing process of a real wafer fab [1-14]. Then, such information can be derived from the shop floor status collected from the simulation model [3]. To generate a demonstrative example, a simulation program coded using Microsoft Visual Basic .NET is constructed to simulate a wafer fab with the following assumptions:

1. Lots are uniformly distributed into the plant.
2. The distributions of the interarrival times of machine downs are exponential.
3. The distribution of the time required to repair a machine is uniform.
4. The percentages of lots with different product types in the fab are predetermined. As a result, this study is only focused on fixed-product-mix cases. However, the product mix in the simulated fab does fluctuate and is only approximately fixed in the long term.
5. The percentages of lots with different priorities released into the fab are controlled.
6. The priority of a lot cannot be changed during fabrication.
7. Lots are sequenced on each machine first by their priorities, then by the first-in-first-out (FIFO) policy. Such a sequencing policy is a common practice in many foundry fabs.
8. A lot has equal chances to be processed on each alternative machine/head available at a step.
9. A lot cannot proceed to the next step until the fabrication on its every wafer has been finished. No preemption is allowed.

The basic configuration of the simulated wafer fab is the same as a real-world wafer fab which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. A trace report was generated every simulation run for verifying the simulation model. The simulated average cycle times have also been compared with the actual values to validate the simulation model. Assumptions (1)~(3), and (7)~(9) are commonly adopted in related studies (e.g. [2-5]), while assumptions (4)~(6) are made to simplify the situation. There are five products (labeled as A~E) in the simulated fab. A fixed product mix is assumed. The percentages of these products in the fab's product mix are assumed to be 35%, 24%, 17%, 15%, and 9%, respectively. The simulated fab has a monthly capacity of 20,000 pieces of wafers and is expected to be fully utilized (utilization = 100%). Lots are of a standard size of 24 wafers per lot. Lots are released one by one every 0.85 hours. Three types of priorities (normal, hot, and super hot) are randomly assigned to lots. The percentages of lots with these priorities released into the fab are restricted to be approximately 60%, 30%, and 10%, respectively. Each product has 150~200 steps and 6~9 reentrances to the most bottleneck machine. The singular production characteristic "reentry" of the semiconductor industry is clearly reflected in the example. It also shows the difficulty for the production planning and scheduling people to provide an accurate due-date for the product with such a complicated routing. Totally 102 machines (including alternative machines) are provided to process single-wafer or batch operations in the fab. Thirty replicates of the simulation are successively run. The time required for each simulation replication is about 12 minute on a PC with 512MB RAM and Athlon™ 64 Processor 3000+ CPU. A horizon of 24 months is simulated. The maximal cycle time is less than 3 months. Therefore, four months and an initial WIP status (obtained from a pilot simulation run) seemed to be sufficient to drive the simulation into a steady state. The statistical data were collected starting at the end of the fourth month. For each replicate, data of 30 lots are collected and classified by their product types and priorities. Totally, data of 900 lots can be collected as training and testing examples. Among them, 2/3 (600 lots, including all product types and priorities) are used to train the network, and the other 1/3 (300 lots) are reserved for testing.

3.1 Results and Discussions

The first part of the proposed methodology is to apply a hybrid and recurrent ANN (SOM-FBPN) to predicting the output time for every lot in the wafer fab. In the demonstrative example, the SOM-FBPN and three traditional approaches (FBPN, CBR, and EFR) were all applied for comparison to five test cases containing the data of full-size (24 wafers per lot) lots with different product types and priorities. The minimal RMSEs achieved by applying these approaches to different cases were recorded and compared in Table 1. The convergence condition was established as either the improvement in RMSE becomes less than 0.001 with one more epoch, or 1000 epochs have already been run. According to experimental results, the following discussions are made:

1. From the effectiveness viewpoint, the prediction accuracy (measured with the RMSE) of the hybrid SOM-FBPN approach was significantly better than those of the other approaches in most cases by achieving a 15%~45% (and an average of 31%) reduction in RMSE over the comparison basis – the FBPN. There is only

one exception, A (super hot lots), in which the RMSE of the hybrid SOM-FBPN approach was 5% worse than that of EFR. Overall, the prediction accuracy of the hybrid SOM-FBPN approach was still better than that of EFR. The average advantage is 3%.

2. In the case that the lot priority was the highest (super hot lot), the hybrid approach has the greatest advantage over FBPN in forecasting accuracy. In fact, the cycle time variation of super hot lots is the smallest, which makes their cycle times easy to predict. Clustering such lots seems to provide the most significant effect on the performance of cycle time prediction.
3. As the lot priority increases, the superiority of the hybrid SOM-FBPN approach over FBPN becomes more evident.
4. The greatest superiority of the hybrid SOM-FBPN approach over EFR happens when the lot priority is the smallest (normal lots).

Table 1. Comparisons of the RMSEs of various approaches

RMSE	A(normal)	A(hot)	A(super hot)	B(normal)	B(hot)
FBPN	177.1	102.27	12.23	286.93	75.98
CBR	172.44(-3%)	86.66(-15%)	11.59(-5%)	295.51(+3%)	78.85(+5%)
EFR	164.29(-7%)	66.21(-35%)	9.07(-26%)	208.28(-27%)	44.57(-41%)
SOM-FBPN	151.34(-15%)	63.66(-38%)	9.72(-21%)	188.55(-34%)	41.43(-45%)

The second part of the hybrid system is a set of FIR applied to evaluating the achievability of an output time forecast. Some results are shown in Table 2.

Table 2. Some results of achievability evaluation

Lot number	Priority elevation	*FWD*	*EAS*	Achievability
PA01	Normal → Hot	-11.8% (SI)	D	L
PA01	Normal → Super Hot	-20% (M)	VD	VL
PB01	Hot → Super Hot	-9.6% (SI)	D	L
PA02	Super hot	-	VD	-

4 Conclusions and Directions for Future Research

A hybrid ANN-FIR system is constructed in this study for lot output time prediction and achievability evaluation in a wafer fab. At first, a hybrid and recurrent ANN (SOM and FBPN) is proposed to predict the output time of a wafer lot. Subsequently, a set of FIRs are established to evaluate the achievability of an output time forecast. According to experimental results,

1. From the effectiveness viewpoint, the prediction accuracy of the proposed hybrid ANN was significantly better than those of some traditional approaches.
2. Achievability has not been investigated in traditional wafer lot output time prediction approaches. With the proposed methodology, both output time prediction and achievability evaluation can be concurrently accomplished.

However, to further evaluate the advantages and disadvantages of the proposed methodology, it has to be applied to a full-scale actual wafer fab in future research.

References

1. Barman, S.: The Impact of Priority Rule Combinations on Lateness and Tardiness. IIE Transactions 30 (1998) 495-504
2. Chang, P.-C., Hsieh, J.-C.: A Neural Networks Approach for Due-date Assignment in a Wafer Fabrication Factory. International Journal of Industrial Engineering 10(1) (2003) 55-61
3. Chang, P.-C., Hsieh, J.-C., Liao, T. W.: A Case-based Reasoning Approach for Due Date Assignment in a Wafer Fabrication Factory. In: Proceedings of the International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, British Columbia, Canada (2001)
4. Chang, P.-C., Hsieh, J.-C., Liao, T. W.: Evolving Fuzzy Rules for Due-date Assignment Problem in Semiconductor Manufacturing Factory. Journal of Intelligent Manufacturing 16 (2005) 549-557
5. Chen, T.: A Fuzzy Back Propagation Network for Output Time Prediction in a Wafer Fab. Applied Soft Computing 2/3F (2003) 211-222
6. Chen, T.: A Fuzzy Set Approach for Evaluating the Achievability of an Output Time Forecast in a Wafer Fabrication Plant. Lecture Notes in Artificial Intelligence 4293 (2006) 483-493
7. Chen, T.: A Hybrid Look-ahead SOM-FBPN and FIR System for Wafer-lot-output Time Prediction and Achievability Evaluation. International Journal of Advanced Manufacturing Technology (2007) DOI 10.1007/s00170-006-0741-x
8. Chen, T.: A Hybrid SOM-BPN Approach to Lot Output Time Prediction in a Wafer Fab. Neural Processing Letters 24(3) (2006) 271-288
9. Chen, T.: A Look-ahead Fuzzy Back Propagation Network for Lot Output Time Series Prediction in a Wafer Fab. Lecture Notes in Computer Science 4234 (2006) 974-982
10. Chen, T.: Applying an Intelligent Neural System to Predicting Lot Output Time in a Semiconductor Fabrication Factory. Lecture Notes in Computer Science 4234 (2006) 581-588
11. Chen, T., Jeang, A., Wang, Y.-C.: A Hybrid Neural Network and Selective Allowance Approach for Internal Due Date Assignment in a Wafer Fabrication Plant. International Journal of Advanced Manufacturing Technology (2007) DOI 10.1007/s00170-006-0869-8
12. Chen, T., Lin, Y.-C.: A Hybrid and Intelligent System for Predicting Lot Output Time in a Semiconductor Fabrication Factory. Lecture Notes in Artificial Intelligence 4259 (2006) 757-766
13. Chung, S.-H., Yang, M.-H., Cheng, C.-M.: The Design of Due Date Assignment Model and the Determination of Flow Time Control Parameters for the Wafer Fabrication Factories. IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part C 20(4) (1997) 278-287
14. Foster, W. R., Gollopy, F., Ungar, L. H.: Neural Network Forecasting of Short, Noisy Time Series. Computers in Chemical Engineering 16(4) (1992) 293-297
15. Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
16. Hung, Y.-F., Chang, C.-B.: Dispatching Rules Using Flow Time Predictions for Semiconductor Wafer Fabrications. In: Proceedings of the 5th Annual International Conference on Industrial Engineering Theory, Applications and Practice, Taiwan (2001)

17. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed Representation of Fuzzy Rules and Its Application to Pattern Classification. Fuzzy Sets and Systems 52(1) (1992) 21-32
18. Lin, C.-Y.: Shop Floor Scheduling of Semiconductor Wafer Fabrication Using Real-time Feedback Control and Prediction. Ph.D. Dissertation, Engineering-Industrial Engineering and Operations Research, University of California at Berkeley (1996)
19. Piramuthu, S.: Theory and Methodology – Financial Credit-risk Evaluation with Neural and Neuralfuzzy Systems. European Journal of Operational Research 112 (1991) 310-321
20. Ragatz, G. L., Mabert, V. A.: A Simulation Analysis of Due Date Assignment. Journal of Operations Management 5 (1984) 27-39
21. Vig, M. M., Dooley, K. J.: Dynamic Rules for Due-date Assignment. International Journal of Production Research 29(7) (1991) 1361-1377
22. Wang, L.-X., Mendel, J. M.: Generating Fuzzy Rules by Learning from Examples. IEEE Transactions on Systems, Man, and Cybernetics 22(6) (1992) 1414-1427
23. Weeks, J. K.: A Simulation Study of Predictable Due-dates. Management Science 25 (1979) 363–373

M-Factor High Order Fuzzy Time Series Forecasting for Road Accident Data

Tahseen Ahmed Jilani and Syed Muhammad Aqil Burney

Department of Computer Science, University of Karachi, Karachi-74270, Pakistan
tahseenjilani@ieee.org, Burney@copmuter.org

Abstract. In this paper, we have presented new multivariate fuzzy time series (FTS) forecasting method. This method assume m-factors with one main factor of interest. Stochastic fuzzy dependence of order k is assumed to define general method of multivariate FTS forecasting and control. This new method is applied for forecasting total number of car road accidents causalities in Belgium using four secondary factors. Practically, in most of the situations, actuaries are interested in analysis of the patterns of casualties in road accidents. Such type of analysis supports in deciding approximate risk classification and forecasting for each area of a city. This directly affects the underwriting process and adjustment of insurance premium, based on risk intensity for each area. Thus, this work provides support in deciding the appropriate risk associated with an insured in a particular area. Finally, comparison is also made with most recent available work on fuzzy time series forecasting.

Keywords: Average forecasting error rate (AFER), Fuzziness of fuzzy sets, Fuzzy If-Then rules, Multivariate fuzzy time series, fuzzy aggregation operators.

1 Introduction

In our daily life, people often use forecasting techniques to model and predict economy, population growth, stock, insurance/ re-insurance, portfolio analysis and etc. However, in the real world, an event can be affected by many factors. Therefore, if we consider more factors for prediction, with higher complexity then we can get better forecasting results.

During last few decades, various approaches have been developed for time series forecasting. Among them ARMA models and Box-Jenkins model building approaches are highly famous. Most of the financial and economic time series are nonlinear and thus the linear models are inadequate to handle nonlinearity present in the process.

In recent years, many researchers used FTS to handle prediction problems. Song and Chissom [9] presented the concept of fuzzy time series based on the concepts of fuzzy set theory to forecast the historical enrollments of the University of Alabama. Huarng [2] presented the definition of two kinds of intervals in the universe of discourse to forecast the TAIFEX .Chen [1] presented a method for forecasting based on high-order fuzzy time series. Melike et. al. [7] proposed forecasting method using first order fuzzy time series. Lee et. al. [6] presented handling of forecasting problems using two-factor high order fuzzy time series for TAIFEX and daily temperature in Taipei, Taiwan.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 246–254, 2007.
springerlink.com

The rest of the work on these new methods is organized as follows. In section 2, we present the new methods for FTS modeling and applied to annual car accident causalities data. We also make indirect comparison of our proposed method with existing methods. The conclusions are discussed in section 3.

2 New Forecasting Method Based on M-Factors High-Order Fuzzy Time Series

From National Institute of Statistics, Belgium, we have taken the data for the period of 1974-2004. The main factor of interest is the yearly road accident causalities and secondary factors are mortally wounded, died within one month, severely wounded and light casualties.

For forecasting purpose, we can define relationship among present and future states of a time series with the help of fuzzy sets. Assume the fuzzified data of the $i\,th$ and $(i+1)th$ day are A_j and A_k, respectively, where $A_j, A_k \in U$, then $A_j \rightarrow A_k$ represent the fuzzy logical relationship between A_j and A_k.

Let $F(t)$ be a FTS. If $F(t)$ is caused by $(F_1(t-1), F_2(t-1))$, $(F_1(t-2), F_2(t-2))$,..., $(F_1(t-n), F_2(t-n))$, then this fuzzy logical relationship is represented by

$$(F_1(t-n), F_2(t-n)),...,(F_1(t-2), F_2(t-2)),(F_1(t-1), F_2(t-1)) \rightarrow F(t) \tag{1}$$

is called the two-factors nth order FTS forecasting model, where $F_1(t)$ and $F_2(t)$ are called the main factor and the secondary factor FTS respectively. In the similar way, we can define m-factor nth order fuzzy logical relationship as

$$\begin{aligned}&(F_1(t-n), F_2(t-n),..., F_m(t-n)),...,(F_1(t-2), F_2(t-2),..., F_m(t-2)),\\&(F_1(t-1), F_2(t-1),..., F_m(t-1)) \rightarrow F(t)\end{aligned} \tag{2}$$

Here $F_1(t)$ is called the main factor and $F_2(t), F_3(t),..., F_m(t)$ are called secondary factor FTS. Using fuzzy composition rules, we establish a fuzzy inference system for FTS forecasting with higher accuracy. The accuracy of forecast can be improved by considering higher number of factors and higher dependence on history. Now we present an extended method for handling forecasting problems based on m-factors high-order FTS. The proposed method is now presented as follows.

Step 1. Define the universe of discourse U of the main factor $U = [D_{\min} - D_1, D_{\max} - D_2]$, where $D_{\min}$ and $D_{\max}$ are the minimum and the maximum values of the main factor of the known historical data respectively, and D_1, D_2 are two proper positive real numbers to divide the universe of discourse into n equal length intervals $u_1, u_{2,} ..., u_l$. Define the universes of discourse $\mathbf{V}_i$, $i = 1, 2,..., m-1$

of the secondary factors $\mathbf{V}_i = \left[(\mathbf{E_i})_{\min} - \mathbf{E}_{i1}, (\mathbf{E_i})_{\max} - \mathbf{E}_{i2} \right]$, where $(\mathbf{E_i})_{\min} = \left[(E_1)_{\min}, (E_2)_{\min}, ..., (E_m)_{\min} \right]$ and $(\mathbf{E_i})_{\max} = \left[(E_1)_{\max}, (E_2)_{\max}, ..., (E_m)_{\max} \right]$ are the minimum and maximum values of the secondary-factors of the known historical data respectively and $\mathbf{E_{i1}}$, $\mathbf{E_{i2}}$ are vectors of proper positive numbers to divide each of the universe of discourse $\mathbf{V}_i$, $i = 1, 2, ..., m-1$ into equal length intervals termed as $\mathbf{v}_{1,l}, \mathbf{v}_{2,l}, ..., \mathbf{v}_{m-1,l}$, $p = 1, 2, ..., l$, where $\mathbf{v}_{1,l} = \left[\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, ...\mathbf{v}_{1,p} \right]$ represents p-intervals of equal length of universe of discourse $\mathbf{V_1}$ for first secondary-factor FTS. Thus we have $(m-1) \times l$ matrix of intervals for secondary-factors given by,

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{V}_{m-1} \end{bmatrix}^T = \begin{bmatrix} \mathbf{v}_{1,1} & \mathbf{v}_{1,2} & \cdot & \cdot & \cdot & \mathbf{v}_{1,l} \\ \mathbf{v}_{2,1} & \mathbf{v}_{2,2} & \cdot & \cdot & \cdot & \mathbf{v}_{2,l} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{v}_{m-1,1} & \mathbf{v}_{m-1,1} & \cdot & \cdot & \cdot & \mathbf{v}_{m-1,l} \end{bmatrix}^T \tag{3}$$

where $\mathbf{v}_{i,j}$, $i = 1, 2, ...m-1$, $p = 1, 2, ..., l$.

Based on annual car accident causalities, we assume that $D_{\min}$=953 and $D_{\max}$=1644, thus for main factor time series we get $U = [850, 1650]$. Similarly for secondary factors Y_1, Y_2, Y_3 *and* Y_4, we assumed $\mathbf{E}_{\min} = [90, 1094, 5949, 38390]$ and $\mathbf{E}_{\max} = [819, 2393, 16645, 46818]$, to get secondary factors' FTSs.

Step 2. Define the linguistic term A_i represented by fuzzy sets of the main factor shown as follows:

$$\begin{aligned}
A_1 &= {}^{1}/_{u_1} + {}^{0.5}/_{u_2} + {}^{0}/_{u_3} + {}^{0}/_{u_4} + {}^{0}/_{u_5} + ... + {}^{0}/_{u_{l-2}} + {}^{0}/_{u_{l-1}} + {}^{0}/_{u_l} \\
A_2 &= {}^{0.5}/_{u_1} + {}^{1}/_{u_2} + {}^{0.5}/_{u_3} + {}^{0}/_{u_4} + {}^{0}/_{u_5} + ... + {}^{0}/_{u_{l-2}} + {}^{0}/_{u_{l-1}} + {}^{0}/_{u_l} \\
A_3 &= {}^{0}/_{u_1} + {}^{0.5}/_{u_2} + {}^{1}/_{u_3} + {}^{0.5}/_{u_4} + {}^{0}/_{u_5} + ... + {}^{0}/_{u_{l-2}} + {}^{0}/_{u_{l-1}} + {}^{0}/_{u_l} \\
&\cdot \\
&\cdot \\
&\cdot \\
A_n &= {}^{0}/_{u_1} + {}^{0}/_{u_2} + {}^{0}/_{u_3} + {}^{0}/_{u_4} + {}^{0}/_{u_5} + ... + {}^{0}/_{u_{l-2}} + {}^{0.5}/_{u_{l-1}} + {}^{1}/_{u_l}
\end{aligned} \tag{4}$$

where $A_1, A_2, \ldots, A_n$ are linguistic terms to describe the values of the main factor. Now define the linguistic term $B_{i,j}, i = 1, 2, \ldots, m-1,\ \ j = 1, 2, \ldots, n$ represented by fuzzy sets of the secondary-factors, as shown below:

$$\begin{aligned}
B_{i,1} &= 1/\mathbf{v}_{i,1} + 0.5/\mathbf{v}_{i,2} + 0/\mathbf{v}_{i,3} + 0/\mathbf{v}_{i,4} + 0/\mathbf{v}_{i,5} + \ldots + 0/\mathbf{v}_{i,l-2} + 0/\mathbf{v}_{i,l-1} + 0/\mathbf{v}_{i,l} \\
B_{i,2} &= 0.5/\mathbf{v}_{i,1} + 1/\mathbf{v}_{i,2} + 0.5/\mathbf{v}_{i,3} + 0/\mathbf{v}_{i,4} + 0/\mathbf{v}_{i,5} + \ldots + 0/\mathbf{v}_{i,l-2} + 0/\mathbf{v}_{i,l-1} + 0/\mathbf{v}_{i,l} \\
B_{i,3} &= 0/\mathbf{v}_{i,1} + 0.5/\mathbf{v}_{i,2} + 1/\mathbf{v}_{i,3} + 0.5/\mathbf{v}_{i,4} + 0/\mathbf{v}_{i,5} + \ldots + 0/\mathbf{v}_{i,l-2} + 0/\mathbf{v}_{i,l-1} + 0/\mathbf{v}_{i,l} \\
&\ \ \vdots \\
B_{i,n} &= 0/\mathbf{v}_{i,1} + 0/\mathbf{v}_{i,2} + 0/\mathbf{v}_{i,3} + 0/\mathbf{v}_{i,4} + 0/\mathbf{v}_{i,5} + \ldots + 0/\mathbf{v}_{i,l-2} + 0.5/\mathbf{v}_{i,l-1} + 1/\mathbf{v}_{i,l}
\end{aligned} \tag{5}$$

Step 3. Fuzzify the historical data described as follows. Find out the interval u_p , $p = 1, 2, \ldots, l$ to which the value of the main factor belongs.

Case 1. If the value of the main factor belongs to u_1 , then the value of the main factor is fuzzified into $1/A_1 + 0.5/A_2 + 0.0/A_3$, denoted by X_1

Case 2. If the value of the main factor belongs to u_l, $p = 2, 3, \ldots, l-1$ then the value of the main factor is fuzzified into $0.5/A_{i-1} + 1/A_i + 0.5/A_{i+1}$, denoted by X_i .

Case 3. If the value of the main factor belongs to u_p , then the value of the main factor is fuzzified into $0/A_{n-2} + 0.5/A_{n-1} + 1/A_n$, denoted by X_n .

Now, for ith secondary-factor, find out the interval $\mathbf{V}_{i,l}$ to which the value of the secondary-factor belongs.

Case 1. If the value of the ith secondary-factor belongs to $\mathbf{v}_{i,1}$, then the value of the secondary-factor is fuzzified into $1/B_{i,1} + 0.5/B_{i,2} + 0.0/B_{i,3}$, denoted by $\mathbf{Y}_{i,1} = \left[Y_{1,1}, Y_{2,1}, \ldots, Y_{m-1,1}\right]$.

Case 2. If the value of the ith secondary-factor belongs to $\mathbf{v}_{i,l}, p = 2, 3, .., l-1$, then the value of the ith secondary-factor is fuzzified into $0.5/B_{i,j-1} + 1/B_{i,j} + 0.5/B_{i,j+1}$, $j = i = 2, 3, \ldots, n-1$ denoted by $Y_{i,j}$, where $j = 2, 3, \ldots., n-1$.

Case 3. If the value of the ith secondary-factor belongs to $\mathbf{v}_{i,p}$, then the value of the secondary-factor is fuzzified into $0.0/B_{i,n-2} + 0.5/B_{i,n-1} + 1/B_{i,n}$, denoted by $Y_{i,n}$.

Step 4. Get the m-factors kth-order fuzzy logical relationships based on the fuzzified main and secondary factors from the fuzzified historical data obtained in Step 3)**.** If the fuzzified historical data of the main-factor of ith day is X_i, then construct the m-factors kth-order fuzzy logical relationship as given in (6).

$$\begin{matrix}\left(X_{j-k};Y_{2,j-k},...,Y_{m-1,j-k}\right),...,\left(X_{j-2};Y_{2,j-2},...,Y_{m-1,j-2}\right), \\ \left(X_{j-1};Y_{1,j-1},Y_{2,j-1},...,Y_{m-1,j-1}\right)\rightarrow X_j\end{matrix} \quad \text{, where } j>k \tag{6}$$

or simply,

$$\begin{bmatrix} X_{j-1} & X_{j-2} & . & . & . & X_{j-k} \\ Y_{1,j-1} & Y_{1,j-2} & . & . & . & Y_{1,j-k} \\ Y_{2,j-1} & Y_{2,j-2} & . & . & . & Y_{2,j-k} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ Y_{m-1,j-1} & Y_{m-1,j-2} & . & . & . & Y_{m-1,j-k} \end{bmatrix}^T \rightarrow X_j \quad \text{, where } j>k \tag{7}$$

where X_{j-k} shows the k-step dependence of jth value of main factor X_j, $Y_{i,j-k}$ $, i=1,2,...,m-1,\ j=1,2,...,k;\ k\subseteq n$. Then, divide the derived fuzzy logical relationships into fuzzy logical relationship groups based on the current states of the fuzzy logical relationships. The secondary factors acts like secondary component in the m-dimensional state vector and is used in Step 5).

Step 5. Calculate the forecasted values based on the following principles.

For m-factor kth order fuzzy logical relationship in Eq. (7), the forecasted value of ith day is calculated using fuzzy aggregation operators along with related properties where necessary.

$$t_j^{\alpha}=\left(\frac{\sum\limits_{j-1}^{j+1} w}{\frac{w_{j-1}}{a_{j-1}^{\alpha}}+\frac{w_j}{a_j^{\alpha}}+\frac{w_{j+1}}{a_{j+1}^{\alpha}}}\right)^{1/\alpha} \quad \text{where } 0<\alpha_i\leq 1 \tag{8}$$

$$\alpha_j=\left|\sum_{k=1}^{3}\left(A_{j-k}-t_{j-k}^{\alpha_k}\right)\right| \Big/ 1000 \quad \text{where } 0<\alpha_i\leq 1 \tag{9}$$

Here A_{j-k} and $t_{j-k}^{\alpha_k}$ are actual and forecasted causalities for the last three years respectively. Initially, we have assumed first three forecasting values using the formula defined by [6]. We have assumed eight partitions of the universe of discourse of the main factor FTS. Assuming that $0 \neq A_i, \forall A_i, i = 1,2,3,...,8$. If $(x_j, y_{i,j}) \to x_{j+1}$; $i = 1,2,...,n$; $j = 1,2,...k$, then the proposed method satisfies the following axioms

Axiom 1: $t(\underline{0}) = 0$ *and* $t(\underline{1}) = 1$ (Boundary Condition)

Axiom 2: $t^{\alpha}(a_1, a_2, ..., a_n) \le t^{\alpha_i}(b_1, b_2, ..., b_n)$ provided $a_i \le b_i, i = 1,2,...,n$ (Monotonicity)

Axiom 3: $t^{\alpha}(a_1, a_2, ..., a_n)$ is continuous (Continuity)

Axiom 4: $(A_k)_{\min} \le t_{\alpha}(a) \le (A_k)_{\max}$; $k = 1,2,....,8$ (Symmetry)

Axiom 5: $t^{\alpha}(a, a, ..., a) = a \ \ \forall a \in [0,1]$ (Idempotency)

For third order m-factor FTS forecasting, we can define Eq (10) using Eq. (8). In Eq. (10) below, the negative suffix indicates time back data points and $a_{p-1,}$ a_p and a_{p+1} are the midpoints of the intervals $u_{p-1,} u_p$ and u_{p+1} respectively. Above defined forecasting formula fulfills the axioms of fuzzy sets like monotonicity, boundary conditions, continuity and idempotency. The performance is measured by average forecasted error rate (AFER) defined in Eq (11).

$$t_j^{\alpha}(a) = \begin{cases} \left(\dfrac{1 + 0.5}{\dfrac{1}{(a_1)^{\alpha}} + \dfrac{0.5}{(a_2)^{\alpha}}} \right)^{1/\alpha} & ; \ for \ \ j = 1 \\ \left(\dfrac{0.5 + 1 + 0.5}{\dfrac{0.5}{(a_{j-1})^{\alpha}} + \dfrac{1}{(a_j)^{\alpha}} + \dfrac{0.5}{(a_{j+1})^{\alpha}}} \right)^{1/\alpha} & ; \ for \ \ 2 \le j \le n-1 \\ \left(\dfrac{0.5 + 1}{\dfrac{0.5}{(a_{n-1})^{\alpha}} + \dfrac{1}{(a_n)^{\alpha}}} \right)^{1/\alpha} & ; \ for \ \ j = n \end{cases} \tag{10}$$

$$AFER = \frac{\sum_{i=1}^{n} \left| (Forecasted\ value\ of\ Day\ j - Actual Value\ of\ Day\ j) \right|}{n} \times 100\% \tag{11}$$

In Table 1, forecasted annual road accident casualties (F_i) are given along with AFER value. Also a comparison is made between our proposed methods and [6]. Clearly our proposed method performs very well as compared to [6].

Table 1. Proposed Fuzzy Time Series Forecasting Method and Lee et. al.(2006) in [6]

Year	Actual Causalities A_i	Proposed Method-I			Lee L. W. et. al. (2006)	
		F_i	alpha	$\left\lvert\frac{F_i - A_i}{A_i}\right\rvert$	Fi	$\left\lvert\frac{F_i - A_i}{A_i}\right\rvert$
1974	1574	--	--	--	--	--
1975	1460	--	--	--	--	--
1976	1536	--	--	--	--	--
1977	1597	1500	--	--	1500	--
1978	1644	1500	--	--	1500	--
1979	1572	1500	--	--	1500	--
1980	1616	1598	0.0069	0.0112	1500	0.0718
1981	1564	1598	0.0715	0.0218	1500	0.0409
1982	1464	1498	0.0950	0.0234	1500	0.0246
1983	1479	1498	0.0468	0.013	1500	0.0142
1984	1369	1398	0.0026	0.0212	1500	0.0957
1985	1308	1298	0.0698	0.0077	1400	0.0703
1986	1456	1498	0.0620	0.029	1300	0.1071
1987	1390	1398	0.0009	0.0058	1500	0.0791
1988	1432	1398	0.0155	0.0236	1400	0.0223
1989	1488	1498	0.0165	0.0069	1400	0.0591
1990	1574	1598	0.0403	0.0155	1500	0.047
1991	1471	1498	0.0612	0.0186	1500	0.0197
1992	1380	1398	0.0382	0.0131	1500	0.087
1993	1346	1298	0.0826	0.0357	1400	0.0401
1994	1415	1398	0.0879	0.0119	1300	0.0813
1995	1228	1198	0.0510	0.0246	1400	0.1401
1996	1122	1098	0.0431	0.0218	1100	0.0196
1997	1150	1198	0.0369	0.0415	1200	0.0435
1998	1224	1198	0.0179	0.0213	1200	0.0196
1999	1173	1198	0.0802	0.0212	1200	0.023
2000	1253	1298	0.0053	0.0359	1300	0.0375
2001	1288	1298	0.0700	0.0078	1300	0.0093
2002	1145	1098	0.0069	0.0415	1100	0.0393
2003	1035	997	0.0715	0.0363	1000	0.0338
2004	953	997	0.0950	0.0465	1000	0.0493
$AFER = \frac{\sum_{i=1}^{n} \frac{\lvert F_i - A_i \rvert}{A_i}}{31} \times 100\% =$				2.0624%		4.7237%

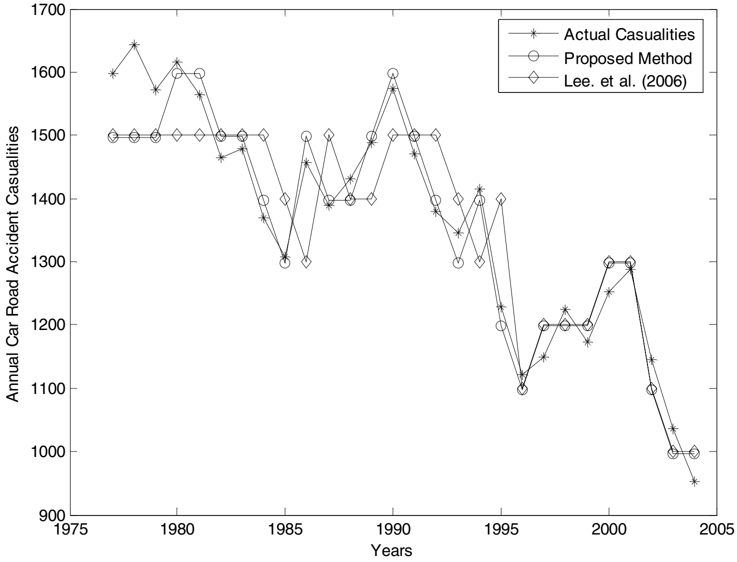


Fig. 1. A Comparison of proposed Method and Lee L. W. et. al. (2006) in [6]

3 Conclusion

From table 1, we can see that our proposed method perform better than [6]. As the work in [6] outperformed the work of [1, 2, 3], therefore, indirectly we can conclude that our proposed method is also better than all of these methods. We have assumed eight partitions for each universe of discourse. Further improved forecasts may be obtained using further partitioning of the universe of discourse. It is also possible to adapt these intervals using some learning paradigms, like supervised artificial neural networks and wavenets [4, 8]. We have considered only triangular MFs in this experiment but further improved results may be obtained by assuming other complex MFs or ensemble of MFs for efficient fuzzification of universe of discourse. Thus our proposed method provides a general class of methods for FTS modeling and forecasting.

Furthermore, we have shown fuzziness of fuzzy observations by presenting each datum of the main series as composed of many fuzzy sets. Thus, FTS modeling extends to type-II FTS modeling. The type-II defuzzified forecasted values (t_j) may also be calculated using some other method, e.g. learning rules from FTS [5, 10, 11].

Acknowledgement

The authors are very thankful to Higher Education Commission of Pakistan. We also thank Institute of Statistics, Belgium for their technical support.

References

[1] Chen, S.M.: Forecasting Enrollments Based on High-Order Fuzzy Time Series. Cybernetic Systems. 33(1) (2002) 1–16

[2] Huarng, K.: Heuristic Models of Fuzzy Time Series for Forecasting. Fuzzy Sets Systems. 123(3) (2001a) 369–386.

[3] Huarng, K.: Effective Lengths of Intervals to Improve Forecasting in Fuzzy Time Series. Fuzzy Sets System. 123(3) (2001b) 387–394.

[4] Ishibuchi, H., Fujioka, R., Tanaka H.: Neural Networks that Learn from Fuzzy If-Then Rules. IEEE Trans. on Fuzzy Systems. 1(1) (1993) 85-97.

[5] Jilani, T.A., Burney S.M.A. and Argil, C.: Multivariate High Order Fuzzy Time Series Forecasting. Trans. Pm Engineering, Computing and Technology. 19 (2007) 288-293.

[6] Klir, G.J., Yuan B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall. India (2005)

[7] Lee, L.W., Wang, L.W., Chen, S.M.: Handling Forecasting Problems Based on Two-Factor High-Order Time Series. IEEE Trans. on Fuzzy Systems. 14(3) (2006) 468-477.

[8] Melike, S., Konstsntin, Y.D.: Forecasting Enrollment Model Based on First-Order Fuzzy Time Series. in proc. Int. Con. on Computational Intelligence, Istanbul, Turkey, 2004.

[9] Park, S., Han, T.: Iterative Inversion of Fuzzified Neural Networks. IEEE Trans. on Fuzzy Systems. 8(3) (2000) 266- 280

[10] Song, Q., Chissom, B.S.: Forecasting Enrollments with Fuzzy Time Series—Part I. Fuzzy Sets and System. 54(1) (1993 a) 1–9

[11] Yager, R.R., Filev, P.P.D.: Essentials of Fuzzy Modeling and Control. John Wiley and Sons (2002)

[12] Zimmerman, H.J.: Fuzzy Set Theory and Its Applications. Kluwer Publishers. Boston MA (2001).

Fuzzy Time Series: A Realistic Method to Forecast Gross Domestic Capital of India

Mrinalini shah

Associate professor, NMIMS University, Mumbai, India
Mrinalini_shah2@rediffmail.com

1 Introduction

In the era of uncertainty and chaos, the decision making is a complex process especially when it is related with the future prediction. The decision making process is fully dependent on the level of accuracy in forecasting. It is obvious that forecasting activities play a vital role in our daily life. The classical time series methods can not deal with forecasting problems in which, the values of time series are linguistic terms represented by fuzzy sets [1, 2]. Therefore, Song and Chissom [4] presented the theory of fuzzy time series to overcome the drawback of the classical time series methods. Time series prediction is a very important practical application with a diverse range of applications including economic and business planning, inventory and production control, weather forecasting, signal processing and control, Pattern matching etc. Based on the theory of fuzzy time series, Song presented some forecasting methods [4,5,6] to forecast the enrollments of the University of Alabama. Chen [8] presented a method to forecast the enrollments of the University of Alabama based on fuzzy time series. It has the advantage of reducing the calculation time and simplifying the calculation process. Hwang [10], used the differences of the enrollments to present a method to forecast the enrollments of the University of Alabama based on fuzzy time series. Huang extended Chen's work presented in [9] and used simplified calculations with the addition of heuristic rules to forecast the enrollments. Chen [12] presented a forecasting method based on high order fuzzy time series for forecasting the enrollments of the University of Alabama. Chen [13] and Hwang presented a method based on fuzzy time series to forecast the temperature.

Song [5] used the following model for forecasting university enrollments:

$$A_i = A_{i-1}{}^{o} R, \tag{1}$$

Where A_i denotes the fuzzified enrollments of year 'i' represented by a fuzzy set, the symbol 'o' denotes the Max-Min composition operator, and R is a fuzzy relation formed by the fuzzified enrollments of the fuzzy time series. This method has few drawbacks i.e.: It requires a large amount of computations to derive the fuzzy relation R. The max-min composition operation of formula (1) takes a large amount of computation time when the fuzzy relation R is very big.

Li [14] presented a dynamic neural network method for time series prediction using the KIII model. Su [15] presented a method for fusing global and local information in predicting time series based on neural networks. Sullivan [16] reviewed the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 255–264, 2007.
springerlink.com

first order time-variant fuzzy time series model and the first order time invariant fuzzy time series model presented by song and Chissom [5], where their models are compared with each other and with a time-variant Markov model using linguistic labels with probability distributions.

However, the forecasting accuracy rates of the existing fuzzy time series methods for forecasting enrollments are not good enough [9]. The method belongs to the first order and time variant methods. It can get a higher forecasting accuracy rate for forecasting data than the existing methods.

2 Basic Concepts for Prediction

Song [3] proposed the definition of fuzzy time series based on fuzzy sets [1]. Let U be the universe of discourse, U={u_1, u_2, …, u_n}, and let A be a fuzzy set in the universe of discourse U defined as follows :

$$A = \mu_A(u_1)/u_1 + \mu_A(u_2)/u_2 + \ldots + \mu_A(u_n)/u_{n,} \tag{2}$$

Where μ_A is the membership function of A, $\mu_A : U \rightarrow [0,1], \mu_A(u_1)$ indicates the grade of membership of u_i in the fuzzy set A. $\mu_A(u_i) \in [0,1], and 1 \le i \le n.$

Let X(t) (t = 1, 2, …,0,1,2, …) be the universe of discourse and be a subset of R, and let fuzzy set $F_i(t)$ (I = 1,2, …) be defined in X(t). Let F(t) is called a fuzzy time series of X(t) (t=…, 0,1,2, …). If F(t) is caused by F(t – 1), denoted by F(t – 1) →F(t), then this relationship can be represented by F(t) = F(t–1)o R(t – 1) [9], where the symbol "o" denotes the Max-Min composition operator; R(t, t-1) is a fuzzy relation between F(t) and F(t-1) and is called the first-order model of F(t).

Let F(t) is a fuzzy time series and let R(t,t-1) be a first order model of F(t). If R(t,t-1)=R(t-1, t-2) for any time t, then F(t) is called a time invariant fuzzy time series. If R(t,t-1) is dependent on time t, that is, R(t, t – 1) may be different from R(t-1, t-2) for any t, then F(t) is called a time variant fuzzy time series.

Song [6] proposed the time-variant fuzzy time-series model and forecasted the enrollments of the University of Alabama based on the model. The method for forecasting the enrollments is briefly reviewed as follows [6]:

Step 1: Define the universe of discourse within which fuzzy sets are defined.
Step 2: Partition the universe of discourse U into several even and equal length intervals.
Step 3: Determine some linguistic values represented by fuzzy sets of the intervals of the universe of discourse.
Step 4: Fuzzify the historical enrollment data.
Step 5: Choose a suitable parameter W, where w> 1, calculate R^w (t, t-1) and fore cast the enrollments as: F(t)=F(t-1)o R^w (t,t-1), (3)

Where F(t) denotes the forecasted fuzzy enrollment of year t, F(t-1) denotes the fuzzified enrollment of year (t-1), and

$$R^{w}(t,t-1)=F^{T}(t-2)\times F(t-1)\cup F^{T}(t-3)\times F(t-2)\bigcup.\bigcup F^{T}(t-w)\times F(t-w+1), \qquad (4)$$

Where w is called the “model basis” denoting the number of years before t, “x” is the Cartesian product operator, and T is the transpose operator

Step 6: Defuzzify the forecasted fuzzy enrollment using neural nets.

Sullivan[16] used the following Markov model to forecast the enrollments of the University of Alabama : $P_{t+1}^{'} = p_{t}^{'} * \mathbf{R}_{m}$ (5)

Where P_t is the vector of state probabilities at time t, P_{t+1} is the vector of state probabilities at time (t + 1), R_m is the transition matrix, and “*” is a conventional matrix multiplication operator. Equation (5) does not change with time and represent time-invariant fuzzy time-series model. The other style of the Markov model is called the time-variant fuzzy time-series model as follows: $P_{1+1}^{'} = p_{1}^{'} * R_{m}^{k}$, k = 1, 2, …, (6) Where R_m^k varies with time.

3 Objective

The classical time series model cannot forecast for problems with linguistically expressed time series. The objective of the study is to prepare a fuzzy time series based model to predict Gross Domestic Capital (GDP) of India.

4 Methodology

The first step is to define the universe of discourse and to partition the universe of discourse into fuzzy interval. The grade of membership is obtained for these fuzzy length interval sets. Then it establishes fuzzy logical relationship among the fuzzy data. Fuzzy inference rules find whether the trend goes upward or downward and predict the capital formation for the coming year. The method takes account of the trend based upon data of (n-1), (n-2) and (n-3) year and thus predicts the value for nth year. The present method is based on second order fuzzy time series.

Define the universe of discourse. According to table 1, the values lie between [12105, 131505]. The universe of discourse in the study is taken as [0, 140000]. Fuzzy time series allows the flexibility to change the universe at any time in future.

Step 1: Dividing the universe of discourse into several even and equal length intervals. For example here U = [12000, 140000] is partitioned into equal length intervals U_1, U_2, U_3, U_4, U_5, U_6 and U_7 where U_1= [0000, 20000), U_2 = (20000, 40000], U_3= [40,000, 60000] U_4=[60000, 80000], U_5=[80000, 100000], U_6=[100000, 120000] U_7=[120000, 140000].

Table 1. Gross domestic Capital Formation: India [T1]

Year	**Public sector (Historical data) In millions rupee**
1980-1981	12105
1981-1982	16986
1982-1983	20139
1983-1984	21265
1984-1985	25600
1985-1986	29990
1986-1987	34772
1987-1988	33757
1988-1989	40136
1989-1990	46405
1990-1991	53099
1991-1992	57633
1992-1993	63997
1993-1994	70834
1994-1995	88206
1995-1996	90977
1996-1997	96187
1997-1998	100653
1998-1999	114545
1999-2000	134484
2000-2001	131505

Table 2. Distribution of Data

0-20000	20000-40000	40,000-60000	60000-80000	80000-100000	10000-120000	120000-140000
2	6	4	2	3	2	2

After following this step the Universe of discourse (0, 140000) is re divided into following fuzzy length interval.

Table 3. Fuzzy length interval

U_l = (0, 20000)	$U_{3.2}$ = (46666.7, 53333.3)
$U_{2.1}$ = (20000, 25000)	$U_{3.3}$ = (53333.3, 60000)
$U_{2.2}$ = (25000, 30000)	U_4 = (60000, 80000)
$U_{2.3}$ = (30000, 35000)	$U_{5.1}$ = (80000, 90000)
$U_{2.4}$ = (35000, 40000)	$U_{5.2}$ = (90000, 100000)
$U_{3.1}$ = (40000, 46666.7)	U_6 = (100000, 120000)
	U_7 = (120000, 140000)

Step 2: After dividing the historical data into the sorted interval from lowest to highest, the statistics of historical data into each interval is given in the table 2. Now convert these equal length intervals into fuzzy length interval. Here for the easiness the interval having the largest number of historical data is divided into four sub intervals of equal length; the interval with the second largest member is divided into three subinterval of equal length. The third largest member interval is divided into two subintervals and the rest of the intervals are with the same length only. If in any interval there is no member from the historical data then that interval will be eliminated. This ensures the accuracy in the prediction.

Step 3: Defining each fuzzy set A_i based on the redivided interval and fuzzifying the historical data {capital formed}, where fuzzy set Ai denotes a linguistic value of the gross domestic capital of India. i.e.

A1 = extremely low, A2 = Very-Very low, A3=Very low, A4=low, A5= slightly low, A6=towards medium, A7= medium, A8 = slightly above medium, A9=above medium, A10=high, A11= Very high, A12=Very-Very High, A13= extremely high.

The grade of membership for these fuzzy length intervals (u_i) have to be decided in fuzzy sets A_i. For simplicity let the MF of fuzzy set Ai are 0, 0.5 or 1. The basic reason of expressing the total capital into fuzzyfied capital is to convert crisp set into fuzzy sets in order to get a fuzzy time series.

$$A_1 = 1/u_1 + 0.5/u_{2,1} + 0/u_{2,2} + 0/u_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_{6,} + 0/u_7$$

$$A_2 = 0.5/u_1 + 1/u_{2,1} + 0.5/u_{2,2} + 0/_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_3 = 0/u_1 + 0.5/u_{2,1} + 1/u_{2,2} + 0.5/_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_4 = 0/u_1 + 0/u_{2,1} + 0.5/u_{2,2} + 1/_{2,3} + 0.5/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_5 = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0.5/{}_{2,3} + 1/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_6 = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0.5/u_{2,4} + 1/u_{3,1} + 0.5/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_7 = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0/u_{2,4} + 0.5/u_{3,1} + 1/u_{3,2} + 0.5/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_8 = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0.5/u_{3,2} + 1/u_{3,3} + 0.5/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_9 = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0.5/u_{3,3} + 1/u_4 + 0.5/u_{5,1} + 0/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_{10} = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0.5/u_4 + 1/u_{5,1} + 0.5/u_{5,2} + 0/u_6 + 0/u_{7,}$$

$$A_{11} = 0/u_1 + 0/u_{1,2} + 0/u_2 + 0/{}_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_{3,4} + 0/u_{4,1} + 0/u_{4,2} + 0.5/u_{4,3} + 1/u_{6,1} + 0.5/u_{6,2} + 0/u_{7,}$$

$$A_{12} = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0.5/u_{5,2} + 1/u_6 + 0.5/u_{7,}$$

$$A_{13} = 0/u_1 + 0/u_{2,1} + 0/u_{2,2} + 0/{}_{2,3} + 0/u_{2,4} + 0/u_{3,1} + 0/u_{3,2} + 0/u_{3,3} + 0/u_4 + 0/u_{5,1} + 0/u_{5,2} + 0.5/u_6 + 1/u_{7,}$$

Step 4: Establish a fuzzy relationship based on the given historical data of the type:

$$\begin{array}{lll} Ai & \rightarrow & Aj \\ Aj & \rightarrow & Ak \\ Ar & \rightarrow & \\ Al & \rightarrow & Am \end{array}$$

If A_i, denotes the fuzzified data for (n-1) year and A_j for nth year and fuzzy relation is $(A_i \rightarrow A_J)$. In a common man's language, this fuzzy logical relationship means: IF fuzzified Capital data of the year (n-1) is Ai, THEN fuzzified capital for nth year will be A_j. According to our table the fuzzy logical relationship is:

Table 4. Fuzzy logical relationship

$A_1 \rightarrow A_1$	$A_4 \rightarrow A_6$	$A_{10} \rightarrow A_{11}$
$A_1 \rightarrow A_2$	$A_6 \rightarrow A_6$	$A_{11} \rightarrow A_{11}$
$A_2 \rightarrow A_2$	$A_6 \rightarrow A_7$	$A_{11} \rightarrow A_{12}$
$A_2 \rightarrow A_3$	$A_7 \rightarrow A_8$	$A_{12} \rightarrow A_{12}$
$A_3 \rightarrow A_3$	$A_8 \rightarrow A_9$	$A_{12} \rightarrow A_{13}$
$A_3 \rightarrow A_4$	$A_9 \rightarrow A_9$	$A_{13} \rightarrow A_{13}$
$A_4 \rightarrow A_4$	$A_9 \rightarrow A_{10}$	

Step 5: In order to find whether the trend is increasing or decreasing or is almost same, divide the fuzzified intervals obtained in step 2, into four subintervals of equal length and obtain first quartile point as decreasing trend value, and third quartile as increasing trend value. The following procedure is used to predict the value for the next year.

The procedure is based on second order difference. The difference of the differences dS is given by: $dS = (S_{n-1} - S_{n-2}) - (S_{n-2} - S_{n-3})$

Case i: If dS is positive the trend is upward
Case ii: If dS is negative the trend is downward
Case iii: If dS=0, the trend changes with a constant rate, than rule 2 or rule 3.

In order to predict the future value apply Rule 1, Rule 2 and Rule 3.

Rule 1: For forecasting data for the year 1982-1983 it is not possible to calculate dS since there is no data for the year 1979-1980.if the data for (n-3) is not available and if $|S_{n-1} - S_{n-2}|$ > half of length of the interval corresponding to fuzzified capital A_j with the membership value 1, then the trend of the forecasting will be upward and forecasted data falls at (3/4) point of this interval.

If $|S_{n-1} - S_{n-2}|$ < half of length of the interval corresponding to fuzzy length interval A_j with the membership value 1, then the trend of the forecasting will be downward and the forecasted capital will be ¼ point of this interval.

If $|S_{n-1} - S_{n-2}|$ = half of length of the interval corresponding to fuzzified data Aj with MF value 1, then the predicted value will be at the middle point of this interval.

Rule 2: If the $|(S_{n-1} - S_{n-2}) - (S_{n-2} - S_{n-3})| \times 2 + S_{n-1})$ Or $(S_{n-1} - (|(S_{n-1} - S_{n-2}) - (S_{n-2} - S_{n-3})| \times 2)$ that is if the difference of the differences between the years {((n-1) and (n-2) and (n-2) and (n-3)) x 2 + data of (n-1)}or {data of (n-1) – twice of difference of the differences of (n-1) and (n-2) and (n-2) and (n-3)} falls in the interval corresponding to the fuzzified data Aj with MF equal to 1, then the trend is upward and will take the value as (3/4) point of the interval.

Rule 3: If the $|(S_{n-1} - S_{n-2}) - (S_{n-2} - S_{n-3}) \div 2| + S_{n-1})$ Or $(S_{n-1} - (|(S_{n-1} - S_{n-2}) - (S_{n-2} - S_{n-3})| \div 2)$ that is if the {data of (n-1) + (difference of the difference (n-1) and (n-2) and data of (n-2) and (n-3)) /2) or {data of (n-1)-(difference of the difference between (n-1) and (n-2) data and (n-2) and (n-3) data)/2)} lies in the fuzzified capital data of Aj with membership function 1 then the trend will be downward and the predicted value will be (1/4) point of the length of the interval.

Rule 4: If the data does not follow rule 2 and rule 3 then the predicted capital will take the middle value of the fuzzified interval A_j with membership value 1.

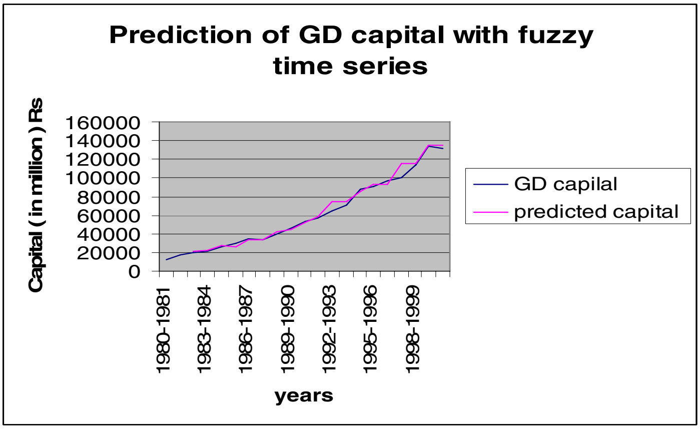
5 Results

By following the abovementioned methodology, the trend of the linguistic time series is predicted [Table 5]. The graph shows both historical and predicted data.

Table 5. Comparison of Historical data & predicted data

Year	Historical data	Trend	Predicted value	Forecasting error
1980-1981	12105			
1981-1982	16986			
1982-1983	20139	Downward	21250	5.516659
1983-1984	21265	Middle	22500	5.807665
1984-1985	25600	Middle	27500	7.421875
1985-1986	29990	Downward	26250	12.47082
1986-1987	34772	Upward	33750	2.939146
1987-1988	33757	Upward	33750	0.020736
1988-1989	40136	Downward	41666.7	3.813783
1989-1990	46405	upward	45000	3.027691
1990-1991	53099	Upward	51666.7	2.697414
1991-1992	57633	Upward	58333.3	1.215102
1992-1993	63997	Upward	75000	17.19299
1993-1994	70834	Upward	75000	5.881356
1994-1995	88206	Middle	85000	3.634673
1995-1996	90977	Downward	92500	1.674049
1996-1997	96187	Downward	92500	3.833158
1997-1998	100653	Upward	115000	14.25392
1998-1999	114545	Upward	115000	0.397224
1999-2000	134484	Upward	135000	0.383689
2000-2001	131505	upward	135000	2.657694

Forecasting error: Forecasting error is defined as ((predicted value ~ actual value)/actual value) *100. Average percentage forecasting error is 4.99 %.



6 Conclusions

Historically much data has been generated to facilitate human analysis for accurate prediction and so this has encouraged the use of fuzzy based analysis techniques over other possible methods of data processing. In addition, since humans are so adept at understanding gray logic i.e. fuzzy based techniques, fuzzy based analysis provides some aid in predicting time series and also helps informally validate results. From the above discussion it can be safely concluded that the present study is able to predict the capital for the coming years as well as the trend can be identified (as shown in the table). By classifying the historical data into fuzzy length interval makes it closer to the real world and narrows down the range of prediction. The present paper is a study of importance as it paves the way for laying the foundation to ensure increased robustness, versatility and reliability of fuzzy based systems.

Acknowledgement

The author acknowledges her deep sense of gratitude towards her Ph. D guide Dr Durgesh Pant.

References

1. Zadeh L A “Fuzzy sets and systems”, IN: Fox, journal ed. System Theory, Polytechnique press, Brooklyn, NY , 1965 –a, pp 29-37
2. Zadeh L A, “Fuzzy sets”, Inform. Control (8) 1965 –b, 338-353.
3. Song Q, A note on fuzzy time series model selection with sample autocorrelation functions. Cybernetics and Systems: An International Journal, 34: 2003 pp 93-107.
4. Song Q and Chissom BS, “Fuzzy Time Series and Its Models”, Fuzzy Sets and Systems, vol. 54, 1993 pp., 269-277.
5. Song Q and Chissom BS, Forecasting enrollments with fuzzy time series Part I, Fuzzy Sets and Systems, 54: 1993 pp1-9.
6. Song Q and Chissom BS, Forecasting enrollments with fuzzy time series - Part II. Fuzzy Sets and Systems, 62: 1994 pp 1-8.
7. Song Q and Leland RP, Adaptive learning defuzzification techniques and applications. Fuzzy Sets and Systems, 81: 1996 pp 321-329
8. Chen S. M, Forecasting enrollments based on fuzzy time series. Fuzzy Sets *and* Systems, 81: 1996 pp 311-319.
9. Chen SM and Hsu CC**,** A New Method to Forecast Enrollments Using Fuzzy Time Series**,** International Journal of Applied Science and Engineering 2004. 2, 3:pp 234-244
10. Huang JR, Chen SM and Lee CH, Handling forecasting problems using fuzzy time series. Fuzzy Sets and Systems, 100: 1998 pp 217-228.
11. Huang K, Heuristic models of fuzzy time series for forecasting. Fuzzy Sets and Systems, 123: 2001 pp 369-386.
12. Chen SM, Forecasting enrollments based on high-order fuzzy time series. Cybernetics and Systems: International Journal, 33: 2002 pp 1-16
13. Chen SM and Hwang JR, Temperature prediction using fuzzy time series. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 30: 2000. pp 263-275.

14. Li H and Kozma R, A dynamic neural network method for time series prediction using the KIII model. Proceedings of the 2003 International Joint Conference on Neural Networks, 1: 2003 pp 347-352.
15. Su SF and Li SH, Neural network based fusion of global and local information in predicting time series. Proceedings of the 2003 IEEE International Joint Conference on Systems, Man and Cybernetics, 5: 2003 pp 4445-4450.
16. Sullivan J and Woodall WH, A comparison of fuzzy forecasting and Markov modeling. Fuzzy Sets and Systems, 64: 1994. pp 279-293.
17. Lee CL, Liu A, Chen WS, "Pattern discovery of fuzzy time series for financial prediction", IEEE transactions on knowledge and data engineering, Vol 18, No 5, May 2006.
18. Tsai CC and Wu SJ, "A Study for Second-order Modeling of Fuzzy Time Series," Proc. IEEE Fuzzy System Conf., pp. 719-725, 1999.
19. Singh S and Stuart E, A Pattern Matching Tool for Forecasting, Proc. 14th International Conference on Pattern Recognition (ICPR'98), Brisbane, IEEE Press, vol. 1, (August 16-20, 1998) pp. 103-105

[T1]: Economic Survey, 2003-04, Gross Domestic Capital (public sector).

Design of Modular Neural Networks with Fuzzy Integration Applied to Time Series Prediction

Patricia Melin, Oscar Castillo, Salvador Gonzalez, Jose Cota, Wendy Lizeth Trujillo, and Paul Osuna

Division of Graduate Studies, Tijuana Institute of Technology, Mexico

Abstract. We describe in this paper the application of several neural network architectures to the problem of simulating and predicting the dynamic behavior of complex economic time series. We use several neural network models and training algorithms to compare the results and decide at the end, which one is best for this application. We also compare the simulation results with the traditional approach of using a statistical model. In this case, we use real time series of prices of consumer goods to test our models. Real prices of tomato and green onion in the U.S. show complex fluctuations in time and are very complicated to predict with traditional statistical approaches.

1 Introduction

Forecasting refers to a process by which the future behavior of a dynamical system is estimated based on our understanding and characterization of the system. If the dynamical system is not stable, the initial conditions become one of the most important parameters of the time series response, i.e. small differences in the start position can lead to a completely different time evolution. This is what is called sensitive dependence on initial conditions, and is associated with chaotic behavior [2, 16] for the dynamical system.

The financial markets are well known for wide variations in prices over short and long terms. These fluctuations are due to a large number of deals produced by agents that act independently from each other. However, even in the middle of the apparently chaotic world, there are opportunities for making good predictions [4,5]. Traditionally, brokers have relied on technical analysis, based mainly on looking at trends, moving averages, and certain graphical patterns, for performing predictions and subsequently making deals. Most of these linear approaches, such as the well-known Box-Jenkins method, have disadvantages [9].

More recently, soft computing [10] methodologies, such as neural networks, fuzzy logic, and genetic algorithms, have been applied to the problem of forecasting complex time series. These methods have shown clear advantages over the traditional statistical ones [12]. The main advantage of soft computing methodologies is that, we do not need to specify the structure of a model a-priori, which is clearly needed in the classical regression analysis [3]. Also, soft computing models are non-linear in nature and they can approximate more easily complex dynamical systems, than simple linear statistical models. Of course, there are also disadvantages in using soft computing

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 265–273, 2007.
springerlink.com

models instead of statistical ones. In classical regression models, we can use the information given by the parameters to understand the process, i.e. the coefficients of the model can represent the elasticity of price for a certain good in the market. However, if the main objective if to forecast as closely as possible the time series, then the use of soft computing methodologies for prediction is clearly justified.

2 Monolithic Neural Network Models

A neural network model takes an input vector X and produces and output vector Y. The relationship between X and Y is determined by the network architecture. There are many forms of network architecture (inspired by the neural architecture of the brain). The neural network generally consists of at least three layers: one input layer, one output layer, and one or more hidden layers. Figure 1 illustrates a neural network with p neurons in the input layer, one hidden layer with q neurons, and one output layer with one neuron.

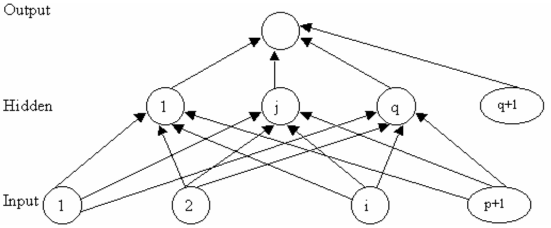


Fig. 1. Single hidden layer feedforward network

In the neural network we will be using, the input layer with p+1 processing elements, i.e., one for each predictor variable plus a processing element for the bias. The bias element always has an input of one, $X_{p+1}=1$. Each processing element in the input layer sends signals Xi (i=1,…,p+1) to each of the q processing elements in the hidden layer. The q processing elements in the hidden layer (indexed by j=1,…,q) produce an "activation" $a_j=F(\Sigma w_{ij}X_i)$ where w_{ij} are the weights associated with the connections between the p+1 processing elements of the input layer and the jth processing element of the hidden layer. Once again, processing element q+1 of the hidden layer is a bias element and always has an activation of one, i.e. $a_{q+1}=1$. Assuming that the processing element in the output layer is linear, the network model will be

$$Y_t = \sum_{j=1}^{p+1} \pi_\iota x_{it} + \sum_{j=1}^{p+1} \theta_j F\left(\sum_{i=1}^{p+1} w_{ij} x_{it} \right) \qquad (1)$$

Here π_ι are the weights for the connections between the input layer and the output layer, and θ_j are the weights for the connections between the hidden layer and the output layer. The main requirement to be satisfied by the activation function F(˙) is that it be nonlinear and differentiable. Typical functions used are the sigmoid, hyperbolic tangent, and the sine functions.

The weights in the neural network can be adjusted to minimize some criterion such as the sum of squared error (SSE) function:

$$E_l = \frac{1}{2} \sum_{l=1}^{n} (d_l - y_l)^2 \tag{2}$$

Thus, the weights in the neural network are similar to the regression coefficients in a linear regression model. In fact, if the hidden layer is eliminated, (1) reduces to the well-known linear regression function. It has been shown [22] that, given sufficiently many hidden units, (1) is capable of approximating any measurable function to any accuracy. In fact F(.) can be an arbitrary sigmoid function without any loss of flexibility.

The most popular algorithm for training feedforward neural networks is the backpropagation algorithm [14,18]. As the name suggests, the error computed from the output layer is backpropagated through the network, and the weights are modified according to their contribution to the error function. Essentially, backpropagation performs a local gradient search, and hence its implementation does not guarantee reaching a global minimum. A number of heuristics are available to partly address this problem, some of which are presented below. Instead of distinguishing between the weights of the different layers as in Equation (1), we refer to them generically as w_{ij} in the following.

After some mathematical simplification the weight change equation suggested by backpropagation can be expressed as follows:

$$w_{ij} = -\eta \frac{\partial E_l}{\partial w_{ij}} + \theta \Delta wij \tag{3}$$

Here, η is the learning coefficient and θ is the momentum term. One heuristic that is used to prevent the neural network from getting stuck at a local minimum is the random presentation of the training data.

3 Modular Neural Networks

There exists a lot of neural network architectures in the literature that work well when the number of inputs is relatively small, but when the complexity of the problem grows or the number of inputs increases, their performance decreases very quickly. For this reason, there has also been research work in compensating in some way the problems in learning of a single neural network over high dimensional spaces.

In the work of Sharkey [20], the use of multiple neural systems (Multi-Nets) is described. It is claimed that multi-nets have better performance or even solve problems that monolithic neural networks are not able to solve. It is also claimed that multi-nets or modular systems have also the advantage of being easier to understand or modify, if necessary.

In the literature there is also mention of the terms “ensemble” and “modular” for this type of neural network. The term “ensemble” is used when a redundant set of neural networks is utilized, as described in Hansen and Salamon [8]. In this case, each of the neural networks is redundant because it is providing a solution for the same task, as it is shown in Figure 2.

On the other hand, in the modular approach, one task or problem is decompose in subtasks, and the complete solution requires the contribution of all the modules, as it is shown in Figure 3.

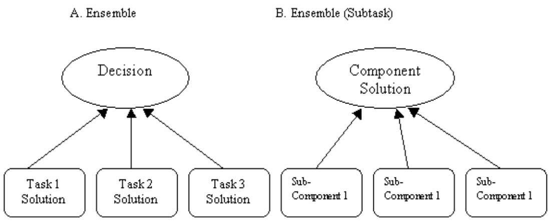


Fig. 2. Ensembles for one task and subtask

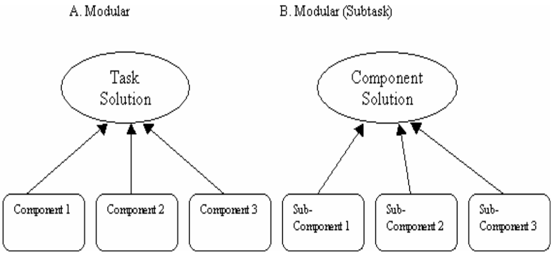


Fig. 3. Modular approach for task and subtask

4 Methods for Response Integration

In the literature we can find several methods for response integration, that have been researched extensively, which in many cases are based on statistical decision methods. We will mention briefly some of these methods of response integration, in particular the ones based on fuzzy logic. The idea of using these types of methods, is that the final decision takes into account all of the different kinds of information available about the time series. In particular, we consider aggregation operators, and the fuzzy Sugeno integral [21].

Yager [23] mentions in his work, that fuzzy measures for the aggregation criteria of two important classes of problems. In the first type of problems, we have a set $Z=\{z_1,z_2,\ldots,z_n\}$ of objects, and it is desired to select one or more of these objects based on the satisfaction of certain criteria. In this case, for each $z_i \in Z$, it is evaluated $D(z_i)=G(A_i(z_i),\ldots,A_j(z_i))$, and then an object or objects are selected based on the value of G. The problems that fall within this structure are the multi-criteria decision problems, search in databases and retrieving of documents.

In the second type of problems, we have a set $G=\{G_1,G_2,\ldots,G_q\}$ of aggregation functions and object z. Here, each G_k corresponds to different possible identifications of object z, and our goal is to find out the correct identification of z. For achieving this, for each aggregation function G, we obtain a result for each z, $D_k(z)=G_k(A_1(z),$

A_2(z), … ,A_n(z)). Then we associate to z the identification corresponding to the larger value of the aggregation function.

A typical example of this type of problems is pattern recognition. Where A_j corresponds to the attributes and A_j(z) measures the compatibility of z with the attribute. Medical applications and fault diagnosis fall into this type of problems. In diagnostic problems, the A_j corresponds to symptoms associated with a particular fault, and G_k captures the relations between these faults.

Fuzzy integrals can be viewed as non-linear functions defined with respect to fuzzy measures. In particular, the "gλ-fuzzy measure" introduced by Sugeno [21] can be used to define fuzzy integrals. The ability of fuzzy integrals to combine the results of multiple information sources has been mentioned in previous works.

Definition 1. A function of sets $g{:}2^x$-(0.1) is called a fuzzy measure if:

1. g(0)=0 g(x)=1
2. $g(A) \leq g(B)$ if $A \subset B$
3. if $\{A_i\}i^{\alpha} = 1$ is a sequence of increments of the measurable set then

$$\lim_{i \to \infty} g(A_i) = g\,(\lim_{i \to \infty} A_i) \tag{4}$$

From the above it can be deduced that g is not necessarily additive, this property is replaced by the additive property of the conventional measure.

From the general definition of the fuzzy measure, Sugeno introduced what is called "gλ-fuzzy measure", which satisfies the following additive property: For every A, B $\subset$ X and $A \cap B = \theta$,

$$g(A \cup B) = g(A) + g(B) + \lambda\, g(A)g(B), \tag{5}$$

for some value of $\lambda > -1$.

This property says that the measure of the union of two disjunct sets can be obtained directly from the individual measures. Using the concept of fuzzy measures, Sugeno [21] developed the concept of fuzzy integrals, which are non-linear functions defined with respect to fuzzy measures like the gλ-fuzzy measure.

Definition 2. Let X be a finite set and $h{:}X \rightarrow [0,1]$ be a fuzzy subset of X, the fuzzy integral over X of function h with respect to the fuzzy measure g is defined in the following way,

$$\begin{aligned} h(x) \circ g\,(x) &= \max_{E \subseteq X} \left[\min \left(\min_{x \in E} h(x), g(E) \right) \right] \\ &= \sup_{\alpha \in [0,1]} \left[\min(\alpha , g(h_\alpha)) \right] \end{aligned} \tag{6}$$

where h_α is the level set α of h,

$$h_\alpha = \{ x \mid h(x) \geq \alpha \}. \tag{7}$$

We will explain in more detail the above definition: h(x) measures the degree to which concept h is satisfied by x. The term min(h_x) measures the degree to which concept h is satisfied by all the elements in E. The value g(E) is the degree to which the subset of objects E satifies the concept measure by g. As a consequence, the obtained value of comparing these two quantities in terms of operator min indicates the degree to which E satifies both criteria g and min(h_x). Finally, operator max takes the greatest of these terms.

5 Simulation and Forecasting Prices in the U.S. Market

We will consider the problem forecasting the prices of tomato in the U.S. market. The time series for the prices of this consumer good shows very complicated dynamic behavior, and for this reason it is interesting to analyze and predict the future prices for this good. We show in Figure 4 the time series of monthly tomato prices in the period of 1960 to 1999, to give an idea of the complex dynamic behavior of this time series.

We will apply both the modular and monolithic neural network approach and also the linear regression method to the problem of forecasting the time series of tomato prices. Then, we will compare the results of these approaches to select the best one for forecasting.

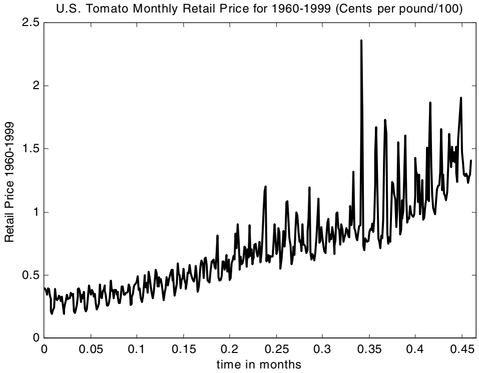


Fig. 4. Prices in US Dollars of tomato from January 1960 to December 1999

6 Experimental Results

We describe, in this section, the experimental results obtained by using neural networks to the problem of forecasting tomato prices in the U.S. Market. We show results of the application of several architectures and different learning algorithms to decide on the best one for this problem. We also compare at the end the results of the neural network approach with the results of linear regression models, to measure the difference in forecasting power of both methodologies.

First, we will describe the results of applying modular neural networks to the time series of tomato prices. We used the monthly data from 1960 to 1999 for training a Modular Neural Network with four Modules, each of the modules with 80 neurons and one hidden layer. We show in Figure 5 the result of training the modular neural network with this data. In Figure 5, we can appreciate how the modular neural network approximates very well the real time series of tomato prices over the relevant period of time.

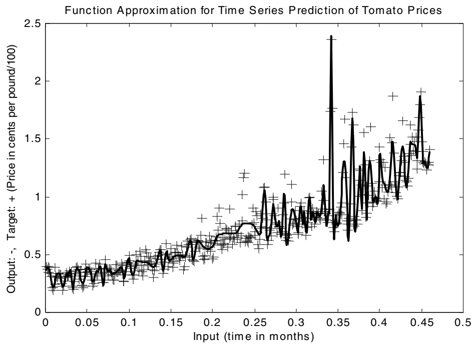


Fig. 5. Modular network for tomato prices with Levenberg-Marquardt algorithm

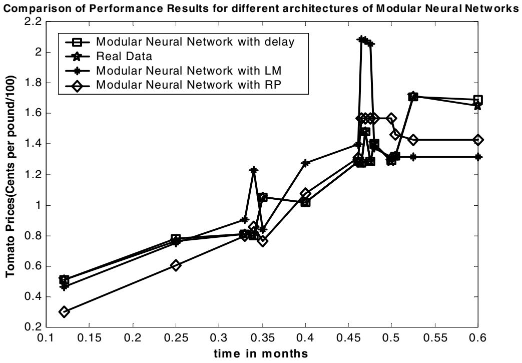


Fig. 6. Comparison of performance results for several modular neural networks

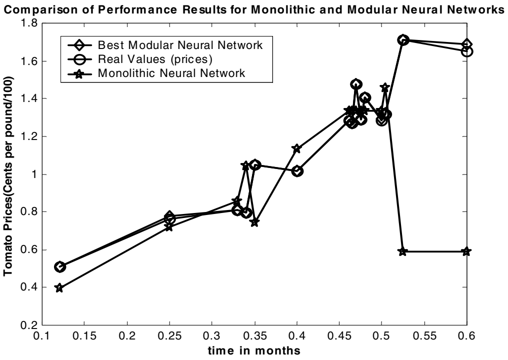


Fig. 7. Comparison of monolithic and modular neural networks

We have to mention that the results shown in Figure 5 are for the best modular neural network that we were able to find for this problem. We show in Figure 6 the comparison between several of the modular neural networks that we tried in our experiments. From Figure 6 we can appreciate that the modular neural network with one time delay and Leverberg-Marquardt (LM) training algorithm is the one that fits best the data and for this reason is the one selected.

We show in Figure 7 the comparison of the best monolithic network against the best modular neural network. The modular network clearly fits better the real data of the problem.

7 Conclusions

We described in this paper the use of modular neural networks for simulation and forecasting time series of consumer goods in the U.S. Market. We have considered a real case to test our approach, which is the problem of time series prediction of tomato prices in the U.S. market. We have applied monolithic and modular neural networks with different training algorithms to compare the results and decide which is the best option. The Levenberg-Marquardt learning algorithm gave the best results. The performance of the modular neural networks was also compared with monolithic neural networks. The forecasting ability of modular networks was clearly superior.

References

[1] Boers, E. and Kuiper, H. (1992) Biological Metaphors and the Design of Modular Artificial Neural Networks. Departments of Computer Science and Experimental and Theoretical Psychology at Leiden University, the Netherlands.

[2] Brock, W.A., Hsieh, D.A., and LeBaron, B. (1991). “Nonlinear Dynamics, Chaos and Instability”, MIT Press, Cambridge, MA, USA.

[3] Castillo, O. and Melin, P. (1996). “Automated Mathematical Modelling for Financial Time Series Prediction using Fuzzy Logic, Dynamical System Theory and Fractal Theory”, Proceedings of CIFEr’96, IEEE Press, New York, NY, USA, pp. 120-126.

[4] Castillo, O. and Melin P. (1998). “A New Fuzzy-Genetic Approach for the Simulation and Forecasting of International Trade Non-Linear Dynamics”, Proceedings of CIFEr’98, IEEE Press, New York, USA, pp. 189-196.

[5] Castillo, O. and Melin, P. (1999). “Automated Mathematical Modelling for Financial Time Series Prediction Combining Fuzzy Logic and Fractal Theory”, Edited Book “Soft Computing for Financial Engineering”, Springer-Verlag, Germany, pp. 93-106.

[6] O. Castillo and P. Melin, Soft Computing and Fractal Theory for Intelligent Manufacturing. Springer-Verlag, Heidelberg, Germany, 2003.

[7] Fu, H.-C., Lee, Y.-P., Chiang, C.-C., and Pao, H.-T. (2001). Divide-and-Conquer Learning and Modular Perceptron Networks in IEEE Transaction on Neural Networks, vol. 12, No. 2, pp. 250-263.

[8] Hansen, L. K. and Salamon P. (1990). Neural Network Ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, pp. 993-1001.

[9] Haykin, S. (1996). “Adaptive Filter Theory”, Third Edition, Prentice Hall.

[10] Jang, J.-S. R., Sun, C.-T., and Mizutani, E. (1997). “Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, Prentice Hall.

[11] Lu, B. and Ito, M. (1998). Task Decomposition and module combination based on class relations: modular neural network for pattern classification. Technical Report, Nagoya Japan, 1998.

[12] Maddala, G.S. (1996). “Introduction to Econometrics”, Prentice Hall.

[13] Murray-Smith, R. and Johansen, T. A. (1997). Multiple Model Approaches to Modeling and Control. Taylor and Francis, UK.

[14] Parker, D.B. (1982). “ Learning Logic”, Invention Report 581-64, Stanford University.

[15] Quezada, A. (2004). Reconocimiento de Huellas Digitales Utilizando Redes Neuronales Modulares y Algoritmos Geneticos. Thesis of Computer Science, Tijuana Institute of Technology, Mexico.

[16] Rasband, S.N. (1990). “Chaotic Dynamics of Non-Linear Systems”, Wiley.

[17] Ronco, E. and Gawthrop, P. J. (1995). Modular neural networks: A State of the Art. Technical Report, Center for System and Control. University of Glasgow, Glasgow, UK, 1995.

[18] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). “Learning Internal Representations by Error Propagation”, in “Parallel Distributed Processing: Explorations in the Microstructures of Cognition”, MIT Press, Cambridge, MA, USA, Vol. 1, pp. 318-362.

[19] Schdmit, A. and Bandar, Z. (1997). A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors, Proceedings of ICANNGA'97, Norwich, England.

[20] Sharkey, A. (1999). Combining Artificial Neural Nets: Ensemble and Modular Multi-Nets Systems, Ed. Springer-Verlag, London, England.

[21] Sugeno, M. (1974). Theory of fuzzy integrals and its application, Doctoral Thesis, Tokyo Institute of Technology, Japan.

[22] White, H. (1989). “An Additional Hidden Unit Test for Neglected Non-linearity in Multilayer Feedforward Networks”, Proceedings of IJCNN’89, Washington, D.C., IEEE Press, pp. 451-455.

[23] Yager, R. R. (1999). Criteria Aggregations Functions Using Fuzzy Measures and the Choquet Integral, International Journal of Fuzzy Systems, Vol. 1, No. 2.

Part V

Pattern Recognition

Characterize the Parameters of Genetic Algorithms Based on Zernike Polynomials for Recovery of the Phase of Interferograms of Closed Fringes Using Hybrid Technique

Luis Ernesto Mancilla Espinosa[1], Juan Martin Carpio Valadez[2], and F.J. Cuevas[3]

[1] Technology Institute of Leon, Systems and Computation Department and CIDESI, Qro, Mexico
lmancilla01@hotmail.com
[2] Technology Institute of Leon, Systems and Computation Department, Mexico
jmcarpio@itleon.edu.mx
[3] Optics Reasearch Center, C.A., P.O. Box 1-948, Leon, Gto., Mexico
fjcuevas@cio.mx

Abstract. In this work a hybrid technique based on heuristics was used to characterize the interval of decoding of the parameters of genetic algorithms (AG) to the recovery of the phase of interferograms of closed fringes.

Keywords: Genetic Algorithms, Heuristics, Interferometry, Phase recover, Zernike polynomials.

1 Introduction

The interferometry is an optical technique does not destructive; it used to measure physics variables (stress, temperature, acceleration, curvature, and so on) with high degree of resolution, because it follows from the magnitude of wavelength used of the light [1].

In optical metrology, the mathematical model to state the phenomena of interference is modeling across a fringes pattern that modulates a signal with cosenoidal profile as [2]:

$$I(x,y)=a(x,y)+b(x,y)\cos(w_x x+w_y y+\Theta(x,y)+\eta(x,y))$$

Where $a(x,y)$ is the background lighting, $b(x,y)$ is the amplitude modulation, w_x and w_y are the carrier frequencies on the axis x and y, $\Theta(x,y)$ is the phase term related to the physical quantity being measured, $\eta(x,y)$ is an additive uniform noise.

The goal of genetics algorithms is to recover $\Theta(x,y)$ from the fringe pattern, which it is directly relationship with the physics quantity that we want to measure. By other part, the experimental array to provoke the interference phenomenon requires recording the interference image that to generate, and it processed digitally in the computer to obtain the information of the phase.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 277–289, 2007.
springerlink.com

One way to calculate the phase term $\Theta(x, y)$ is the phase shift technical (PST) [10-14], which needs unless of three phase interferograms. The shift of phase between the interferograms must to be known and controlled experimentally. This technical can be used when the mechanical conditions are satisfied in the interferometry experiment.

In other conditions, when the stability condition is not cover, there are a lot of techniques to obtain the term of phase from the fringes pattern, as: the Fourier Method [15, 16], Synchronous Method [17] and the Phase Lock Loop (PLL) [18], between others. However, these techniques work well only if the interferogram analyzed has a carrier frequency, a narrow bandwidth and the signal has low noise. Even more, these methods fail when we calculate the phase of an interferogram with closed fringes, as the figure 1. Besides, the Fourier and Synchronous methods estimate the phase wrapped because of the arctangent function used in the phase calculation, so an additional unwrapping process is required. The unwrapping process is difficult when the fringes pattern includes high amplitude, which causes differences greater than 2π radians between adjacent pixels [19-20]. In PLL technique, the phase is estimated by following the phase changes of the input signal by varying the phase of a computer simulated oscillator (VCO) such that the phase error between the fringe pattern and VCO's signal vanishes.

Recently, the regularization [21-24] and neural network techniques [25,26] have been used to work with fringes pattern, which contain a narrow bandwidth and noise. The regularization technique established a cost function that constrains the estimated phase using two considerations: (a) the fidelity between the estimated function and the observed fringes pattern and (b) smoothness of the modulated phase field. In the neural network technique, a multilayer neural network is trained with a set of fringes pattern and a set of phase gradients provided from calibrated objects. After that the network has been trained, the phase gradients is estimated in the network output when the fringes pattern (interferograms) are presented in the network input. The defect of this technique is the requirement of a set of training fringe images and theirs related phase measurements.

Therefore to the case of interferograms as the figure 1, we require different techniques, as the proposals in the articles [2], [3], [4], [8], [9].

Our proposal consists to use Zernike polynomials to optimistic a quadratic criterion. The Zernike polynomials are used to obtain the interferogram phase that phase is showed as a surface and due to the orthogonal characteristics of the Zernike polynomials, these are very suitable to carry out the fitting of that surface. By other way, the parameters of the Zernike polynomials have direct relation with the physics properties as: aberration spherical coefficient, come coefficient, astigmatism coefficient, focus shift coefficient, tilt in y, and tilt in x, so on. [1].

Therefore, we use the Zernike polynomials to recover the closed interferograms phase using AG.

The paper is organized as follows, in the section 2 we establish the mathematics models that used to carry out the optimization by AG. In the section 3 the heuristics model, in the section 4 we show the experimental results obtained. And finally in the section 5, conclusions and comments.

2 Mathematics Models

The model used to represent the interference phenomena is:

$$I(x,y)=a(x,y)+b(x,y)\cos(w_x x+w_y y+\Theta(x,y))$$

The optimization criterion that we use, it was:

$$\begin{gathered}f_{opt}(z^k)=\alpha-\sum_{y=1}\sum_{x=1}\{(I(x,y)-\cos(w_x x+w_y y+\\ f_{ajuste}(z^k,x,y))^2+\lambda\{(f_{ajuste}(z^k,x,y)-\\ f_{ajuste}(z^k,x-1,y))^2+(f_{ajuste}(z^k,x,y)-f_{ajuste}(z^k,x,y-1))^2\}\}\\ +\beta\{(f_{ajuste}(z^k,x,y)-2f_{ajuste}(z^k,x-1,y)+f_{ajuste}(z^k,x-2,y))^2+\\ (f_{ajuste}(z^k,x,y)-2f_{ajuste}(z^k,x,y-1)+f_{ajuste}(z^k,x,y-2))^2\}\end{gathered}$$

Where:

x, y are integer values representing indexes of the pixel location in the fringes image, α is used to convert the proposal from minimal to maximal optimization, w_x is the frequency in the axe x, w_y is frequency in the axe y, λ parameter which control the smoothness level of the image to obtain, β parameter which control the impact level of second derivative term and discriminate the concavity or convexity of the image, z^k is the k eth chromosome of the population, k is the index into the whole population of chromosomes.

The fitness function that we use is the Zernike polynomial which cartesian model is [2, pag.465], [7]:

$$f_{ajuste}(z^k,x,y)=\sum_{j=1}^{j=L} z^k{}_j U_j(x,y)$$

Where:

j is the order of Zernike polynomial, U is the result polynomial of mapping the plane (ρ,θ) to plane (x,y).

The AG that we use to have the following characteristic:

Select Operator, Boltzman type.

Cross Operator with two points of cross.

Mutation Operator with stepped degraded.

The Zernike polynomial that we use is the four order.

The similarity criterion between the inteferograms that we used was the calculus of the Euclidian distance between the images [8].

3 Heuristics Model

The search intervals to the Zernike variables polynomial were obtained by heuristics model follows.
It shows as algorithms in pseudo-code format.

```
Read the N genetics parameters with the values of theirs minimal interval and maximal interval.
Run the Genetics Algorithms and find the best value of Fitness, and store into FitnessMax_old variable.
If the parameter value( i ) >0 then his value of minimal interval=0,
else his value of maximal interval=0.
Run the Genetics Algorithms and find the best value of Fitness, and store into FitnessMax_new variable.
If FitnessMax_new>FitnessMax_old then FitnessMax_old=FitnessMax_new and go to subrutine of Parameters Change, else send a warning and End.

Subrutine of Parameters Change.
Do
{Label A
If the parameter(i) >0 then his value of maximal interval=maximal interval/10 and run the Genetics Algorithms, Else his value of minimal interval=value of minimal interval/10 and run the Genetics Algorithms.
{If FitnessMax_new>FitnessMax_old then FitnessMax_old=FitnessMax_new and go to Label A,
Else
If the parameter(i) >0 his value of maximal interval=value of maximal interval*10,
Else his value of minimal interval=value of minimal interval*10.
}
Add one to i parameter.
If i > N then End,
Else go to label A.
}
```

4 Experimental Results

The computer equipment that we use to do the tests is a PC-Pentium Centrino 1.6GHz, 256 MB of RAM.

Case 1. Image of 40x40 pixels shows figure 1, resolved with: population=1000, cross probability=90%, mutation probability=10%, number of generations=50, time used to obtain the figure 2 is of: 540 seconds.

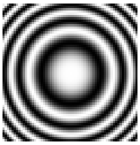


Fig. 1.

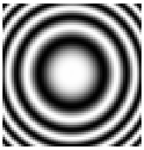


Fig. 2.

The figures 3,5,7,9 y 11; were obtained [5] with a fitness function:

$$f_{ajuste}(z,x,y) = z_0 + z_1 x + z_2 y + z_3 x^2 + z_4 xy + \\ z_5 y^2 + z_6 x^3 + z_7 x^2 y + z_8 xy^2 + z_9 y^3 + z_{10} x^3 y + \\ z_{11} x^2 y^2 + z_{12} xy^3 + z_{13} x^4 + z_{14} y^4$$

And with search interval found by heuristics model:

$$z_1, z_2 \in [-2,2]; z_3, z_4, z_5 \in [-0.05,0.05]; \\ z_6, z_7, z_8, z_9 \in [-0.003,0.003]; \\ z_{10}, z_{11}, z_{12}, z_{13}, z_{14} \in [-0.0001,0.0001]$$

The figures 4,6,8,10 and 12; were obtained with a fitness function:

$$f_{ajuste}(z,x,y) = z_1 U_1 + z_2 U_2 + z_3 U_3 + z_4 U_4 + \\ z_5 U_5 + z_6 U_6 + z_7 U_7 + z_8 U_8 + z_9 U_9 + z_{10} U_{10} + \\ z_{11} U_{11} + z_{12} U_{12} + z_{13} U_{13} + z_{14} U_{14} + z_{15} U_{15}$$

Case 2. Image of 40x40 pixels show figure 3, resolved with: population=1500, cross probability=90%, mutation probability=1%, number of generations=40, time used to obtain the figure 4 is of: 360 seconds.

With a search interval found by heuristics model:

$$\begin{gathered} z_1 \in [0.99,1.01], z_2 \in [-1,0], z_3 \in [0,0.5], \\ z_4, z_8 \in [0,0.001], \\ z_5 \in [-0.01,0], z_6 \in [-0.1,0], z_7 \in [-0.001,0], \\ z_9, z_{12} \in [-0.000001,0], z_{10}, z_{11} \in [-0.00001,0], \\ z_{13}, z_{15} \in [0,0.00001], z_{14} \in [0,0.0001] \end{gathered}$$

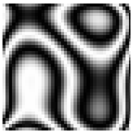


Fig. 3.

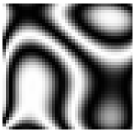


Fig. 4.

Case 3. Image of 40x40 pixels show figure 5, resolved with: population=500, cross probability=100%, mutation probability=4%, number of generations=170, time used to obtain the figure 6 is of: 560 seconds.

With a search interval found by heuristics model:

$$\begin{gathered} z_1 \in [0.99,1.01], z_2 \in [-1,0], \\ z_3, z_9, z_{12} \in [-0.000001,0], z_4, z_8 \in [0,0.001], \\ z_5, z_7 \in [-0.01,0], z_6 \in [-0.1,0], \\ z_{10}, z_{11} \in [-0.00001,0], z_{13} \in [0,0.00001], \\ z_{14} \in [0,0.0001], z_{15} \in [0,0.000001] \end{gathered}$$

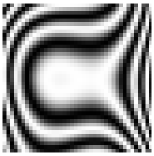


Fig. 5.

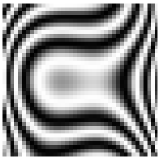


Fig. 6.

Case 4. Image of 40x40 pixels show figure 7, resolved with: population=500, cross probability=100%, mutation probability=4%, number of generations=170, time used to obtain the figure 8 is of: 560 seconds.

With a search interval found by heuristics model:

$$z_1 \in [0.99,1.01], z_2, z_3 \in [-1,0], z_4, z_5, z_8 \in [0,0.001],$$
$$z_9, z_{12} \in [-0.000001,0],$$
$$z_{10}, z_{11} \in [-0.00001,0], z_{13}, z_{14} \in [0,0.00001],$$
$$z_6 \in [-0.01,0], z_7 \in [-0.001,0],$$
$$z_{15} \in [0,0.0001]$$

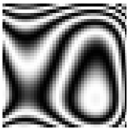


Fig. 7.

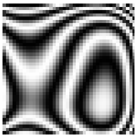


Fig. 8.

Case 5. Image of 40x40 pixels show figure 9, resolved with: population=500, cross probability=100%, mutation probability=4%, number of generations=170, time used to obtain the figure 10 is of: 360 seconds.

With a search interval found by heuristics model:

$$z_1 \in [0.99,1.01], z_2, z_3 \in [0,1], z_4 z_5 \in [-0.01,0],$$
$$z_8, z_{11} \in [-0.0001,0],$$
$$z_6 \in [0,0.01], z_7 \in [-0.001,0], z_9 \in [0,0.0001],$$

$$z_{12}, z_{14} \in [-0.00001,0], z_{13}, z_{15} \in [0,0.00001],$$
$$z_{10} \in [0,0.001]$$

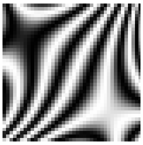


Fig. 9.

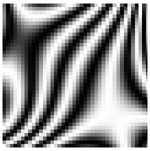


Fig. 10.

Case 6. Image of 40x40 pixels show figure 11, resolved with: population=500, cross probability=100%, mutation probability=4%, number of generations=170, time used to obtain the figure 12 is of: 360 seconds.

With a search interval found by heuristics model:

$$z_1 \in [0.99,1.01], z_2, z_6 \in [-0.1,0], z_3 \in [-1,0],$$
$$z_9, z_{12} \in [-0.000001,0], z_8 \in [0,0.001]$$
$$z_5 \in [-0.01,0], z_6 \in [-0.1,0], z_7 \in [-0.001,0],$$
$$z_{10}, z_{11} \in [-0.00001,0], z_{13}, z_{14} \in [0,0.00001],$$
$$z_{15} \in [0,0.00001], z_4 \in [0,0.01]$$

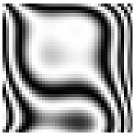


Fig. 11.

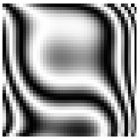


Fig. 12.

The case 1 is selected from reference [2] and the cases 2 to 6 is selected from reference [5].

Case 7. Image of 40x40 pixels shows figure 13, resolved with: population=500, cross probability=100%, mutation probability=1%, number of generations=30, time used to obtain the figure 14 is of: 60 seconds.

With a search interval found by heuristics model:

$$z_1 \in [5,7], z_2 \in [6,8], z_3 \in [4,6], z_4 \in [3,5],$$
$$z_5 \in [2,4], z_6 \in [1,3], z_7 \in [0,-2], z_8 \in [-1,-3],$$
$$z_9 \in [-2,-4], z_{10} \in [3,5], z_{11} \in [5,7], z_{12} \in [7,9],$$
$$z_{13} \in [4,6], z_{14} \in [6,8], z_{15} \in [0,2]$$

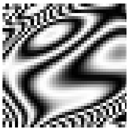


Fig. 13.

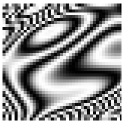


Fig. 14.

Case 8. Image of 40x40 pixels shows figure 15, resolved with: population=500, cross probability=100%, mutation probability=1%, number of generations=20, time used to obtain the figure 16 is of: 60 seconds.

With search interval found by heuristics model:

$$z_1 \in [0.05,0.15], z_2 \in [0.15,0.25], z_3 \in [0.25,0.35],$$
$$z_4 \in [0.35,0.45], z_5 \in [0.45,0.55], z_6 \in [0.55,0.65],$$
$$z_7 \in [0.65,0.75], z_8 \in [0.75,0.85], z_9 \in [0.15,0.05],$$
$$z_{10} \in [-0.25,-0.15], z_{11} \in [-0.35,-0.25],$$
$$z_{12} \in [-0.45,-0.35], z_{13} \in [-0.55,-0.45],$$
$$z_{14} \in [-0.65,-0.55], z_{15} \in [-0.75,-0.65]$$

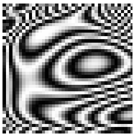


Fig. 15.

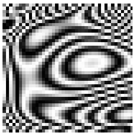


Fig. 16.

The cases 7 and 8 is selected from the reference [6]

Case 9. Image of 64x64 pixels shows figure 17, resolved with: population=500, cross probability=90%, mutation probability=1%, number of generations=30, time used to obtain the figure 18 is of: 180 seconds.

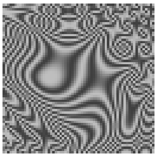


Fig. 17.

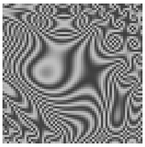


Fig. 18.

The case 9, we were selected from the reference [2].

5 Conclusions

We have presented results selected the phase recover, in basis to its recover time, for several closed interferograms applying a fitness function by Zernike polynomial and using AG whose parameters interval was found by heuristics model.

We searched to prove the advantages of our technique with respect to others and under criterions as:

a).-Recover the interferogram and consequently its phase.
b).-Direct relationship of physics parameters.
c).-Recover Time.

We can state that:

1.-In the figures 3,5,7,9,11 that were generated with not Zernike polynomials and using AG, in accordance with reference [5], and similar conditions of search range, we get similar results, with the difference that we can correlation the parameters of Zernike polynomials with optics values and the reference [5] did not. Moreover, we were reported the recover time and the reference [5] did not, in that point we could not contrast the advantages.

2.-For the figures 13, 15 of the reference [6] that were generated with Zernike polynomial and recovered the interferogram parameters with technique of quadratic minimal using Zernike polynomial, also were recovered satisfactorily and acceptable time. Newly, we can not to compare recover times because the reference [6] did not report.

3.-For the figure 17 of the reference [2] that use AG but not Zernike polynomial, it report a recover time 330 sec., we recover it in 180 sec., under similar conditions of computer equipment.

4.-About to reference [8],

a).-They used real interferograms, we did not by the moment, we are working in this part.

b).-Use an evolutionary algorithms and not a genetics algorithms.

c).-Use a Seidel polynomial and not a Zernike polynomial.

d).-Establish the problem as an optimization problem in the real space, similar to ours.

e).-Use a Pentium IV, but did not mention the speed computer.

f).-The experimentals (2) take between 3 to 6 minutes, we carry out several experimentals with results of better time.

g).-Use MatLab, we use “C”.

h).-Have serious problems for Seidel Polynomial of order bigger than 3, we have not problem with order Zernike polynomial.

i).-Assert that the evolutionary strategy is generally recommended to the genetics algorithms to resolve problems that deal with the optimization function of real numbers, but has not reference to this assert.

j).-Use a factor c=0.01 for the experimentals but does not say how to obtain it.

k).-Assert that if the success of mutation happen rarely, the search say us that we are near to a minimal, by recommend to decrease the size of neighborhood when we search. If the mutations are successful and they happen very frequently, it means that the converse can acceleration across the increase of the step, however did not show any references about theses assertions.

l).-They mention that after a lot of attempt found the cross and mutation configuration in terms following: a 10% for the equation (12), a 30% for the equation (13) and a 70% for the equation (14). Similar to ours, they carry out several attempts for fit the ranks.

5.-Respect to the reference [9],

a).-Use a polynomial equation not Zernike to obtain the phase.

b).-Use simulate interferograms with noise.

c).-Use an evolution algorithms and not a genetics algorithms.
d).-Drive one experimental, we report several.
e).-Do not report recover time, we do.
f).-Do not report the equipment type that they were used, we do.
g).-Use Matlab, we use “C”.

We conclude that our proposal shows better results respect to other techniques.

As future work we seek to carry off AG to embedded software using FPGA technology not parallel way, after in parallel way to recover the phase in real time and reduce the recover time.

References

[1] D. Malacara, *Optical Shop Testing*, (Wiley, 1992).

[2] F.J. Cuevas, J.H. Sossa-Azuela, M. Servin, “A parametric method applied to phase recovery from a fringe pattern based on a genetic algorithms”, *Optics Comunications* (February 2002).

[3] F.J.Cuevas, O.Gonzalez, Y. Susuki, D. Hernandez, M. Rocha, “Genetic Algorithms applied to optics and engineering”, *Research Congress and Technology Development, Leon, Gto., Mexico* (2005).

[4] F.J. Cuevas, “Soft computing algorithms for phase detection from fringe pattern”, *Research Signpost* 21,39, ISBN: 82-271-0028-5, Kerala India, 2003.

[5] Otoniel González Angeles, master degree thesis *Demodulation of Closed Fringe Patterns across a Genetics Algorithms* (Technology Institute of Leon, León, Gto., December 2005).

[6] Gerardo David Aguayo Ríos, degree thesis *Development of software in C++ to evaluate aspherics optics superface by minimal squares using zernike polynomials* (Technology Institute of Leon, León, Gto., January 1998).

[7] Martin Carpio, Daniel Malacara “Closed cartesian representation of the Zernike polynomials”, *Optics Commnications* (October 1993).

[8] Juan Jaime Sanchez Escobar, Sergio Vazquez Montiel, “Experimental Interferogram Analysis Using An Automatic Polynomial Fitting Method Based On Evolutionary Computation”, *Optical Engineering*, VOL. 44, No.4, (2005).

[9] Sergio Vazquez Montiel, Juan Jaime Sanchez Escobar, Olac Fuentes, “Obtaining The Phase of Interferograms Employing An Evolution Strategy, Part 1”, *Applied Optics*, VOL. 41, pag. 3448 a 3452, (2002).

[10] D.W. Robinson, G.T. Reid, *Interferogram Analysis: Digital Fringe Measurement Techniques* (IOP publishing Ltd., London, 1993).

[11] D. Malacara, M. Servin, Z. Malacara, *Interferogram Analysis for Optical Testing* (Marcel Decker, 1998).

[12] D. Malacara, *Optical Shop Testing* (Wiley, 1992).

[13] K. Creath, in: E. Wolft(Ed.), “Progress in Optics”, VOL 26, Elsevier, Amsterdam, (1988), p. 350.

[14] Creath, in: D. Robinson, G.T. Reid (Eds.), *Interferogram Analysis*, (JOP Publishing Ltd., 1993), p. 94.

[15] Takeda, H. Ina, S. Kobayashi, H. *Opt. Soc. Am.* VOL.72 (1981) 156.

[16] X. Su, W. Chen, *Opt. Laser Eng.* VOL.35 (2001) 263.

[17] K.H. Womack, *Opt. Eng.* VOL.23 (1984) 391.

[18] M. Servin, R. Rodriguez-Vera, *J. Mod. Opt.* VOL.40 (1993) 2087.

[19] D. C. Ghiglia, G.A. Mastin, L.A. Romero, *J. Opt. Soc. Am.* VOL.4 (1987) 267.

[20] X. Su, L. Xue, *Opt. Eng.* VOL.40 (2001) 637.

[21] M. Servin, F.J. Cuevas, D. Malacara, J.L. Marroquin, R. Rodríguez-Vera, *Appl. Opt.* VOL.38 (1999) 1934.

[22] M. Servin, J. L. Marroquin, F.J. Cuevas, *Appl. Opt.* VOL.36 (1997) 4540.

[23] J. Villa, M. Servin, *Opt. Laser Eng.* VOL.31 (1999) 279.

[24] J. A. Quiroga, A. Gonzalez-Cano, *Appl. Opt.* VOL39 (2000) 2931.

[25] F.J. Cuevas, M. Servin, O.N. Stavroudis, R. Rodríguez Vera, *Opt. Común.* VOL.181 (2000) 239.

[26] F.J. Cuevas, M. Servin, R. Rodríguez-Vera, *Opt. Común.* VOL.163 (1999) 270.

Rotated Coin Recognition Using Neural Networks

Adnan Khashman[1], Boran Sekeroglu[2], and Kamil Dimililer[1]

[1] Electrical and Electronic Engineering Department
[2] Computer Engineering Department Near East University, Lefkosa, Mersin 10, Turkey
{amk, bsekeroglu, kdimililer}@neu.edu.tr

Abstract. Neural networks have been used in the development of intelligent recognition systems that simulate our ability recognize patterns. However, rotated objects may cause incorrect identification by recognition systems. Our quick glance provides an overall approximation of a pattern regardless of noise or rotations. This paper proposes that the overall approximation of a pattern can be achieved via pattern averaging prior to training a neural network to recognize that pattern in various rotations. Pattern averaging provides the neural network with "fuzzy" rather than "crisp" representations of the rotated objects, thus, minimizing computational costs and providing the neural network with meaningful learning of various rotations of an object. The proposed method will be used to recognize rotated coins and is implemented to solve an existing problem where slot machines in Europe accept the new Turkish 1 Lira coin as a 2 Euro coin.

Keywords: Neural networks, Pattern Recognition, Coin Identification, Rotational Invariance.

1 Introduction

Computational Intelligence research and applications have gained an increased momentum to model human perception and recognition. While computer vision methods prepare efficient data acquisition and presentation of patterns, neural networks perform the learning of these patterns. However, in complicated recognition tasks, such as recognizing rotated objects, careful and effective data presentation to the neural network is required in order to achieve meaningful learning.

Many intelligent and conventional recognition systems have been developed which are insensitive to different kinds of transformation; including rotation, translation and scale [1],[2],[3]. The use of neural networks for classifying rotated objects was investigated by Fukumi et al. [1] who presented an efficient rotation-invariant recognition system using a back propagation neural network. They described a system based on a rotation-invariant neural network that is capable of identifying circular objects. Rotational invariance is achieved by explicitly generating the rotational group for a coarse model of the circular patterns in a preprocessing step and feeding the results into a neural network. This approach has the advantage of identifying patterns at any degree with an improved recognition accuracy by using circular slabs, however, dividing and determining circular pattern slabs is complex and time consuming [2].

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 290–297, 2007.
springerlink.com

Other works on coin identification also include Adameck et al. method that uses coin surface colour [4] and Nolle et al. method that uses coin pattern edge detection [5]. The use of colour seems to increase the computational costs unnecessarily, whereas edge-based pattern recognition has the problem of noise sensitivity.

More recently, an intelligent coin identification system for distnguishing the 2 Euro coins from the new Turkish 1 Lira coin was proposed [6]. This system used minmal coin image database (using only two 2-Euro coins) and larger coin rotation angles. In order to achieve rotation invariance, further work [3] suggested reducing the rotation angles and increasing the number of euro coin varations. This development and system perfermance improvement was further increased using larger databases (using five 2-Euro coins) in addition to one Turkish 1-Lira coin [7]. The maximum number of euro coin variations is 12 which represents the available officially used 2-Euro coins in the European Union. The problem of increasing the number of euro coin variations during training a neural network is that the intelligent system could be more biased into classifying the output as euro instead of Turkish coin which has only one variation of coin. Additionally, larger database and small angle rotations lead to increase in computational and time cost.

The work in this paper aims at introducing an efficient intelligent rotation-invariant system for coin recognition, using small rotation angles and a large coin image database (using nine euro coins and one Turkish coin), while keeping computational cost to a minimum. The method uses pattern averaging and a back propagation neural network classifier. The proposed method is implemented to solve a real-life problem, which is the incorrect identification by slot machines in Europe of the new 1 Turkish Lira (1-TL) coin as the 2 Euro (2-EURO) coins due to physical similarities between the coins. A 2-EURO coin is roughly worth 4 times the 1-TL. This particular implementation is novel and is intended for use as a support tool to standard physical measurements in slot machines.

This paper is organized as follows: Section 2 presents the rotated coin image database. Section 3 describes the identification system and reviews the two phases of its implementation. Section 4 presents the system implementation results. Finally, Section 5 concludes the work that is presented within this paper.

2 Rotations and Coin Image Database

On 1st of January 2002, the euro banknotes and coins were introduced in 12 Member States of the European Union, with seven banknotes and eight coins [8]. The obverse side of each coin has the same design for all 12 countries in the euro area. The reverse side displays different designs for each country. All coins can be used anywhere in the euro area and the coin with the highest value is the 2-EURO coin. On 1st of January 2005, the new Turkish Lira banknotes and coins were introduced in Turkey, with six banknotes and six coins and the coin with the highest value is the 1-TL coin [9]. Due to physical similarities in diameter, thickness and weight, the Turkish 1-TL coin is identified by slot machines as the 2-EURO coin which has more value [10],[11].

The implementation of the work that is presented within this paper involves distinguishing the 2-EURO coins from the 1-TL coin. *Ten* coins are used for this purpose: *one* 1-TL coin of Turkey and *nine* 2-EURO coins of Belgium-Luxembourg, Netherlands, Belgium, Spain, Austria, Germany, Finland, Italy and France.

Images of the obverse and reverse sides of the ten coins at various rotations were captured using a Creative WebCam (Vista Plus) as shown in Figure 1. The coins were rotated at intervals of (15^o) degrees as shown in Figure 2, and a total of 288 rotated coin images were captured. Rotation by 15^o results in 48 images of the 1-TL coin (24 obverse sides and 24 reverse sides) and 240 images of the 2-EURO coins (24 obverse sides, 216 reverse sides of the nine coins of Belgium-Luxembourg, Netherlands, Belgium, Spain, Austria, Germany, Finland, Italy and France). Table 1 shows the number of coin images obtained using 15^o degree rotation intervals and examples of rotated coins can be seen in Figure 3.

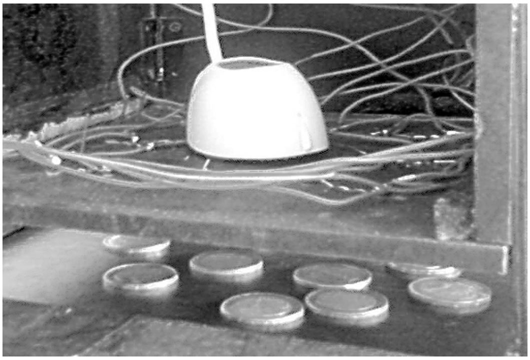


Fig. 1. Coin Rotation and Image Capturing

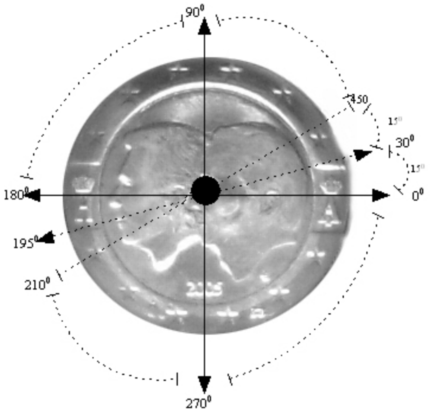
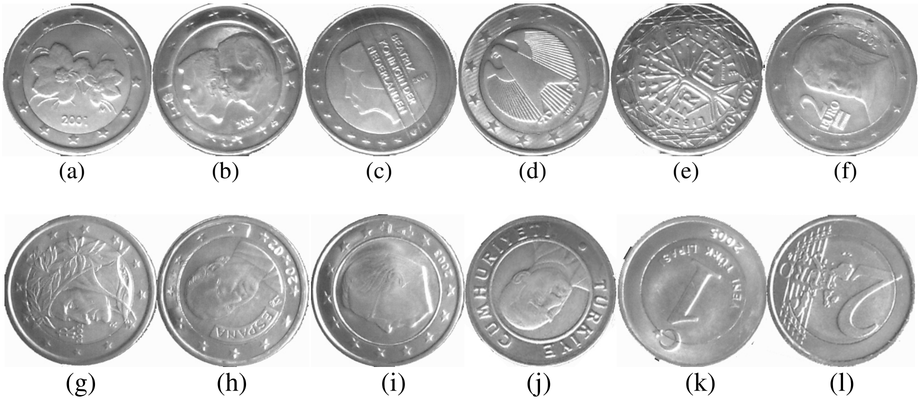


Fig. 2. Rotation Intervals of the Coins

Table 1. Number of Coin Images at 15^0 Rotations

Images	Obverse	Reverse	Total
2-EURO	24	216	240
1-TL	24	24	48
Total	48	72	288



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Fig. 3. Rotated Coins (a) 0^0 Rotated 2-Euro of Finland (b) 15^0 Rotated 2-Euro of Belgium-Luxembourg (c) 30^0 Rotated 2-Euro of Netherlands (d) 45^0 Rotated 2-Euro of Germany (e) 60^0 Rotated 2-Euro of France (f) 75^0 Rotated 2-Euro of Austria (g) 90^0 Rotated 2-Euro of Italy (h) 105^0 Rotated 2-Euro of Spain (i) 120^0 Rotated 2-Euro of Belgium (j) 135^0 Rotated 1-TL of Turkey - Reverse Side (k) 150^0 Rotated 1-TL of Turkey - Obverse Side (l) 165^0 Rotated 2-Euro Common Obverse Side

Captured images of all the coins at (0^o, 90^o, 180^o and 270^o) degrees rotations will be used for training the back propagation neural network, thus, providing 48 coin images for training (8 1-TL and 40 2-EURO coin images). The remaining 240 coin images in the database (40 1-TL and 200 2-EURO coin images) will be used for generalizing the trained neural network. This method of rotation using 15^o degree interval is considered sufficient for all possible rotations of a coin in a slot machine.

3 The Intelligent Identification System

The implementation of the proposed rotation invariant intelligent identification system consists of two phases: an image processing phase where coin images undergo compression, segmentation and pattern averaging in preparation to be presented to the second phase; which is training a back propagation neural network to classify the coin as *1-TL* coin, *2-EURO* coin or *Unidentified* coin. Once the neural network converges and learns, the second phase consists only of one forward pass that yields the identification result.

3.1 Image Processing Phase

The first phase of the proposed system involves preparing the training/generalization data for the neural network. Care must be taken in order to provide the neural network with sufficient data representations of the rotated objects (coins) if we are to achieve meaningful learning, while attempting to keep the computational costs to a minimum.

Image processing is carried out in this phase, where images of the rotated coins are captured in RGB color with the dimensions of 352x288 pixels. These are converted to grayscale which consists of pixel values between 0 and 255. The grey coin images are then cropped to 250x250 pixels images. Thresholding is then applied using a threshold value of 135 as shown by equation 1.

$$T[x,y]=\begin{cases}0, if & P[x,y]\le 135\\ 255, else.\end{cases}, \tag{1}$$

where *T[x,y]* is thresholded image and *P[x,y]* is the grey pixel at coordinates *x* and *y*.

The binary coin image is then compressed to 125x125 pixels using sampling compression as shown in equation 2.

$$C\left[\frac{x}{2},\frac{y}{2}\right] = \underset{\substack{x+2\to x\ \mathrm{dim}\\ y+2\to y\ \mathrm{dim}}}{T\left[x,y\right]}, \tag{2}$$

where *C* represents the compressed image array, *T[x,y]* is thresholded image and *x* and *y* are the pixel coordinates: *x=x+2* and *y=y+2* (initial values for *x* and *y* is *0*).

The compressed image is then trimmed to a 100x100 pixels size image. This is followed by pattern averaging, where segmentation is applied to the image and pixel values within segments are averaged and stored as feature vectors for training the neural network. After various experiments 5x5 segment size was chosen, thus resulting in a 20x20 "fuzzy" bitmap that represents the various coins at various rotations. Other segment sizes can also be used, however, the larger the segment size is, the higher the computational cost will be. A 5x5 segment size results in 20x20 feature vector bitmap, thus requiring 400 neurons in neural network input layer. Pattern averaging can be defined as in equations 3 and 4.

$$PA_i=\sum_{z=i.s_y}^{z+s_y}\sum_{t=i.s_x}^{t+s_x}P(z,t).\frac{1}{D}, \tag{3}$$

$$D=\left(TP_x.TP_y\right)\Big/\left(S_x.S_y\right), \tag{4}$$

where *x* and t; and y and z represents x and y coordinates of image respectively, *i* represents segment number, *P[x,y]* represents pixel value of the pattern, S_x *and* S_y is the x and y coordinates lengths of segments, *D* is total number of each pixel. Figure 4 shows the image processing phase of the coin identification system.

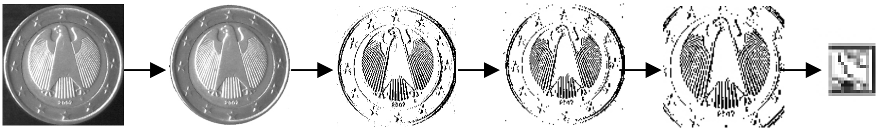


Fig. 4. Image Processing Phase: from original grey coin image to fuzzy feature vector

3.2 Neural Network Implementation Phase

The second phase of the proposed identification system is the implementation of a back propagation neural network classifier. This phase consists of training the neural network using the averaged patterns (feature vectors) obtained from the first phase. Once the network converges (learns), this phase will only comprise generalizing the trained neural network using one forward pass.

A 3-layer feed forward neural network trained by back propagation with 400 input neurons, 25 hidden neurons and 2 output neurons is used to classify the 1-TL and the 2-EURO coins. The number of neurons in the input layer is dictated by the number of averaged segments in the 20x20 bitmap. The choice of 25 neurons in the hidden layer was a result of various training experiments using lower and higher hidden neuron values. The chosen number assured meaningful training while keeping the time cost to a minimum. The two neurons in the output layer represent the 1-TL and the 2-EURO coins, which were represented as binary [1 0] or [0 1] outputs for the Euro or the Turkish Lira respectively; if neither output response is obtained the coin is classified as "Unidentified".

The activation function used for the neurons in the hidden and output layers is the sigmoid function. During the learning phase, initial random weights of values between –0.6 and 0.6 were used. The learning rate and the momentum rate; were

Table 2. Neural Network Final Training Parameters

Input Neurons	Hidden Neurons	Output Neurons	Learning Rate	Momentum Rate	LMS Error	Iterations	Training Time *
400	25	2	0.00099	0.6	0.001	2880	99 seconds

* Using a 2.4 GHz PC with 256 MB of RAM, Windows XP OS and Borland C^{++} compiler.

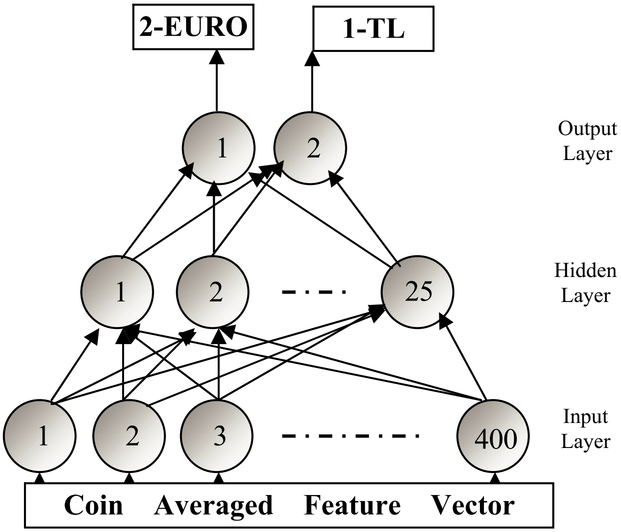


Fig. 5. Neural Network Topology for Rotated Coin Identification System

adjusted during various experiments in order to achieve the required minimum error value of 0.001; which was considered as sufficient for this application. Table 2 shows the final parameters of the trained neural network, whereas Figure 5 shows the topology of the neural network.

The neural network is trained using only 48 coin images of the available 288 coin images. The 48 training images are of rotated coins at (0°, 90°, 180° and 270° degrees) resulting in 8 1-TL coin images (4 obverse and 4 reverse sides) and 40 2-EURO coin images (4 obverse common side, 4 reverse sides of each coin of Germany, France, Spain, Belgium-Luxembourg, Belgium, Austria, Finland, Italy and the Netherlands). The remaining 240 coin images are the testing images which are not exposed to the network during training and shall be used to test the robustness of the trained neural network in identifying the coins despite the rotations.

4 Simulation Results

The two phases of the intelligent identification system were implemented using the C-programming language. The neural network learnt and converged after 2880 iterations and within 99 seconds. The processing time for the generalized neural network after training and using one forward pass, in addition to the image preprocessing phase was a fast 0.03 seconds. The efficiency and speed of this novel rotation invariant coin identification system have been demonstrated through this application.

Coin identification results using the training image set (48 coin images) yielded 100% recognition as would be expected. The system implementation using the testing image sets, that were not previously exposed to the neural network, yielded correct identification of 227 coin images out of the available 240 coin test images, thus achieving a 94.6% recognition rate. Combining testing and training coin image sets, an overall recognition rate of 95.5% has been achieved. This successful result was obtained by only using the 90° rotated coin images for training the neural network. Table 3 shows the coin identification results in details.

Table 3. Intelligent Coin Identification Results

Coins	**2-EURO**		**1-TL**		**BOTH**		
Image Set	Training	Testing	Training	Testing	Training	Testing	Total
Recognition Rate	40/40 (100 %)	198/200 (99 %)	8/8 (100%)	29/40 (72.5 %)	48/48 (100%)	227/240 (94.6 %)	**275/288** **(95.5 %)**

5 Conclusion

This paper presented an intelligent rotation-invariant coin identification system that uses coin surface patterns and a neural network for the identification of rotated coins at intervals of 15° degrees. The system uses image preprocessing in its first phase, where averaged segments of the coin image are obtained and used as the input to the neural network in the second phase. The averaged patterns form fuzzy feature vectors that enable the neural network to learn the various rotations of the objects. Thus, reducing the computational costs and providing a rotation invariant identification system.

A real life application has been successfully implemented, as shown in this paper, to identify the 2-EURO and 1-TL coins where the system can be used as a support tool to standard physical measurements in slot machines. This solves an existing problem where physical similarities between these coins led to slot machine abuse in Europe. Moreover, the developed system is flexible and can thus be trained to recognize other coins.

An overall 95.5% correct identification of both the 2-EURO and the 1-TL coins has been achieved, where 275 out of 288 rotated coin images were correctly identified. The recognition rate for the Euro coin is higher than that for the Turkish Lira due to the higher number of training Euro coin patterns. Rotation by intervals of 15^o degrees provides sufficient coin image database for a robust learning of the neural network within the developed rotation-invariant identification system. These results are very encouraging when considering the time costs. The neural network training time was 99 seconds, whereas the system run time for both phases (image preprocessing and neural network generalization) was 0.03 seconds.

References

1. Fukumi, M., Omatu, S., Takeda, F., Kosaka, T.: Rotation-Invariant Neural Pattern Recognition System with Application to Coin Recognition, IEEE Transactions on Neural Networks, (2) (1992) 272–279
2. Fukumi, M., Omatu, S., Nishikawa, Y.: Rotation Invariant Neural Pattern Recognition System Estimating a Rotation Angle, IEEE Transactions on Neural Networks, 8(3) (1997) 568-581
3. Khashman, A., Sekeroglu, B., Dimililer, K.: Intelligent Rotation-Invariant Coin Identification System. WSEAS Transactions on Signal Processing, 2(5) (2006) 781-786
4. Adameck, M., Hossfeld, M., Eich, M.: Three Color Selective Stereo Gradient Method for Fast Topography Recognition of Metallic Surfaces, Proceedings of Electronic Imaging, Science and Technology, Machine Vision Applications in Industrial Inspection XI, SPIE 5011. (2003) 128–139
5. Nolle, M. et al.: Dagobert – A New Coin Recognition and Sorting System, Proceedings of VIIth Digital Image Computing: Techniques and Applications, (2003) 329-338
6. Khashman, A., Sekeroglu, B., Dimililer, K.: Intelligent Coin Identification System. Proceedings of IEEE International Symposium on Intelligent Control (ISIC'06), Germany (2006)
7. Khashman, A., Sekeroglu, B., Dimililer, K.: ICIS: A Novel Coin Identification System. Lecture Notes in Control and Information Sciences, Vol. 345. Springer-Verlag, Berlin Heidelberg New York (2006) 913-918
8. European Central Bank: Euro Banknotes & Coins http://www.euro.ecb.int/en/section/euro0/coins.html (2005)
9. New Turkish Lira, http://www.tcmb.gov.tr/ytlkampanya/ images/NewTL.jpg (2005)
10. Deutsche Welle, Current Affairs, Currency Confusion Helps Smokers, http://www.dw-world.de/dw/article/0,1564,1477652,00.html (04.02.2005)
11. Verband Turkischer Industrieller und Unternehmer: New Turkish Lira Symbolizes Closeness to Europe, Voices of Germany, Press review compiled by the Berlin Office of TÜSIAD (19.01.2005)

Selected Problems of Intelligent Handwriting Recognition

Wojciech Kacalak[1], Keith Douglas Stuart[2], and Maciej Majewski[1]

[1] Koszalin University of Technology, Faculty of Mechanical Engineering
Raclawicka 15-17, 75-620 Koszalin, Poland
{Wojciech.Kacalak, Maciej.Majewski}@tu.koszalin.pl
[2] Polytechnic University of Valencia, Department of Applied Linguistics, Camino de Vera, s/n, 46022 Valencia, Spain
KStuart@idm.upv.es

Abstract. We propose a new method for handwriting recognition that utilizes geometric features of letters. The paper deals with recognition of isolated handwritten characters using an artificial neural network. The characters are written on a regular sheet of paper using a pen, and then they are captured optically by a scanner and processed to a binary image which is analyzed by a computer. In this paper we present a new method for off-line handwriting recognition and also describe our research and tests performed on the neural network.

Keywords: Handwriting Recognition, Isolated Handwritten Characters, Artificial Neural Networks, Artificial Intelligence.

1 Introduction

Handwriting recognition has been studied for nearly forty years and there are many proposed approaches. The problem is quite complex, and even now there is no single approach that solves it both efficiently and completely in all settings. In the handwriting recognition process in Fig. 1, an image containing text must be appropriately supplied and preprocessed. Next, the text must either undergo segmentation or feature extraction. Small processed pieces of the text will be the result, and these must undergo recognition by the system. Finally, contextual information should be applied to the recognized symbols to verify the result. Artificial neural networks, applied in handwriting recognition, allow for high generalization ability and do not require deep background knowledge and formalization to be able to solve the written language recognition problem.

Handwriting recognition can be divided by its input method into two categories: off-line handwriting recognition and on-line handwriting recognition. For off-line recognition, the writing is usually captured optically by a scanner. For on-line recognition, a digitizer samples the handwriting to time-sequenced pixels as it is being written. Hence, the on-line handwriting signal contains additional time information which is not presented in the off-line signal.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 298–305, 2007.
springerlink.com

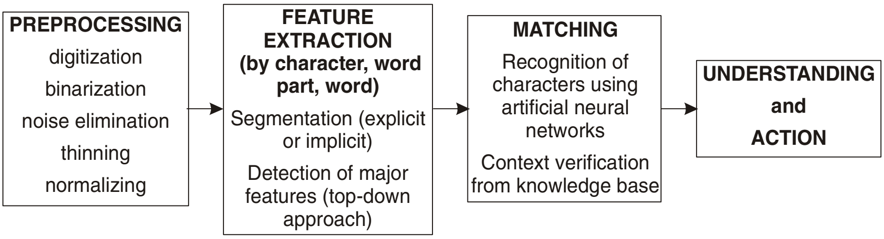

PREPROCESSING
digitization
binarization
noise elimination
thinning
normalizing
FEATURE EXTRACTION
(by character, word part, word)
Segmentation (explicit or implicit)
Detection of major features (top-down approach)
MATCHING
Recognition of characters using artificial neural networks
Context verification from knowledge base
UNDERSTANDING
and
ACTION

Fig. 1. Steps involved in handwriting recognition

2 The State of the Art

The state of the art of automatic recognition of handwriting at the beginning of the new millennium is that as a field it is no longer an esoteric topic on the fringes of information technology, but a mature discipline that has found many commercial uses. On-line systems for handwriting recognition are available in hand-held computers such as personal digital assistants. Their performance is acceptable for processing handprinted symbols, and, when combined with keyboard entry, a powerful method for data entry has been created.

Off-line systems are less accurate than on-line systems. However, they are now good enough that they have a significant economic impact on for specialized domains such as interpreting handwritten postal addresses on envelopes and reading courtesy amounts on bank checks [8,9,14,17].

The success of on-line systems makes it attractive to consider developing off-line systems that first estimate the trajectory of the writing from off-line data and then use on-line recognition algorithms [16]. However, the difficulty of recreating the temporal data [5,6] has led to few such feature extraction systems so far [1].

Research on automated written language recognition dates back several decades. Today, cleanly machine-printed text documents with simple layouts can be recognized reliably by OCR software. There is also some success with handwriting recognition, particularly for isolated handprinted characters and words. For example, in the on-line case, the recently introduced personal digital assistants have practical value. Similarly, some online signature verification systems have been marketed over the last few years and instructional tools to help children learn to write are beginning to emerge. Most of the off-line successes have come in constrained domains, such as postal addresses, bank checks, and census forms. The analysis of documents with complex layouts, recognition of degraded printed text, and the recognition of running handwriting continue to remain largely in the research arena. Some of the major research challenges in on-line or off-line processing of handwriting are in word and line separation, segmentation of words into characters, recognition of words when lexicons are large, and the use of language models in aiding preprocessing and recognition. In most applications, the machine performances are far from being acceptable, although potential users often forget that human subjects generally make reading mistakes [2,3,7].

The design of human-computer interfaces [10,11,12] based on handwriting is part of a tremendous research effort together with speech recognition [13], language processing and translation to facilitate communication of people with computers. From this perspective, any successes or failures in these fields will have a great impact on the evolution of languages [4,15].

3 Method Description

The proposed system attempts to combine two methods for handwriting recognition, neural networks and preprocessing for geometric features extraction. The motivation behind that preprocessor is to reduce the dimensionality of the neural network input. The system consists of the preprocessing subsystem and the neural network subsystem, as shown in Fig. 2. However, another benefit given by the preprocessor is immunity against image translation, because all the information is relative to the image's center of mass in Fig. 3.

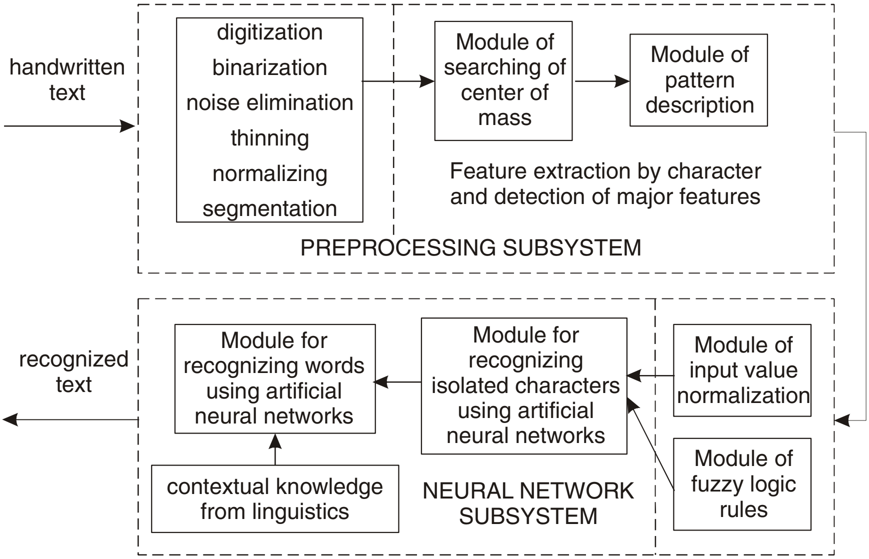

handwritten text
digitization
binarization
noise elimination
thinning
normalizing
segmentation
Module of searching of center of mass
Module of pattern description
Feature extraction by character and detection of major features
PREPROCESSING SUBSYSTEM
recognized text
Module for recognizing words using artificial neural networks
Module for recognizing isolated characters using artificial neural networks
Module of input value normalization
Module of fuzzy logic rules
contextual knowledge from linguistics
NEURAL NETWORK SUBSYSTEM

Fig. 2. The proposed system of handwriting recognition

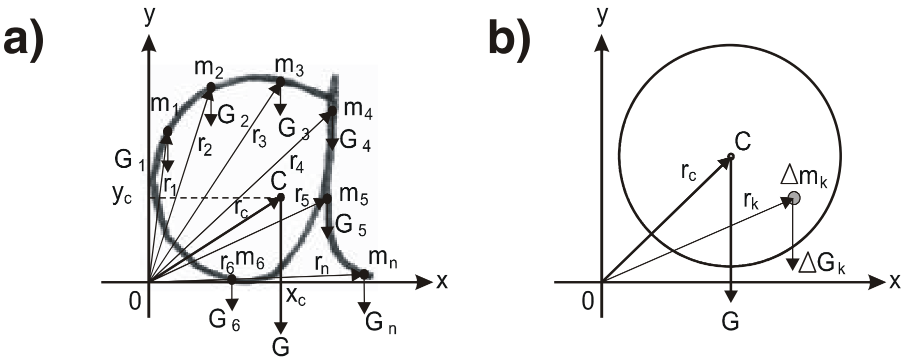


Fig. 3. a) Determining the center of mass of a character, b) determining the approximate position of the center of an optional letter

The handwritten text is made subject to the following preprocessing: digitization, binarization, noise elimination, thinning, normalizing and segmentation. The next step

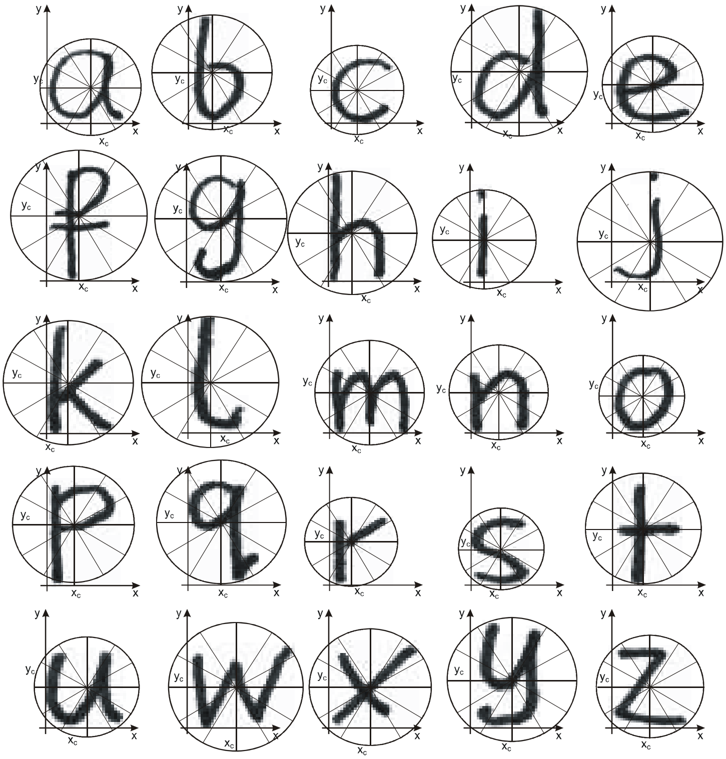


Fig. 4. Geometric analysis of letters for geometric feature extraction

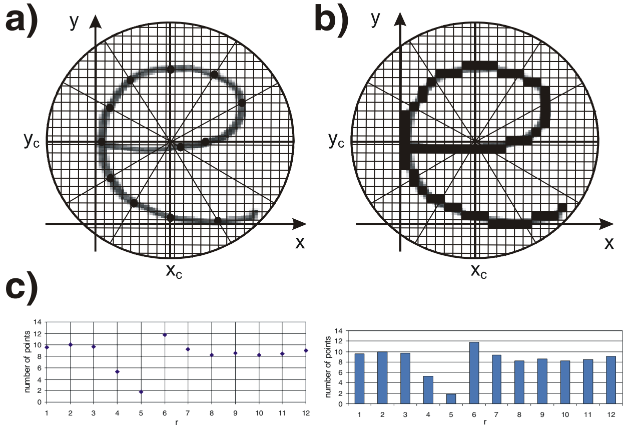


Fig. 5. Scheme of the preprocessing method for geometric feature extraction of characters: a) geometric analysis of letter e; b) pattern description by creating a binary image; c) histograms for letter e

is to find the center of mass of the character image. With the center of mass as a reference point, the vectors are drawn, creating a set of points describing the contour of the character so that its pattern description is made. The neural network training patterns are based on the geometric analysis of letters in Fig. 4. The description patterns of each isolated character in Fig. 5, after the normalization process, are inputs for an artificial neural network.

The letter can be considered as a set of points which mass m_k is constant, and have gravity forces $G_k=m_{k*}g$, and also vectors r_k (1):

$$r_c = \frac{\sum_{k=1}^{n} r_k \Delta G_k}{G} \tag{1}$$

The center of a letter i.e. the center of mass of the character image is found by determining gravity center coordinates (2):

$$x_c = \frac{\sum_{k=1}^{n} x_k \Delta G_k}{G} \qquad y_c = \frac{\sum_{k=1}^{n} y_k \Delta G_k}{G} \tag{2}$$

In (1) and (2) gravity force G is total weight of the set of points (**3**):

$$G = \sum_{k=1}^{n} \Delta G_k \tag{3}$$

4 Research Results

The data set for the system in Fig. 6 was the "Optical Recognition of Handwritten Digits" database, which was originally assembled by the National Institute of Standards and Technology (NIST) in 1994. It consists of handwritten numerals from a total of 43 people, 30 contributed to the training set database and 13 different people to the test set database.

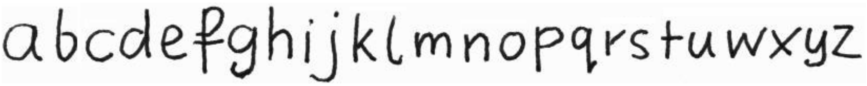


Fig. 6. The data set for this system

In the proposed hybrid system, the description patterns of each isolated character, after the process of input value normalization and application of letter description rules using fuzzy logic, are the input signals for the neural network as presented in Fig. 7.

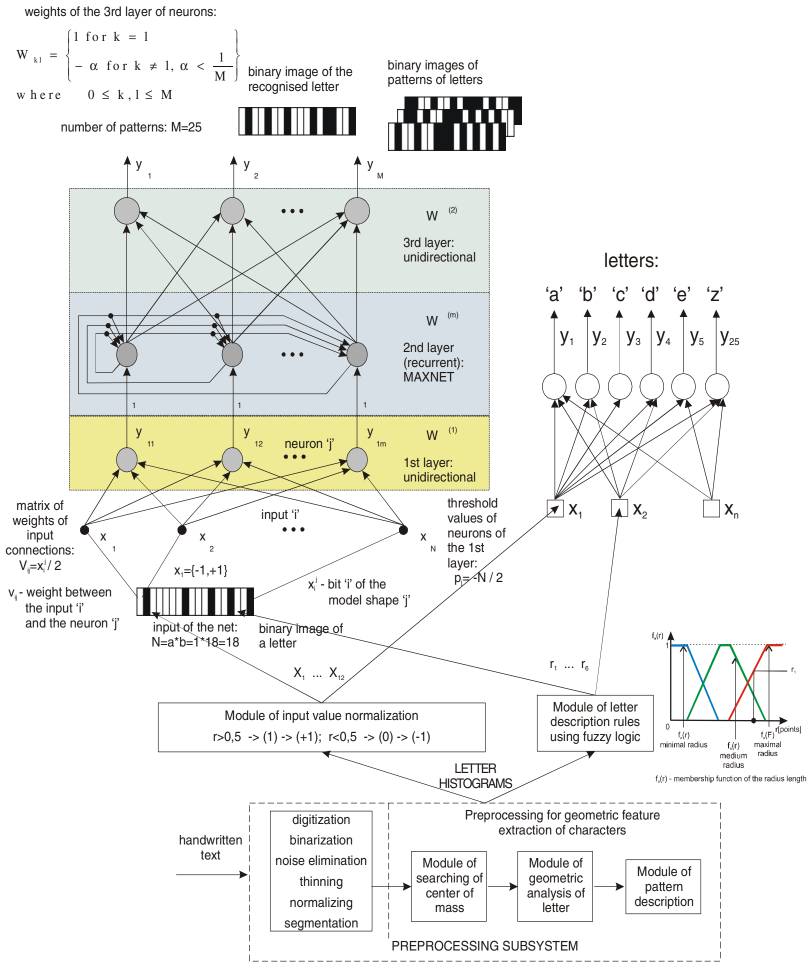

weights of the 3rd layer of neurons:
$W_{kl} = \begin{cases} 1 \text{ for } k = l \\ -\alpha \text{ for } k \neq l, \alpha < \frac{1}{M} \end{cases}$
where $0 \leq k, l \leq M$
number of patterns: M=25
binary image of the recognised letter
binary images of patterns of letters
y_1
y_2
y_M
$W^{(2)}$
3rd layer: unidirectional
$W^{(m)}$
2nd layer (recurrent): MAXNET
y_{11}
y_{12}
y_{1m}
neuron 'j'
$W^{(1)}$
1st layer: unidirectional
matrix of weights of input connections: $V_{ij}=x_i^j/2$
x_1
x_2
x_N
input 'i'
threshold values of neurons of the 1st layer: $p_j=-N/2$
$x_i=\{-1,+1\}$
v_{ij} - weight between the input 'i' and the neuron 'j'
x_i^j - bit 'i' of the model shape 'j'
input of the net: N=a*b=1*18=18
binary image of a letter
X_1 ... X_{12}
letters:
'a' 'b' 'c' 'd' 'e' 'z'
y_1 y_2 y_3 y_4 y_5 y_{25}
X_1
X_2
X_n
r_1 ... r_6
$f_a(r)$
1
0
r_1
r[points]
$f_a(r)$ minimal radius
$f_a(r)$ medium radius
$f_a(F)$ maximal radius
$f_a(r)$ - membership function of the radius length
Module of input value normalization
r>0,5 -> (1) -> (+1); r<0,5 -> (0) -> (-1)
Module of letter description rules using fuzzy logic
LETTER HISTOGRAMS
handwritten text
digitization
binarization
noise elimination
thinning
normalizing
segmentation
Preprocessing for geometric feature extraction of characters
Module of searching of center of mass
Module of geometric analysis of letter
Module of pattern description
PREPROCESSING SUBSYSTEM

Fig. 7. Proposed hybrid system: input signals for the neural network

The simulation set of the recognition of isolated handwritten characters, built for creating and training artificial neural networks is shown in Fig. 8a. The neural networks are trained with the model of isolated written language characters. The applied neural network architecture is presented in Fig. 8b. The networks consist of two layers of neurons with the competitive mechanism. The ability of the neural network to learn to recognize specific letters depends on the number of learning epochs. The specified time of learning enables the network to minimize the error so that it could work more efficiently. Based on the research, the following conclusion has been reached as shown in Fig. 8c. Error rate is about 20% at learning time equals 50 epochs and 5% at 100 epochs. The error rate dropped by about 90% after training with 60 series of all patterns.

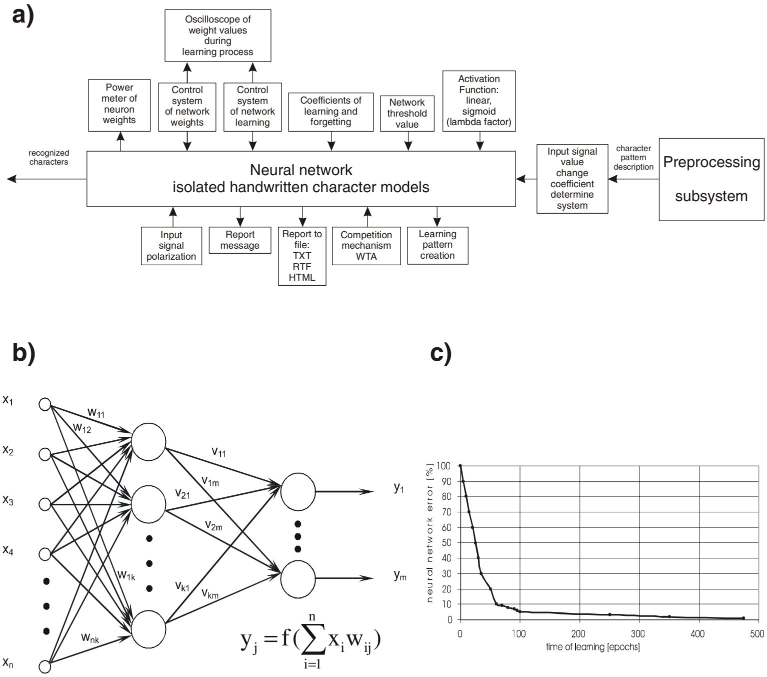

a)
Oscilloscope of weight values during learning process
Power meter of neuron weights
Control system of network weights
Control system of network learning
Coefficients of learning and forgetting
Network threshold value
Activation Function: linear, sigmoid (lambda factor)
recognized characters
Neural network isolated handwritten character models
Input signal value change coefficient determine system
character pattern description
Preprocessing subsystem
Input signal polarization
Report message
Report to file: TXT RTF HTML
Competition mechanism WTA
Learning pattern creation
b)
x_1
x_2
x_3
x_4
x_n
w_{11}
w_{12}
w_{1k}
w_{nk}
v_{11}
v_{1m}
v_{21}
v_{2m}
v_{k1}
v_{km}
y_1
y_m
$y_j = f(\sum_{i=1}^{n} x_i w_{ij})$
c)
neural network error [%]
100
90
80
70
60
50
40
30
20
10
0
0
100
200
300
400
500
time of learning [epochs]

Fig. 8. Neural network simulations of isolated handwritten characters models, neural network architecture and error rate

5 Conclusions and Perspectives

Many advances and changes have occurred in the field of automated written language recognition, over the last decade. The different sources of variability of various psychophysical aspects of the generation and perception of written language make handwriting processing so difficult.

Considerable progress has been made in handwriting recognition technology particularly over the last few years. Handwriting recognition systems have been limited to small and medium vocabulary applications, since most of them often rely on a lexicon during the recognition process. The capability of dealing with large lexicons, however, opens up many more applications.

References

1. Artieres, T., Gauthier, N., Gallinari, P., Dorizzi, B.: A Hidden Markov Models combination framework for handwriting recognition. International Journal on Document Analysis and Recognition, 5(4) (2003). Springer Berlin Heidelberg. 233-243
2. Besner, D., Humphreys, G. W.: Basic Processes in Reading: Visual Word Recognition. Lawrence Erlbaum Associates. Hillsdale, New Jersey (1991)

3. Bishop, C. M.: Neural Networks for Pattern Recognition. Oxford University Press Inc., New York (2004)
4. Bradford, R., Bradford R. B.: An Introduction to Handwriting Examination and Identification. Nelson-Hall Publishers. Chicago (1992)
5. Boccignone, G., Chianese, A., Cordella, L. P., Marcelli, A.: Recovering Dynamic Information from Static Handwriting. Pattern Recognition, 26(3) (1993). Elsevier Science (1993) 409-418
6. Doermann, D. S., Rosenfeld, A.: Recovery of Temporal Information from Static Images of Handwriting. Computer Vision, 15(1-2) (1995). Springer Netherlands. 143-164
7. Dori, D., Bruckstein, A.: Shape, Structure and Pattern Recognition. World Scientific Publishing Co., New Jersey (1995)
8. Kavallieratou, E., Fakotakis, N., Kokkinakis, G.: An unconstrained handwriting recognition system. International Journal on Document Analysis and Recognition, 4(4) 2002. Springer Berlin Heidelberg. 226-242
9. Li, Z. C., Suen, C. Y.: The partition-combination method for recognition of handwritten characters. Pattern Recognition Letters, 21(8) 2000. Elsevier Science. 701-720
10. Majewski, M., Kacalak, W.: Automatic recognition and evaluation of natural language commands. International Symposium on Neural Networks ISNN 2006. Chengdu, China. Lecture Notes in Computer Science 3973: Springer (2006) 1155-1160
11. Majewski, M., Kacalak, W.: Natural language human-machine interface using artificial neural networks. International Symposium on Neural Networks ISNN 2006. Chengdu, China. Lecture Notes in Computer Science 3973: Springer (2006) 1161-1166
12. Majewski, M., Kacalak, W.: Intelligent Layer of Two-Way Voice Communication of the Technological Device with the Operator. International Conference KES 2005. Melbourne, Australia. Lectures Notes in Artificial Intelligence 3683, Springer (2005) 930-936
13. Kacalak, W., Douglas Stuart, K., Majewski, M.: Intelligent Natural Language Processing. International Conference on Natural Computation ICNC2006. Xi'an, China. Lectures Notes in Artificial Intelligence 4221, Springer (2006) 584-587
14. Mitiche, A., Lebidoff, M.: Pattern Classification by a Condensed Neural Network. Neural Networks, 14(4-5) (2001). Elsevier Science. 575-580
15. Mori, S., Nishida, H., Yamada, H.: Optical Character Recognition. John Wiley & Sons, Inc., New York, New Jersey (1999)
16. Nishida, H.: An Approach to Integration of Off-Line and On-Line Recognition of Handwriting. Pattern Recognition Letters, 16(11) 1995. Elsevier Science. 1213-1219
17. Zhou, J., Krzyzak, A., Suen, C. Y.: Verification-a Method of Enhancing the Recognizers of Isolated and Touching Handwritten Numerals. Pattern Recognition, 35(5) 2002. Elsevier Science. 1179-1189

3-D Object Recognition Using an Ultrasonic Sensor Array and Neural Networks

Keeseong Lee

School of Electronics and Electrical Engineering, Hongik University, Seoul, Korea
leeksyh@yahoo.com

Abstract. 3-D object recognition which is independent of translation and rotation using an ultrasonic sensor array, invariant moment vectors, and neural network is presented. With invariant moment vectors of the acquired 16x8 pixel data of square, rectangular, cylindrical, and regular triangular blocks, SOFM (Self Organizing Feature Map) neural network can classify 3-D objects. Invariant moment vectors are constants independent of translation and rotation. The experimental results of the 3-D object recognition using an ultra sensor array are presented to show the effectiveness of the proposed algorithm.

Keywords: 3-D Object Recognition, Ultrasonic Sensor Array, Neural Networks, Invariant Moment Vectors.

1 Introduction

3-Dimensional (3-D) object recognition is considered as one of the important technologies in the field of computer vision. Since the efficiency of industrial robots depends on the working environment conditions, environment recognition can be performed in various conditions where the processing time must be short [1].

In 3-D object recognition the distance information between the sensor and the object is used for the basic information. To measure the distance information, stereovision, laser distance sensor, radar sensor, and ultrasonic sensor may be used. Because 3-D image is recognized by 2-D image by the stereo camera, the distance information is heavily distorted. Due to the distorted information, the complex algorithm and iteration processing are needed to analyze the 2-D image. Since the direction and intensity of the light can influence the recognition rate, it is difficult to recognize the object in the dark environments or the transparent objects [2]. Recognizing objects in dark environment or transparent objects can be solved using the ultrasonic sensor. Because the ultrasonic sensor has the property that the ultrasonic wave is reflected from the surface of object, the ultrasonic sensor can detect the objects in the dark environments. Even though the camera can't detect the transparent objects, the ultrasonic sensor can detect them. Measuring a distance is a simple procedure for ultrasonic sensors. With these advantages, the ultrasonic system can be utilized for recognition of the objects.

For the object recognition, it is required to recognize the object even though there is translation, rotation or both. But the researches by Lee [2] and Watanabe and

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 306–315, 2007.
springerlink.com

Yoneyama [3] only dealt the object recognition without considering the translation and rotation. This reveals the need for extending the recognition algorithm.

A method for 3-D object recognition which is independent of translation and rotation by the ultrasonic sensor array, invariant moment vectors [4][5] and SOFM neural network [2][6] will be presented. The experimental results will be shown to prove the effectiveness of the proposed method.

2 Invariant Moment Vectors

In order to recognize the object independent of translation and rotation, the features of an object must be extracted using the invariant moment vectors method.

M.K. Hu [7] showed how to obtain the moments that are invariant with translation and rotation. The 2-D moment of order $(p+q)$ of an image, computed from the continuous image intensity function $f(x,y)$, is defined as

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p \ y^q \ f(x,y)dxdy \tag{1}$$

where $p,q = 0,1,2,3......$

In equation (1), the zero order moment m_{00} represents a total number of input pattern pixel of the binary input. The first order moments, m_{10} and m_{01}, represent the center of gravity in x- and y-axis, respectively. The second order moments, m_{20}, m_{11} and m_{02} contain the information about size and rotation of the pattern.

If the input pattern $f(x,y)$ is binary number, equation (1) can be expressed as follows:

$$m_{pq} = \sum_{(x,y)\in S} x^p \ y^q \tag{2}$$

where $S = \{(x,y) \mid f(x,y) = 1\}$

The components of moment in equation (2) are not invariant with translation of input pattern. If the first order moment is used, the center of gravity of the object, $(\bar{x},\bar{y})$, is

$$\bar{x} = \frac{m_{10}}{m_{00}}, \ \bar{y} = \frac{m_{01}}{m_{00}} \tag{3}$$

In order to obtain invariant features with respect to the translation, the central moment μ_{pq} must be calculated with the coordinates of center of gravity for the input pattern.

$$\mu_{pq} = \sum_{(x,y)\in S} (x-\bar{x})^p (y-\bar{y})^q \tag{4}$$

In addition, the invariant moment with the variety of size can be obtained by converting the central moment μ_{pq} into the normalized moment as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{where } \gamma = \frac{p+q}{2} + 1 \tag{5}$$

If the second and third order moments are used, a set of seven invariant moment vectors can be obtained [4]. These values are invariant with translation, rotation and variety of size of objects. The set of seven invariant moments are

$$\phi_1 = \eta_{20} + \eta_{02} \tag{6}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{7}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{8}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{9}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{10}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned} \tag{11}$$

$$\begin{aligned}\phi_7 = {} & 3(\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{12}$$

The features of an object, $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7$ will be used for the inputs of SOFM neural network.

3 SOFM (Self Organizing Feature Map) Neural Network

The SOFM neural network is a two-layer network. The first layer of network is the input layer. The second layer, the competitive layer, is organized as a 2-D grid. All the interconnections start from the first layer to the second; two layers are fully interconnected so that each input unit is connected to all of the units in the competitive layer. With the 2-dimensional grid system the output neuron space is N x N square and the input vector is constructed as V-Dimension.

The winning neuron is determined based on the similarity measures between input vector and the weight vectors. The method to determine the winning neuron uses the Euclid distance as follows;

$$d[x, w_{j(c)}(t)] = \min_{1 \le j \le M} \sum \{d[x - w_{j(c)}(t)]\} \tag{13}$$

where x, $w_{j(c)}(t)$ and M are the input vector, the weight vector which is corresponding to j^{th} output neuron, and the number of output neuron, respectively. Among the output neurons, winning neuron $j(c)$ is determined by selecting a neuron j that has the minimum distance d.

The learning algorithm of the neural network updates not only the weight vector of the winning neuron but also the weight vector of all the neurons in the winning neuron's neighborhood. In figure 1, an example of topological relation for winning neuron is shown. $N_{j(c)}(t)$ represents the collection of neighbor neuron for winning neuron $j(c)$ in time t. The learning rule is as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) N_{j(c)}(t)[x_i(t) - w_{ij}(t)], \quad j \in N_{j(c)}(t) \tag{14}$$

$$\alpha(t) = 0.9\left(1 - \frac{t}{\text{Number of Iterations}}\right) \tag{15}$$

where $w_{ij}(t)$, $x_i(t)$ and $\alpha(t)$ are the weight vector between input neuron i and output neuron j, the input vector, and the learning rate that ranges from 0 to 1, respectively. The neighborhood size of winning neuron, $N_{j(c)}(t)$ is decreased from $N_j = 3$ to 0 as the time elapses shown in figure 1. In the first step of learning, all the neurons may be included and almost all the neurons are learned for each pattern. The neighborhood size is decreased as the learning procedures progress. Only the weight vector of winning neuron will be adjusted finally.

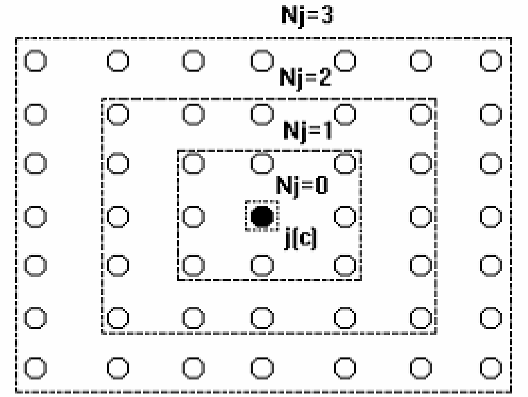


Fig. 1. Topological Neighborhood around Winning Neuron

4 Experimental Setup and Method

4.1 Experimental Setup

The experimental setup is composed of ultrasonic sensor array, sensor driving part and control part. The ultrasonic sensor array is comprised of 8 ultrasonic sensors that are located in the straight line with the interval of 5 cm. Using this ultrasonic array, the 8 x 1 pixel information can be obtained. With stepper motor and motor driver, 16 x 8 pixel information can be obtained by moving the ultrasonic sensor array by 16 steps. The sensor array is moved by 2.5 cm each step.

The ultrasonic sensor used in the setup is Electrostatic Transducer [9] made by Polaroid. The specifications are:

Diameter: 3.5 cm
Input peak-to-peak voltage: 380 V
Sending and receiving frequency: 50 KHz

The sensor driving part performs the following functions: 1) it sends the ultrasonic wave, 2) it receives the echo signal and 3) it measures the distance by amplifying the echo signal, eliminating the noise and converting into the digital signal. The one-pulse echo mode is used due to the short processing time and its high reliability.

The control part is 8751 microprocessor and performs the following functions: 1) it independently controls the sensor to reduce the multiple-reflection and interference between the sensors and 2) it controls the sending and receiving time of the sensors. After compensating the disturbances by the changes of base height and temperature, the distance information is sent to the PC. The experimental setup is illustrated in figure 2.

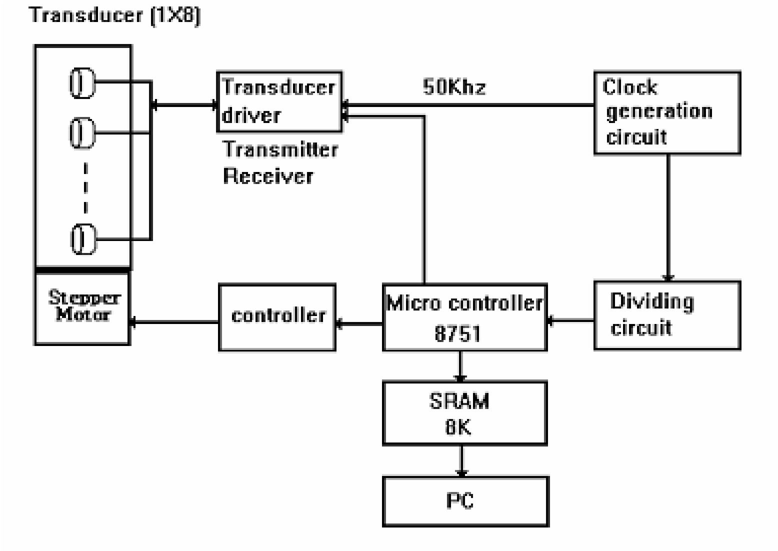


Fig. 2. Experimental Setup

4.2 Experimental Method

In the experiment it is assumed that the ultrasonic wave propagates to the straight direction only. The distance information may be contaminated by the errors resulting from: 1) ultrasonic wave is diminished in the air, 2) the transmitting speed is changed due to temperature changes and 3) there is a variation from the surface of the object and the bending degree of the ultrasonic sensor array. Even though the controller compensates the errors, the neural network can also compensate the errors to obtain the high reliability of the data and perform the preprocessing procedures.

With the preprocessed 16 x 8 pixel information, the invariant moment vectors are obtained. The invariant moment method is used to obtain the features that are independent of the translation and rotation of the object. Table 1 depicts the patterns of the experimental object.

For the experiment, the 26 patterns are used: 6 square, 8 rectangle, 5 cylindrical and 7 regular triangle blocks. By engaging the total of 130 data of 30 square, 40 rectangle, 25 cylindrical and 35 regular triangle blocks and applying equation (6)-(12), a set of seven invariant moment vectors are obtained. The 78 data are used for the training data and 52 data are used for the test data. SOFM neural network is trained with 7 input vectors and N x N output neuron space for recognizing 4 kinds of object. By observing the size of output neuron space, the numbers of iterations and the recognition rate, the size of output neuron space and the number of iteration in the each neuron space are adjusted from 2x2 to 10x10 and from 10 to 50, respectively. To compare the cases whether invariant moment method is used or not, the preprocessed 16 x 8 pixel distance information is trained using the neural network for each case. In the experiment the 128 dimension of input vector V is used.

Table 1. Patterns of the Experimental Objects

Displacement Objects	Base	Translation				Rotation		
		Dis-placement Ob-jectBase Left 2㎝	Right 2㎝	Up 2㎝	Down 2㎝	45°	90°	135°
Square Block W6.3×L6.3×H1.5	◦	◦	◦	◦	◦	◦	×	×
Rectangular Block W7×L11×H1.5	◦	◦	◦	◦	◦	◦	◦	◦
Cylindrical Block D7.5×H1.5	◦	◦	◦	◦	◦	×	×	×
Regular Triangular Block L7×H1.5	◦	◦	◦	◦	◦	◦	◦	×

(◦ : Pattern, × : No Pattern)

5 Experimental Results

When the objects are placed in the center point and with translation and/or rotation, the invariant moment vectors for square, rectangle, cylindrical, and regular triangle

Table 2. Invariant Moment Vectors for Square Block

Invariant	Base	2㎝	2㎝	2㎝	2㎝	45°
Vector		Left	Right	Up	Down	Rotation
ψ_1	0.162835	0.162835	0.162835	0.161925	0.162835	0.164352
ψ_2	0.000206	0.000206	0.000206	0.000526	0.000206	0.000322
ψ_3	0.000078	0.000078	0.000078	0.000016	0.000078	0.0
ψ_4	0.0	0.0	0.0	0.0	0.0	0.0
ψ_5	0.0	0.0	0.0	0.0	0.0	0.0
ψ_6	0.0	0.0	0.0	0.0	0.0	0.0
ψ_7	0.0	0.0	0.0	0.0	0.0	0.0

Table 3. Invariant Moment Vectors for Rectangular Block

Invariant	Base	2㎝	2㎝	2㎝	2㎝	45°	90°
Vector		Left	Right	Up	Down	Rotation	Rotation
ψ_1	0.172389	0.173323	0.172389	0.171476	0.171376	0.173700	0.171376
ψ_2	0.003017	0.003277	0.003017	0.003080	0.002906	0.004076	0.002906
ψ_3	0.000032	0.000059	0.000032	0.000007	0.0	0.0	0.0
ψ_4	0.000001	0.000001	0.0	0.0	0.0	0.0	0.0
ψ_5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ψ_6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ψ_7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4. Invariant Moment Vectors for Regular Triangular Block

Invariant	Base	2㎝	2㎝	2㎝	45°	90°
Vector		Left	Up	Down	Rotation	Rotation
ψ_1	0.249000	0.225000	0.273000	0.249000	0.210648	0.249000
ψ_2	0.040401	0.030625	0.052929	0.040401	0.018460	0.040401
ψ_3	0.001558	0.0	0.002898	0.001558	0.000826	0.001558
ψ_4	0.000124	0.0	0.001311	0.000124	0.000216	0.000124
ψ_5	0.0	0.0	0.000003	0.0	0.0	0.0
ψ_6	0.000025	0.0	0.000286	0.000025	0.000022	0.00025
ψ_7	0.0	0.0	0.000001	0.0	0.0	0.0

blocks are shown in table 2 - 4, respectively. Because the invariant moment vectors are maintained within certain range of values, the invariant moment vectors have the features for recognizing the objects.

In table 5 the recognition rates with the SOFM neural network are shown when 1) with the invariant moment vectors and 2) without invariant moment vectors. For two cases, 16 x 8 pixel is used and number of iteration is 10.

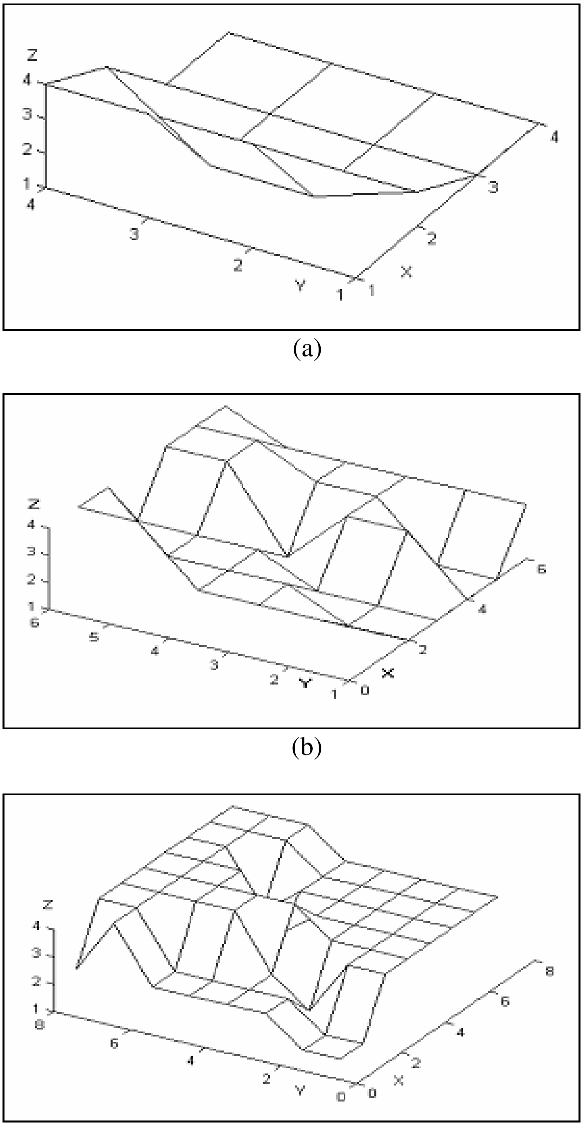
As shown in table 5, when the invariant moment vectors are not used, the recognition rates for the training data and the testing data are 55.1% and 17.3%, respectively. It is obvious that the method without invariant moment vectors is not applicable for object recognition. But with the invariant moment vectors the recognition rates for the training data and testing data are 96.15% and 92.3%, respectively. It is shown that the method with the invariant moment vectors is an effective method for object recognition. The possible cause for the small error is generated from the temperature variation and digital data conversions.

Table 5. Recognition Rates with and without Invariant Moment Vectors

Recognition Rates Objects	with invariant moment vector		without invariant moment vector	
	Recognition rates Object Training Data	Test Data	Training Data	Test Data
Square Block	94.44%	91.66%	66.66%	25.0%
Rectangular Block	95.83%	93.75%	62.5%	18.75%
Cylindrical Block	93.33%	90.0%	46.66%	10.0%
Regular Triangular Block	100%	92.85%	42.80%	14.28%

The maps of SOFM neural network when the output neuron spaces are 4x4, 6x6 and 8x8 after 10 iterations are shown in Figure 3(a), (b), and (c), respectively.

The recognition rates are unchanged when the number of iteration is changed for each output neuron space in figure 3. Even though the output neuron space of SOFM neural network and the number of iteration are adjusted from 4x4 to 10x10 and from 10 to 50, respectively, the recognition rates are constant except when 2 x 2 neuron space is used. The recognition rates for total testing data are all 92.3%. In case of 2 x 2 output neuron space, the recognition result is not acceptable because of the inconsistency between the distribution of input data and the size of the output neuron space.



(a)

(b)

(c)

(a) Neuron Space 4 x 4 (b) Neuron Space 6 x 6 (c) Neuron Space 8 x 8

Fig. 3. Map of SOFM Neural Network with Different Output Neuron Space after 10 Iterations of the Testing Data

6 Conclusions

An object recognition technique with an ultrasonic sensor array is proposed. To recognize the objects that are independent of the translation and rotation, the invariant moment vectors are used. A set of seven invariant moment vectors is utilized as an input of the SOFM neural network. The experiments were performed to find the relation in the size of output neuron spaces, the number of iteration and the recognition rate. Because the ultrasonic sensor uses the time of flight, it can recognize the object without concerning the intensity of light. This means that one can recognize the objects without lights. Also the transparent object can be recognized because it can be reflected by the ultrasonic wave. It is also shown that the 3-D object recognition can be performed without the complex mathematical modeling or calculations.

With a simple processing of input data and short training time, the recognition rate with ultrasonic sensors is 92.3% even though there are translation and rotation. When the output neuron space of the SOFM neural network is varied from 2x2 to 10x10, the recognition rate is same for all the case except 2x2 case. Also, when the number of iteration is varied from 10 to 50, the recognition rates are constant for all cases. For the case without the invariant moment vectors, the recognition rate is only 55.12%. Counting much higher recognition rate, the invariant moment vector method must be used to recognize the objects for ultrasonic sensors.

The advantages of the ultrasonic sensor are low cost, the simple circuits, and simple mathematical models. The use of ultrasonic sensor will benefit in the field of 3-D object recognition using the neural network and invariant moment vectors.

References

1. A. Pugh, *Robot Sensors,* Vol. 2. IFS Ltd. (1986) 261-285
2. K. Lee, “3-D Object Recognition and Restoration Using an ultrasonic Sensor Array,” *Journal of KIEE,* Vol. 44, No. 5, (1995) 671-677
3. S. Watanabe and M. Yoneyama, "An Ultrasonic Visual Sensor for Three Dimensional Object Recognition using Neural Network," *IEEE Trans. on Robotics and Automation,* Vol. 8, No. 2, (1992) 240-249
4. C.H. Teh and R.T. Chin, "On Digital Approximation of Moment Invariants," *Computer Vision, Graphics, and Image Processing* 33 (1986) 318-326
5. R.Y. Wong and E.L. Hall, "Scene Matching with Invariant Moments," *Computer Vision, Graphics, and Image Processing* 8, (1978) 16-24
6. J.E. Dayhoff, *Neural Network Architectures,* Van Nostrand Reinhold (1990) 163-191
7. M.K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Information Theory,* Vol. 8, (1962) 179-187
8. T.K Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag (1994)
9. *Ultrasonic Ranging System Handbook,* Polaroid Corporation (1980)

Soft System for Road Sign Detection

Bogusław Cyganek

AGH - University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@uci.agh.edu.pl

Abstract. In this paper a fuzzy system for detection of the triangular and rectangular traffic signs is presented. In many sign recognition systems reliable and fast shape detection is a prerequisite for successful classification. The proposed method operates on colour images in which it detects the characteristic points of signs by sets of fuzzy rules. These points are used then for extraction of the shapes that fulfil the fuzzy verification rules. The method allows very accurate and real-time detection of the planar triangles, inverted triangles, rectangles, and diamond shapes. The presented detector is a part of a driver-assisting-system for recognition of the road signs. The experimental results verify the method accuracy and robustness.

1 Introduction

Computers can help to increase safety on our roads. The driver-assisting-systems (DAS) can assist drivers to properly react to the road conditions. A road sign (RS) recognition system constitutes a vital part of DAS [4]. It can remind a driver of the passing RSs and verify her or his reaction on some of them. For instance, a detected speed limit or a “STOP” sign can be checked against the current speed of a vehicle. Usually the RS recognition systems at first detect shapes of interest, and then do signs classification. However, it is also possible to perform the two actions at once, as e.g. in the RS recognition system based on the matching in the log-polar space [1]. An interesting method of RS detection has been proposed by Escalera et.al. [6]. It relies on detection in the HSI space. Then the recognition is done based on the energy functional. Another work by Gavrila presents the RSs detection from monochrome images based on the distance transform [9]. The CIECAM97 colour space for RS detection is proposed by Gao et.al. [8]. A fuzzy approach to RS shape determination presents Fleyeh et.al. [7]. Their system is based on a set of fuzzy rules for object detection from the colour images. For other systems one can refer to [11][12][13]. In this paper we propose a fuzzy system which finds the triangular and rectangular shaped Polish RSs. The problem is difficult for real scenes since RSs can have different size and tint, can be tilted and rotated, or can be worn out due to weather conditions At first the fuzzy colour segmentation is performed and the salient points are found in the HSI colour space. Then the detected points are used to build shape hypothesis from which only the ones which have been verified with the RS fuzzy shape templates are selected. The method is fast and accurate.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 316–326, 2007.
springerlink.com

2 Architecture of the Road Signs Detection System

Fig. 1 depicts the architecture of the fuzzy RS detection system. The processing starts with the colour image acquisition followed by an optional low pass filtering for noise removal. In some setups this stage will also adjust the input image to a desired resolution. In our system the Olympus C-70 and the Marlin F-033C camera by Allied

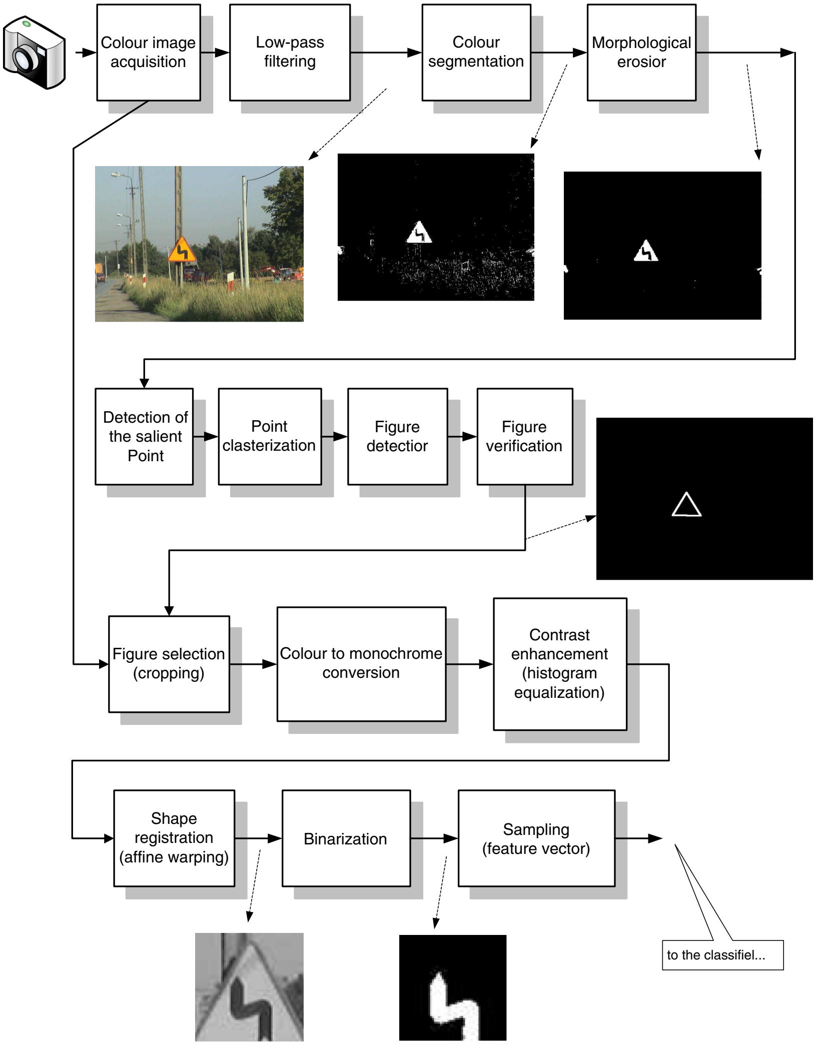
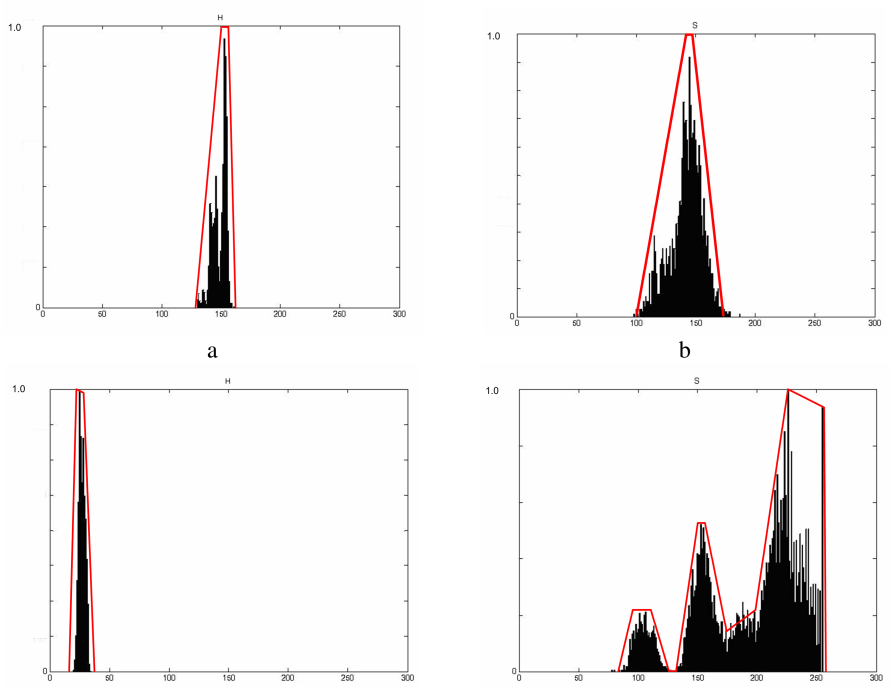

Colour image acquisition
Low-pass filtering
Colour segmentation
Morphological erosior
Detection of the salient Point
Point clasterization
Figure detectior
Figure verification
Figure selection (cropping)
Colour to monochrome conversion
Contrast enhancement (histogram equalization)
Shape registration (affine warping)
Binarization
Sampling (feature vector)
to the classifiel...

Fig. 1. Architecture of the fuzzy RSs detector. Triangular, inverted triangular, rectangular, and diamond shaped signs can be detected. Further classification is done by the neural network [3].



a b

c d

Fig. 2. Histograms of the H and S channels of the road signs (black) and their fuzzy membership functions (red). Blue colour (rectangular RS, group “D”) hue (a), saturation (b). Yellow colour (triangular RS, group “A”), hue (c), multimodal saturation (d).

Vision Technologies were used. The first one operates over the USB connection, while the latter through the IEEE 1394 FireWire, which is much faster and thus preferable. However, the Olympus camera has wider zoom which helps in some situations. The colour segmentation stage follows. Its purpose is to partition an image into areas of road signs and the background (i.e. all other objects).

Table 1. Definitions of the fuzzy membership functions of Fig. 2 for RS colour segmentation

Attribute	Piecewise-linear membership functions – coordinates (x,y)
Blue H (H_B)	(125, 0) - (145, 1) - (150, 1) - (160, 0)
Blue S (S_B)	(100, 0) - (145, 1) - (152, 1) - (180, 0)
Yellow H (H_Y)	(20, 0) - (23, 1) - (33, 1) - (39, 0)
Yellow S (S_Y)	(80, 0) - (95, 0.22) - (115, 0.22) - (125, 0) - (128, 0) - (150, 0.48) - (155, 0.48) - (175, 0.18) - (200, 0.22) - (225, 1) - (249, 0.95) - (251, 0)

The segmentation is done by fuzzy reasoning based on the membership functions defined for the channels H and S of the HSI space. They are based on the experimentally found histograms (Fig. 2) of the blue and yellow dye encountered in the information “D” and warning “A” signs, respectively. For this purpose a special application was built that allows selection of the interest areas from many hundreds of road scenes. This operation is performed by a human operator, then the histograms in different colour spaces are computed. We experimented with the RGB, HSI, YCrCb, and CIELAB. However, the best results in terms of computation time were obtained with the HSI space. Especially the hue channel (H) was shown to be fairly invariant to the varying illumination conditions, as well as to the figure deformation, shadows and highlights when the incident illumination is white [7].

The following fuzzy rules are used for pixel segmentation:

$$\begin{array}{ll} R_1: & \text{IF } H_B=high \text{ AND } S_B=high \text{ THEN } P_B=high; \\ R_2: & \text{IF } H_Y=high \text{ AND } S_Y=high \text{ THEN } P_Y=high; \end{array} \tag{1}$$

where the membership functions μ_{HB}, μ_{SB}, μ_{HY}, and μ_{SY} are given in and Table 1. P_B and P_Y are fuzzy measures denoting a degree of pixels membership into one of the two classes: blue and yellow. For the reasoning we used the two common methods of Mamdami and Larsen, respectively, given as follows [10][5]:

$$\begin{aligned} \mu_{A\Rightarrow B}(x,y) &= \min(\mu_A(x), \mu_B(y)) \\ \mu_{A\Rightarrow B}(x,y) &= \mu_A(x)\mu_B(y) \end{aligned} \tag{2}$$

Both were tested and performed well in our system. However, in practice the first one is preferred since it avoids the multiplication. Thus, P_B and P_Y follow first rule in (2).

The binary images after segmentation are morphologically eroded with a structural element which was set to remove small separated blobs (*5×5* square in our experiments). Then the fuzzy detection of the salient points, which are specific vertices of the signs, and figure verification are performed (see the next sections).

It is interesting to note that the classification process is also *soft* and done by the ensemble of Hamming neural networks operating on deformable pictograms [2][3].

3 Detection of the Characteristic Points

The characteristic points for the triangular or rectangular shapes are their corner points. Positions of three of them are sufficient for unique localization of the whole object they belong to. This feature is used in our system for detection of the signs for which the corner points can be found. Certainly, for the circular objects the set of salient points is more ample and contains their edges.

The salient points are sought in the colour segmented binary images. Thus, the technique is very fast. To assess a point we need to analyze its neighbourhood. This analysis is done with the symmetrical detector of local binary features (SDLBF). It is composed of four equal rectangular panes, centred at a point of interest P_C – see Fig. 3a. For a discrete pixel grid, this central point takes on a virtual position which does not coincide with the grid. Thus, SDLBF is anchored in a real point P_C on a discrete grid. SDLBF consists of four panes W_0, W_1, W_2, and W_3 - Fig. **3**a – each of size $h_i{\times}v_i$. Detection with a SDLBF is done by counting number of set pixels in each pane. Thus, for each point we obtain a set of four counters c_0, c_1, c_2, and c_3. Their values are then compared with the predefined templates for salient points. If a match is found then a point P_C is classified as a salient point. Each pane W_i is additionally divided alongside its diagonal into two partitions. Thus, a SDLBF contains eight segments.

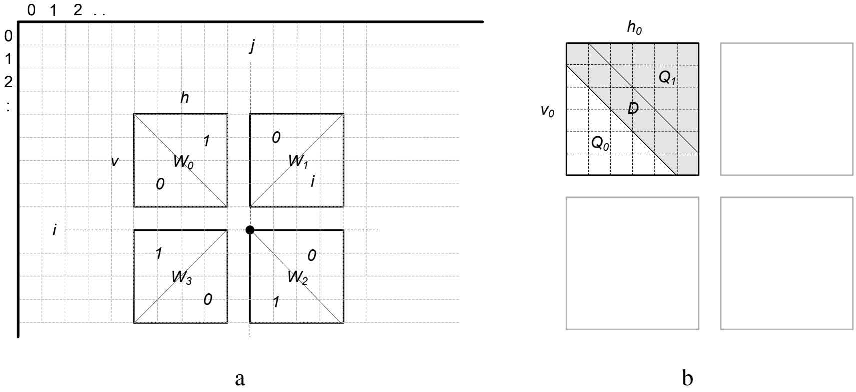


Fig. 3. A detector of local binary features (a). Partitioning of a single pane in the SDLBF (b).

Fig. 3b depicts a detailed partitioning into regions Q_0 and Q_1 of a single pane in the SDLBF detector. However, this partitioning is not symmetrical since one of the parts contains diagonal elements *D*. For instance, for a *9×9* pane we have *81* detection elements, from which *36* belongs to Q_0 and *36+9=45* to Q_1. Thus, SDLBF can be seen as a circular detector set around a central point. Detection of characteristic points relies on setting pixel membership values for each pane.

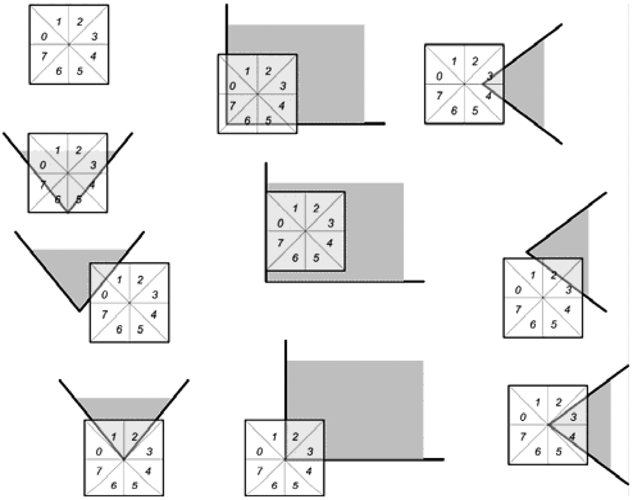


Fig. 4. Detection of salient points with the SDLBF. Type of a central point is determined based on the counted number of pixels of an object (crossed area) belonging to each pane.

Fig. 4 shows some examples of the detection of different characteristic points with the SDLBF. For instance, if the panes *1* and *2* are almost entirely filled while the other are empty then we infer a bottom corner points for a reversed triangle (left image in the last row in Fig. 4). Similarly, if the panes *2* and *3* are set whereas the rest is (almost) empty then the point corresponds to the lower bottom corner of a rectangle (middle image in the last row in Fig. 4). In practice, SDLBF showed to be very robust, both in terms of accuracy and speed.

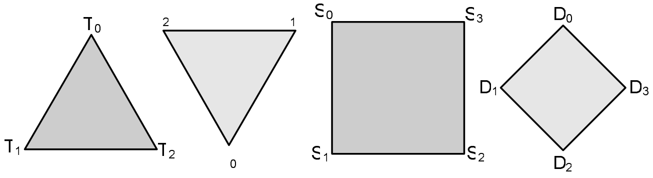


Fig. 5. Salient point definitions for road sign shapes. Point names are used in fuzzy rules.

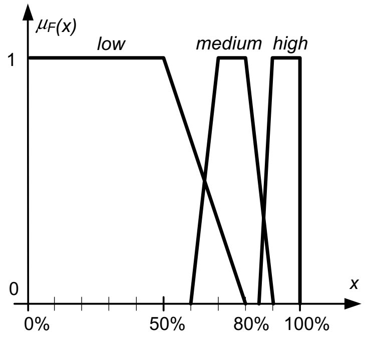


Fig. 6. SDLBF fill ratio membership function

The necessary fill ratios for each pane are governed by the fuzzy membership function in Fig. 6 which resolves one of the three states *{low, medium, high}* fill ratios, depending on the number of set pixels to the total capacity of a pane. For detection of different RS salient points – denoted in Fig. 5 – we designed a set of fuzzy rules (3)-(6) for shapes in Fig. 5.

The binary result *{low, high}* tells whether a given pixel belongs to one of the characteristic points.

R$_3$: IF Q_5= *medium* AND Q_6= *medium* AND Q_{other}=*low* THEN T_0=*high*;

R$_4$: IF Q_2=*medium* AND Q_3= *high* AND Q_{other}=*low* THEN T_1=*high*;

R$_5$: IF Q_0= *high* AND Q_1= *medium* AND Q_{other}=*low* THEN T_2=*high*;

(3)

R$_6$: IF Q_1= *medium* AND Q_1= *medium* AND Q_{other}=*low* THEN I_0=*high*;

R$_7$: IF Q_6= *medium* AND Q_7= *high* AND Q_{other}=*low* THEN I_1=*high*;

R$_8$: IF Q_4=*hig*h AND Q_5= *medium* AND Q_{other}=*low* THEN I_2=*high*;

(4)

$$\begin{aligned}
&\mathrm{R}_9: && \text{IF } Q_4=\mathit{high} \text{ AND } Q_5=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } S_0=\mathit{high};\\
&\mathrm{R}_{10}: && \text{IF } Q_2=\mathit{high} \text{ AND } Q_3=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } S_1=\mathit{high};\\
&\mathrm{R}_{11}: && \text{IF } Q_0=\mathit{high} \text{ AND } Q_1=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } S_2=\mathit{high};\\
&\mathrm{R}_{12}: && \text{IF } Q_6=\mathit{high} \text{ AND } Q_7=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } S_3=\mathit{high};
\end{aligned}\tag{5}$$

$$\begin{aligned}
&\mathrm{R}_{13}: && \text{IF } Q_5=\mathit{high} \text{ AND } Q_6=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } D_0=\mathit{high};\\
&\mathrm{R}_{14}: && \text{IF } Q_3=\mathit{high} \text{ AND } Q_4=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } D_1=\mathit{high};\\
&\mathrm{R}_{15}: && \text{IF } Q_1=\mathit{high} \text{ AND } Q_2=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } D_2=\mathit{high};\\
&\mathrm{R}_{16}: && \text{IF } Q_0=\mathit{high} \text{ AND } Q_7=\mathit{high} \text{ AND } Q_{other}=\mathit{low} \text{ THEN } D_3=\mathit{high};
\end{aligned}\tag{6}$$

Sometimes many neighbouring pixels get the same saliency label. Therefore, for such clouds of points we find their centres which then substitute such group of points.

4 Fuzzy Rules for Figure Verification

Triangular, inverted triangular, rectangular, and diamond shaped RSs can be detected with the presented technique. Figure detection relies on checking *all possible configurations* of their characteristic points. Then the fuzzy figure verification follows, which accepts only those which fulfil the allowable shape definitions for RSs. For triangles we check whether a candidate figure is equilateral and whether its base is not excessively tilted. The former condition is checked by measuring lengths of sides. The latter by measuring slope of the base side. Details are fairly straightforward and omitted here due to limited space. Similar conditions are set and checked for other shapes as well.

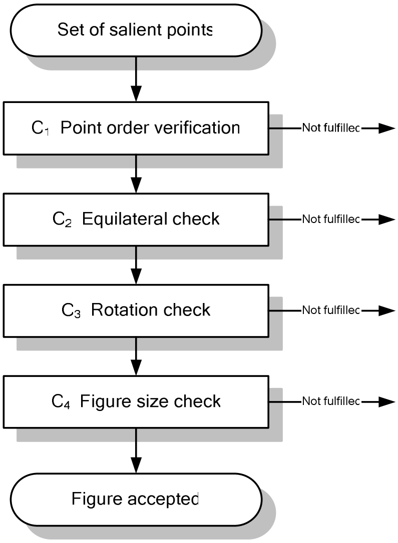


Fig. 7. Fuzzy figure verification process

If the following fuzzy rule is fulfilled then a figure measure F is high, as follows:

$$R_{17}: \text{ IF } C_1=high \text{ AND } C_2=high \text{ AND } C_3=high \text{ AND } C_4=high \text{ THEN } F=high; \qquad (7)$$

Thus, R_{17} has four input and one output fuzzy variables. However, the verification rules are very fast in computations.

5 Experimental Results

The system was implemented in C++ and was tested on the IBM PC with Pentium IV 3.4GHz and 2GB RAM. Fig. 8 depicts stages of detection of the warning RSs. Their characteristic feature is a yellow background and a red rim (the Polish RSs). However, the rim is usually very thin or totally missing due to wear and tear under harsh weather conditions. Thus, only the yellow areas are taken for segmentation. However, the presented technique can be easily applied to other world warning RSs which usually have white background and thick red border. In such a case we simply substitute the yellow with the red segmentation. The other stages are the same.

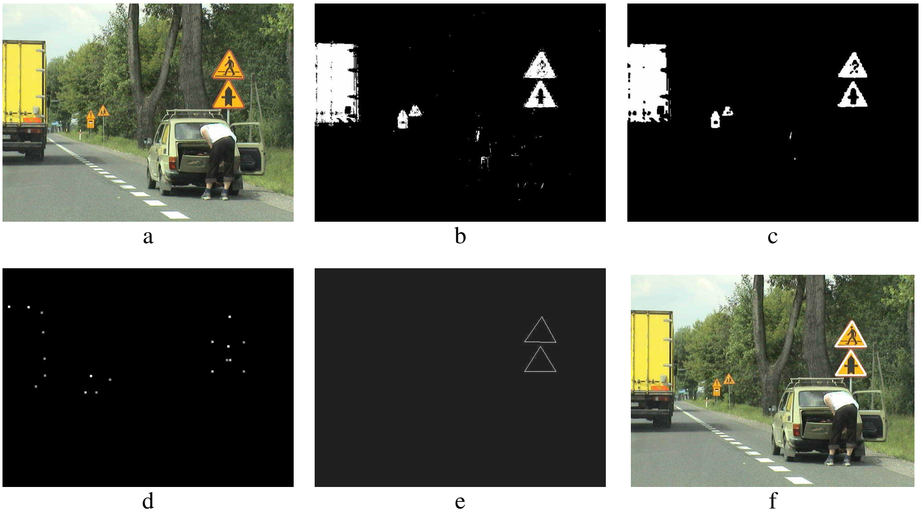


Fig. 8. Experimental results of detection of the warning signs group "A". The original scene (a), the colour segmented map (b), after erosion (c), salient points (d), detected figures (e,f).

Images in Fig. 8 follow the scheme in Fig. 1: (b) depicts the areas after fuzzy segmentation of the yellow colour (Fig. 2c,d); (c) shows (b) after the morphological erosion with the 5×5 square structural element; (d) visualizes the salient points – different lightness was used to denote different types of the characteristic points, such as top-corner, left-bottom, etc (see Fig. 5). Fig. 8ef show the detected triangles after the fuzzy figure detection and verification stages. Let us note that *all possible configurations* have been checked but only the ones that *fulfil* the triangle conditions R_{17}, were

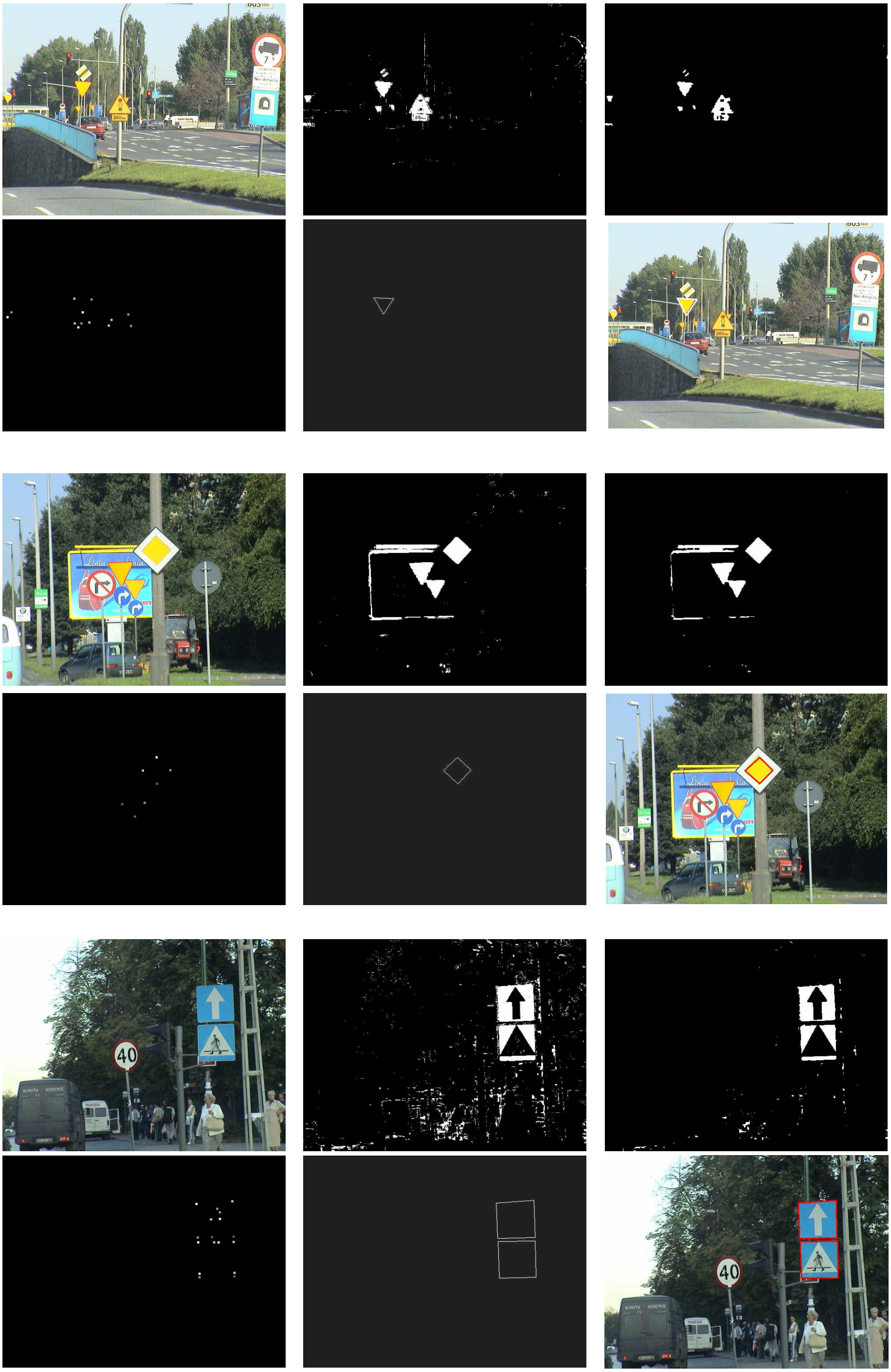


Fig. 9. Experimental results of detection of RSs with different shapes: A7, D1, D3 and D6

left. The detected figures are correct despite many false points due to the yellow lorry. Fig. 9 illustrates experimental results of detection of signs with different shapes; These are the signs: A7, D1, as well as D3 with D6 (from top to bottom, image order is the same as in Fig. 8). Let us note that for the A7 sign – the inverted triangle – as well as for the D1 – the diamond – detection of a sign is also its recognition. For all the others, we need to analyze their pictograms – for more details refer to [2][3]. The detection accuracy is also very high and is above 96% for all RS groups. Table 2 contains average execution times for different signs from the groups "A" and "D". System performs the incoming *320×240* video stream in real-time. The most time consuming stages are the morphological erosion and segmentation, respectively.

Table 2. Average execution times (ms) for detection of the road signs with different shapes

"A", Δ	"A", ∇	"D", □	"D", ◊
41	32	37	32

6 Conclusions

In this paper we present a system for fuzzy detection of the triangular and rectangular shaped road signs. The three groups of fuzzy rules have been proposed to perform colour segmentation, detection of the salient points, as well as shape verification. Although designed to work in Polish conditions the system can be easily adopted to other countries as well. Moreover, it is fast and very accurate, what was shown by experimental results. However, the method can fail if parts of a sign, which contain salient points, are not visible. Nevertheless this is not so severe, since we process a video stream from a moving vehicle. Thus, it is highly probable that the occlusions will affect only negligible amount of frames since RSs are usually placed in the most visible areas of the roads. However, the other parts which do not contain areas with salient point, can be occluded. This does not affect the method. False detections can also happen. However, they can be verified by two other mechanisms: The first is a verification in the next video frame (tracking). The second mechanism is the classification module – it simply rejects objects with unknown pictograms [3].

Acknowledgement

This work was supported from the Polish funds for the scientific research in 2007.

References

1. Cyganek, B.: Road Signs Recognition by the Scale-Space Template Matching in the Log-Polar Domain. Iberian Conf. on Pattern Recognition and Image Analysis, Spain (2007)
2. Cyganek, B.: Rotation Invariant Recognition of Road Signs with Ensemble of 1-NN Neural Classifiers, LNCS 4132 (2006) 558 – 567
3. Cyganek, B.: Recognition of Road Signs with Mixture of Neural Networks and Arbitration Modules. Proc. of ISNN, China, LNCS 3973, Springer (2006) 52 – 57

4. DaimlerChrysler, The Thinking Vehicle, http://www.daimlerchrysler.com (2002)
5. Driankov, D., Hellendoorn, H., Reinfrank, M.: An Introduction to Fuzzy Control. (1996)
6. Escalera, A., Armingol, J.A.: Visual Sign Information Extraction and Identification by Deformable Models. IEEE Tr. On Int. Transportation Systems, v. 5, no 2, (2004) 57-68
7. Fleyeh, H., Gilani, S.O., Dougherty, C.: Road Sign Detection And Recognition Using Fuzzy Artmap. IASTED Int. Conf. on Art. Intell. and Soft Computing (2006) 242-249
8. Gao, X.W., Podladchikova, L., Shaposhnikov, D., Hong, K., Shevtsova, N.: Recognition of traffic signs. Journal of Visual Communication & Image Representation (2005)
9. Gavrila, D.M.: Multi-feature Hierarchical Template Matching Using Distance Transforms. Proc. of the Int. Conf. on Pattern Recognition, Brisbane (1998) 439-444
10. Kecman, V.: Learning and Soft Computing. MIT Press (2001)
11. Paclik, P., Novovicova, J., Pudil, P., Somol, P.: Road sign classification using Laplace kernel classifier," Pattern Recognition Letters, vol. 21, (2000) 1165-1173
12. Piccioli, G., Micheli, E.D., Parodi, P., Campani, M.: Robust method for road sign detection and recognition. Image and Vision Computing, v.14 (1996) 209-223
13. Zheng, Y. J., Ritter, W., Janssen, R.: An adaptive system for traffic sign recognition, in Proc. IEEE Intelligent Vehicles Symp. (1994) 165–170

Nonlinear Neuro-fuzzy Network for Channel Equalization

Rahib Abiyev, Fakhreddin Mamedov, and Tayseer Al-shanableh

Near East University, Department of Computer Engineering, Lefkosa, Mersin-10, Turkey
{rahib,fahri,shanableh}@neu.edu.tr

Abstract. This paper presents a Nonlinear Neuro-Fuzzy Network (NNFN) for equalization of channel distortion. The NFNN is constructed by using fuzzy rules that incorporate nonlinear functions. The structure and learning algorithm of NNFN are presented and the development of adaptive equalizer has been performed. The equalizer is applied for equalization of channel distortion of time-invariant and time-varying channels. The performance of NNFN based equalizer has been compared with the performance of other nonlinear equalizers. The obtained simulation results of NNFN based system satisfies the effectiveness of the proposed system in equalization of channel distortion.

1 Introduction

Recently, fuzzy neural systems have been developed and applied in engineering fields to solve control, communication, prediction, signal processing and pattern recognition problems [1-7]. Development of fuzzy neural systems includes the generation of proper knowledge base that has IF-THEN form and then finding the optimal definition of the premise and consequent parts of these fuzzy rules through the training capability of neural networks. Neural fuzzy controller (NEFCON) [2], adaptive neuro-fuzzy inference system (ANFIS) [3], evolving fuzzy neural network [6] and TSK-type recurrent neuro-fuzzy neural network [7]are developed for modeling and control of nonlinear processes. In [4] a training procedure with variable system structure approach for fuzzy inference system is presented. In [5] using α - level procedure the training of fuzzy neural network is carried out and developed system is applied for control of technological processes.

The structures of most of neuro-fuzzy systems mainly implement the TSK-type or Mamdani-type fuzzy reasoning mechanisms. Adaptive neuro-fuzzy inference system (ANFIS) implements TSK-type fuzzy system [3]. The consequent parts of the TSK-type fuzzy system include linear functions. This fuzzy system can describe the considered problem by means of combination of linear functions. Sometimes these fuzzy systems need more rules, during modeling complex nonlinear processes in order to obtain the desired accuracy. Increasing the number of the rules leads to the increasing the number of neurons in the hidden layer of the network. To improve the computational power of neuro-fuzzy system, we use nonlinear functions in the consequent part of each rule. Based on these rules, the structure of the nonlinear neuro-fuzzy network (NNFN) has been proposed. Because of these nonlinear functions, NNFN network has

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 327–336, 2007.
springerlink.com

more computational power, and, it can describe nonlinear processes with desired accuracy. In this paper, the NNFN is used for equalization of nonlinear channel distortion.

Different approaches have been proposed for channel equalization. Classical approaches for adaptive equalizer design are based on knowledge of the parametric channel model [9]. Other type of adaptive equalizers is based on linear system theory, such as decision feedback equalizer that improves the performance of equalizer. Nowadays neural networks are widely used for channel equalization. One class of nonlinear adaptive equalizer is based on multiplayer perceptrons (MLP) and radial basis functions (RBF) [11,12]. MLP equalizers require long training and are sensitive to the initial choice of network parameters. The RBF equalizers are simple and require less time for training, but usually require a large number of centers, which increase the complexity of computation. The application of neural networks for adaptive equalization of nonlinear channel is given in [11]. Using 16 QAM (quadrature amplitude modulation) scheme the simulation of equalization of communication systems is carried out. In [13] neural decision-feedback equalizer is developed by using adaptive filter algorithm for equalization of nonlinear communication channels.

One of the effective ways for development of adaptive equalizers for nonlinear channels is the use of fuzzy technology. This type of adaptive equalizer can process numerical data and linguistic information in natural form [14]. Fuzzy equalizer that includes fuzzy IF-THEN rules was proposed for nonlinear channel equalization [15,16,17]. The incorporation of linguistic and numerical information improves the adaptation speed and its bit error rate (BER). Sometimes the construction of proper fuzzy rules for equalizers is difficult. One of the effective technologies for construction of equalizer's knowledge base is the use of neural networks. In this paper the NNFN structure and its learning algorithm are applied for development of equalizer. The NNFN network allows in short time train equalizer and gives better bit error rate (BER) results, at the cost of computational strength. The paper is organized as follows. In section 2 the architecture and learning algorithm of NNFN are given. In section 3 the simulation of NNFN system for channel equalization is presented. Finally, section 4 includes the conclusion of the paper.

2 Nonlinear Neuro-fuzzy Network

The kernel of a fuzzy inference system is fuzzy knowledge base. In a fuzzy knowledge base the information that consists of input-output data points of the system is interpreted into linguistic interpretable fuzzy rules. In the paper the fuzzy rules that have IF-THEN form and constructed by using nonlinear quadratic functions are used. They have the following form.

$$\text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots\text{and } x_m \text{ is } A_{jm} \text{ Then } y_j = \sum_{i=1}^{m} (w1_{ij} x_i^2 + w2_{ij} x_i) + b_j \qquad (1)$$

Here $x_1, x_2,\ldots,x_m$ are input variables, y_j $(j=1,..,n)$ are output variables which are nonlinear quadratic functions, A_{ji} is a membership function for i-th rule of the j-th input defined as a Gaussian membership function. $w1_{ij}$, $w2_{ij}$ and b_j $(i=1,..m, j=1,\ldots,n)$ are parameters of the network. The fuzzy model that is described by IF-THEN rules can be obtained by modifying parameters of the conclusion and premise parts of the rules. In this paper, a gradient method is used to train the parameters of rules in the neuro-fuzzy network structure. Using fuzzy rules in equation (1), the structure of the NNFN is proposed (Fig. 1). The NNFN includes seven layers. In the first layer the number of nodes is equal to the number of input signals. These nodes are used for distributing input signals. In the second layer each node corresponds to one linguistic term. For each input signal entering to the system the membership degree to which input value belongs to a fuzzy set is calculated. To describe linguistic terms the Gaussian function is used.

$$\mu 1_j(x_i) = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, \quad i=1..m, \quad j=1..J \tag{2}$$

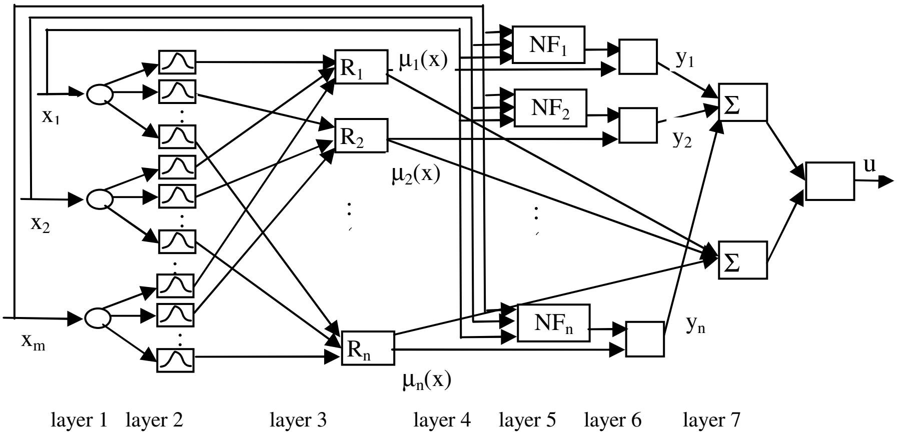


Fig. 1. Structure of NNFN

Here *m* is number of input signals, *J* is number of linguistic terms assigned for external input signals x_i. c_{ij} and σ_{ij} are centre and width of the Gaussian membership functions of the *j*-th term of *i*-th input variable, respectively. $\mu 1_j(x_i)$ is the membership function of *i*-th input variable for *j*-th term. *m* is number of external input signals.

In the third layer, the number of nodes corresponds to the number of the rules (R_1, R_2,…,R_n). Each node represents one fuzzy rule. To calculate the values of output signals, the AND (min) operation is used. In formula (3), Π is the min operation

$$\mu_l(x) = \prod_j \mu 1_j(x_i), \; l=1,..,n, \; j=1,..,J \tag{3}$$

The fourth layer is the consequent layer. It includes n Nonlinear Functions (NF) that are denoted by NF_1, NF_2,…,NF_n, The outputs of each nonlinear function in Fig.1 are calculated by using the following equation (1-3).

$$y_j = \sum_{i=1}^{m} (w1_{ij} x_i^2 + w2_{ij} x_i) + b_j, \quad j = 1,...,n \tag{4}$$

In the fifth layer, the output signals of third layer $\mu_l(x)$ are multiplied with the output signals of nonlinear functions. In the sixth and seventh layers, defuzzification is made to calculate the output of the entire network.

$$u = \sum_{l=1}^{n} \mu_l(x) y_l \Big/ \sum_{l=1}^{n} \mu_l(x) \tag{5}$$

Here y_l is the outputs of fourth layer that are nonlinear quadratic functions, u is the output of whole network.

After calculating the output signal of the NNFN, the training of the network starts. Training includes the adjustment of the parameter values of membership functions c_{ij} and σ_{ij} $(i=1,..,m,\ j=1,..,n)$ in the second layer (premise part) and parameter values of nonlinear quadratic functions $w1_{ij}$, $w2_{ij}$, b_j $(i=1,..,m,\ j=1,..,n)$ in the fourth layer (consequent part). At first step, on the output of network the value of error is calculated.

$$E = \frac{1}{2} \sum_{i=1}^{O} (u_i^d - u_i)^2 \tag{6}$$

Here O is number of output signals of network (in given case O=1), u_i^d and u_i are the desired and current output values of the network, respectively. The parameters $w1_{ij}$, $w2_{ij}$, b_j $(i=1,..,m,\ j=1,..,n)$ and c_{ij} and σ_{ij} $(i=1,..,m,\ j=1,..,n)$ are adjusted using the following formulas.

$$w1_{ij}(t+1) = w1_{ij}(t) + \gamma \frac{\partial E}{\partial w1_j} + \lambda (w1_{ij}(t) - w1_{ij}(t-1)) \tag{7}$$

$$w2_{ij}(t+1) = w2_{ij}(t) + \gamma \frac{\partial E}{\partial w2_{ij}} + \lambda (w2_{ij}(t) - w2_{ij}(t-1)) \tag{8}$$

$$b_j(t+1) = b_j(t) + \gamma \frac{\partial E}{\partial b_j} + \lambda (b_j(t) - b_j(t-1)) \tag{9}$$

$$c_{ij}(t+1) = c_{ij}(t) + \gamma \frac{\partial E}{\partial c_{ij}}; \qquad \sigma_{ij}(t+1) = \sigma_{ij}(t) + \gamma \frac{\partial E}{\partial \sigma_{ij}} \tag{10}$$

Here $\gamma$is the learning rate, λ is the momentum, m is number of input signals of the network (input neurons) and n is the number of rules (hidden neurons). The values of derivatives in (7-8) are determined by the following formulas.

$$\frac{\partial E}{\partial w1_{ij}} = (u(t) - u^d(t)) \cdot \frac{\mu_l}{\sum_{l=1}^{n} \mu_l} \cdot x_i^2; \tag{11}$$

$$\frac{\partial E}{\partial w2_{ij}} = (u(t) - u^d(t)) \cdot \frac{\mu_l}{\sum_{l=1}^{n} \mu_l} \cdot x_i$$

$$\frac{\partial E}{\partial b_j} = u(t) - u^d(t)) \cdot \frac{\mu_l}{\sum_{l=1}^{n} \mu_l}$$

The derivatives in (10) are determined by the following formulas.

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u} \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial c_{ij}} \quad , \quad \frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u} \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial \sigma_{ij}} \tag{12}$$

Here

$$\frac{\partial E}{\partial u} = u(t) - u^d(t), \quad \frac{\partial u}{\partial \mu_l} = \frac{y_l - u}{\sum_{l=1}^{L} \mu_l}, \quad \mathrm{i} = 1,..,\mathrm{m}, \mathrm{j} = 1,..,\mathrm{n}, \mathrm{l} = 1,..,\mathrm{n} \tag{13}$$

$$\frac{\partial \mu_l(x_j)}{\partial c_{ji}} = \begin{cases} \mu_l(x_j) \dfrac{2(x_j - c_{ji})}{\sigma_{ji}^2} \; \textit{if } j \text{ node} \\ \quad \text{is connected to rule node } l \\ 0, \quad \text{otherwise} \end{cases} \qquad \frac{\partial \mu_l(x_j)}{\partial \sigma_{ji}} = \begin{cases} \mu_l(x_j) \dfrac{2(x_j - c_{ji})^2}{\sigma_{ji}^3} \; \textit{if } j \text{ node} \\ \quad \text{is connected to rule node } l \\ 0, \quad \text{otherwise} \end{cases} \tag{14}$$

Using equations (11) and (14) in (7)-(10) the learning of the parameters of the NNFN is carried out.

3 Simulation

The architecture of the NNFN based equalization system is shown in Fig. 2. The random binary input signals s(k) are transmitted through the communication channel. Channel medium includes the effects of the transmitter filter, transmission medium, receiver filter and other components. Input signals can be distorted by noise and intersymbol interference. Intersymbol interference is mainly responsible for linear distortion. Nonlinear distortions are introduced through converters, propagation environment. Channel output signals are filtrated and entered to the equalizer for equalizing the distortion.

During simulation the transmitted signals *s(k)* are known input samples with an equal probability of being –1 and 1. These signals are corrupted by additive noise *n(k)*. These corrupted signals are inputs for the equalizer. In channel equalization, the problem is the classification of incoming input signal of equalizer onto feature space which is divided into two decision regions. A correct decision of equalizer occurs if $\overline{s}(k) = s(k)$. Here *s(k)* is transmitted signal, i.e. channel input, $\overline{s}(k)$ is the output of equalizer. Based on the values of the transmitted signal *s(k)* (i.e. ±1) the channel state can be partitioned into two classes R^+ and R^-.Here $R^+=\{x(k) \mid s(k)=1\}$ and $R^-=\{x(k) \mid s(k)=-1\}$. *x(k)* is the channel output signal.

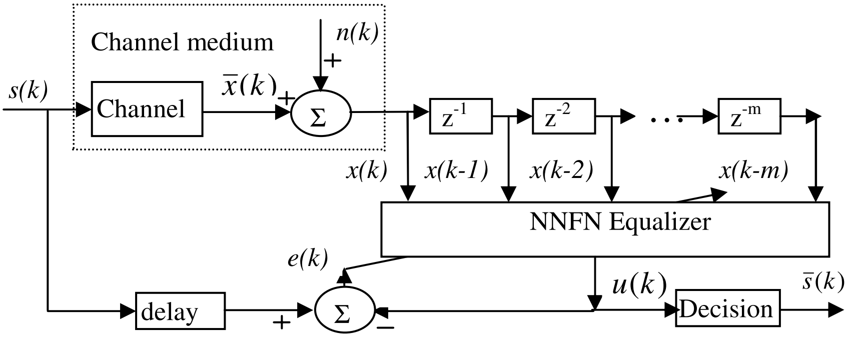


Fig. 2. The architecture of the NNFN based equalization system

In this paper the NNFN structure and its training algorithm are used to design equalizer. Simulations have been carried out for the equalization of linear and nonlinear channels.

In the first simulation, we use the following nonminimum-phase channel model.

$$x(k) = a_1(k)s(k) + a_2(k)s(k-1) + a_3(k)s(k-2) + n(k) \tag{15}$$

where $a_1(k) = 0.3482$, $a_2(k) = 0.8704$ and $a_3(k) = 0.3482$. *n(k)* is additive noise. This type of channel is encountered in real communication systems. During equalizer design, the sequence of transmitted signals is given to the channel input. 200 symbols are used for training and 10^3 signals for testing. They are assumed to be an independent sequence taking values from {-1,1} with equal probability. The additive Gaussian noise *n(k)* is added to the transmitted signal. In the output of the equalization system, the deviation of original transmitted signal from the current equalizer output is determined. This error *e(k)* is used to adjust network parameters. Training is continued until the value of the error for all training sequence of signals is acceptably low.

During simulation, the input signals for the equalizer are outputs of channel *x(k), x(k-1), x(k-2), x(k-3)*. Using NNFN, ANFIS [19], and feedforward neural networks the computer simulation of equalization system has been performed. During simulation, we used 27 rules (hidden neurons) in the NNFN, 27 hidden neurons in the feedforward neural network and 36 rules (hidden neurons) in the ANFIS based equalizer. The learning of equalizers has been carried out for 3000 samples. After simulation the performance characteristics (bit error rate versus signal-noise ratio) for all equalizers have been determined. Bit Error Rate (BER) versus Signal-Noise Ratio (SNR) characteristics have been obtained for different noise levels. Fig. 3 show the

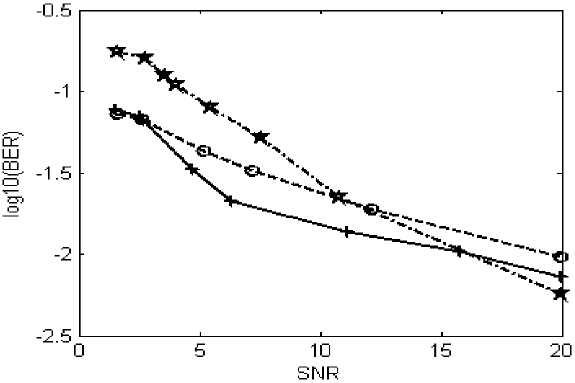


Fig. 3. Performance of NNFN (solid line with '+'), ANFIS (dashed line with 'o') and feedforward neural network (dash-doted line with '*') based equalizers

performance of equalizers based on NNFN, ANFIS and feedforward neural networks. In Fig. 3 solid line is the performance of the NNFN based equalizer, dashed line is the performance of the equalizer based on ANFIS and dash-dotted line is the performance of feedforward neural network based equalizer. As shown in figure, at the area of low SNR (high level of noises) the performance of NNFN based equalizer is better than other ones.

In the second simulation, the following nonlinear channel model was used

$$x(k)=a_1(k)s(k)+a_2(k)s(k\text{-}1)\ \text{-}0.9\cdot(a_1(k)s(k)+a_2(k)s(k\text{-}1))^3+n(k) \qquad (16)$$

where $a_1(k)=1$ and $a_2(k)=0.5$. We consider the case when the channel is time varying, that is $a_1(k)$ and $a_2(k)$ coefficients are time-varying coefficients. These are generated by using second-order Markov model in which white Gaussian noise source drives a second-order Butterworth low-pass filter [4,22]. In simulation a second order Butterworth filter with cutoff frequency 0.1 is used. The colored Gaussian sequences which were used as time varying coefficients a_i are generated with a standard deviation of 0.1. The curves representing the time variation of the channel coefficients are depicted in Fig. 4. The first 200 symbols are used for training. 10^3 signals are used for testing. The simulations are performed using NNFN, ANFIS and feedforward neural networks. 36 neurons are used in the hidden layer of each network. Fig. 5 illustrates the BER performance of equalizers for channel (16), averaged over 10 independent trials. As shown in figure performance of NNFN based equalizer is better than other ones.

In Fig. 6, error plot of learning result of NNFN equalizer is given. The channel states are plotted in Fig. 7. Here Fig.7(a) demonstrates noise free channel states, 7(b) is channel states with additive noise, and Fig. 7(c) is channel states after equalization

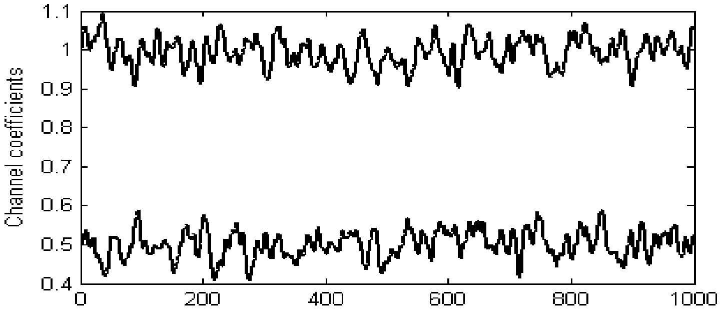


Fig. 4. Time-varying coefficients of channel

of distortion. Here 7(c) describes the simulation result after 3000 learning iterations. The obtained result satisfies the efficiency of application of NNFN technology in channel equalization.

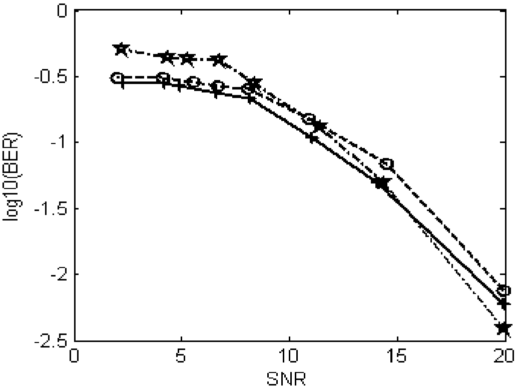


Fig. 5. Performance of the NNFN (solid line with '+'), ANFIS (dashed line with 'o') and feed-forward neural network (dash-doted line with '*') based equalizers

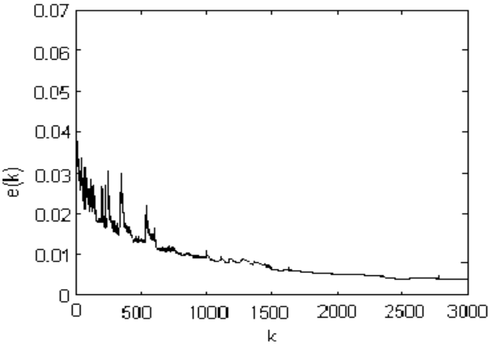
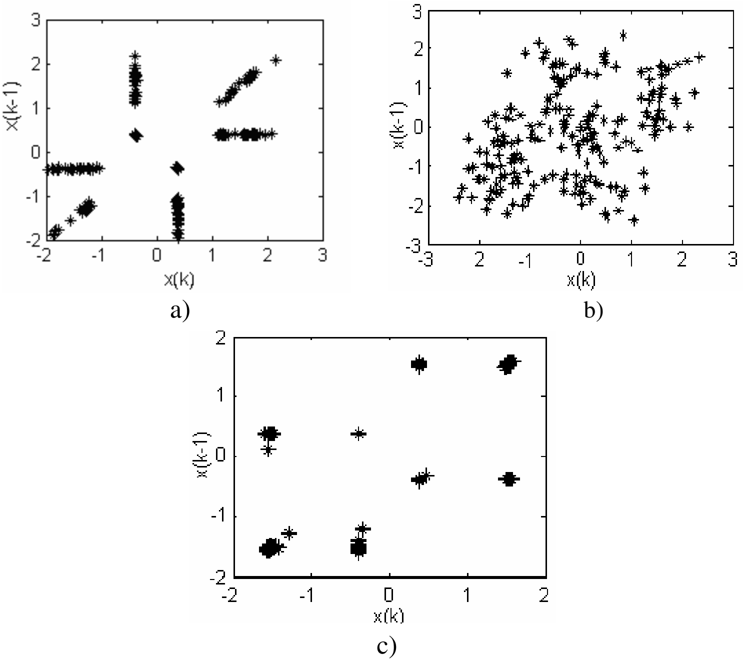


Fig. 6. Error plot



a) b)

c)

Fig. 7. Channel states: a) noise free, b) with noise, c) after equalization

4 Conclusion

The structure and learning of NNFN are represented. The NNFN and its learning algorithm are applied for equalization of the linear and nonlinear time-varying channels in presence of additive distortion. Simulation result of NNFN based equalizer is compared with the simulation results of equalizers based on feedforward neural network and ANFIS. It was found that NNFN based equalizer has better BER performance than other equalizers in the noise channels. Comparative simulation results satisfy the efficiency of application of the NNFN in adaptive channel equalization.

References

[1] Yager R.R., Zadeh L.A.(Eds).”Fuzzy sets, neural networks and soft computing”, Van Nostrand Reinhold, New York (1994).

[2] Nauck, Detlef, and Kruse, Rudolf, “Designing neuro-fuzzy systems through backpropagation”, In Witold Pedryz, editor, Fuzzy Modelling: Paradigms and Practice, pages 203-228, Kluwer Academic Publisher, Boston (1996).

[3] J.S.R. Jang, C.-T.Sun, E.Mizutani. Neuro-fuzzy and Soft Computing: a Computational Approach to Learning and Machine Intelligence, Prentice-Hall, NJ, (1997).

[4] M.Onder Efe, and Okyay Kaynak, "On stabilization of Gradient-Based Training Strategies for Computationally Intelligent Systems". IEEE Transactions on Fuzzy Systems, Vol.8, No.5 (2000) 564-575.

[5] R.H.Abiyev, R.A.Aliev, and R.R.Aliev, "The synthesis of fuzzy control system with neural networks based tuner". News of Academy of Sciences, Tech. Cybernetics N2: (1994) 192-197.

[6] N. Kasabov. "Evolving fuzzy neural networks for online, adaptive, knowledge-based learning". IEEE Transaction, Systems, Man, Cybernetica. B, vol.31, pp.902-918 (2001)

[7] Chia-Feng Juang, "A TSK –type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithm", IEEE Trans. Fuzzy Systems, vol.10 (2002) 155-170.

[8] D.D.Falconer, "Adaptive Equalization of Channel Nonlinearites in QAM Data Transmission Systems", Bell System Technical Journal, vol.27, no.7 (1978)

[9] C.F.N.Cowan, S.Semnani. Time-variant equalization using novel nonlinear adaptive structure. Int.J.Adaptive Contr. Signal Processing, vol.12, no.2, (1998) 195-206.

[10] Chen, S., Gibson, G.J., Cowan, C.F.N., and Grant, P.M. "Adaptive equalization of finite non-linear channels using multiplayer perceptrons". Signal Process, 20, (2)(1990) 107-119.

[11] Chen, S., Gibson, G.J., Cowan, C.F.N., and Grant, P.M. Reconstruction of binary signals using an adaptive radial-basis function equalizer", Signal Processing, 22,(1), (1991) 77-93

[12] Deniz Erdogmus, Deniz Rende, Jose C. Principe, Tan F. Wong. Nonlinear channel equalization using multiplayer perceptrons with information-theoretic criterion. In Proc. Of 2001 IEEE Signal Processing Society Workshop (2001) 443-451.

[13] Zhe Chen, Antonio C. de C.Lima. a new neural equalizer for Decision-Feedback Equalization. IEEE Workshop on Machine Learning for signal processing, 2004.

[14] Li-Xin Wang, Jerry M. Mendel. Fuzzy adaptive filters, with application to Nonlinear Channel Equalization. IEEE Transaction on Fuzzy Systems, vol.1, No.3 (1993)

[15] P.Sarwal and M.D.Srinath, A fuzzy logic system for channel equalization. IEEE Trans. Fuzzy System, vol.3, (1995) 246-249.

[16] S.K.Patra, B Mulgrew. Fuzzy implementation of Bayesian equalizer in the presence of intersymbol and co-channel interference. Proc. Inst. Elect. Eng. Commun., vol. 145 (1998) 323-330.

[17] S.Siu, Chia-Lu, Chien-Min Lee. TSK-based decision feedback equalization using an evolutionary algorithm applied to QAM Communication Systems. IEEE Transactions on Circuits and Systems, vol.52, No.9, 2005.

[18] Rahib Abiyev, Fakhreddin Mamedov , Tayseer Al-shanableh. Neuro-fuzzy system for Channel noise Equalization. International Conference on Artificial Intelligence.IC-AI'04, Las Vegas, Nevada, USA, June 21-24 (2004)

On the Possibility of Reliably Constructing a Decision Support System for the Cytodiagnosis of Breast Cancer

Nicandro Cruz-Ramírez[1], Héctor-Gabriel Acosta-Mesa[1],
Humberto Carrillo-Calvet[2], and Rocío-Erandi Barrientos-Martínez[1]

[1] Facultad de Física e Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho # 5, Col. Centro, C.P. 91000, Xalapa, Veracruz, México
{ncruz, heacosta}@uv.mx, erandi_bm@yahoo.com.mx
[2] Facultad de Ciencias, Universidad Nacional Autónoma de México, Circuito Exterior Ciudad Universitaria, México, D.F.
carr@servidor.unam.mx

Abstract. We evaluate the performance of three Bayesian network classifiers as decision support system in the cytodiagnosis of breast cancer. In order to test their performance thoroughly, we use two real-world databases containing 692 cases collected by a single observer and 322 cases collected by multiple observers respectively. Surprisingly enough, these classifiers generalize well only in the former dataset. In the case of the latter one, the results given by such procedures have a considerable reduction in the sensitivity and PV- tests. These results suggest that different observers see different things: a problem known as interobserver variability. Thus, it is necessary to carry out more tests for identifying the cause of this subjectivity.

1 Introduction

One of the most common types of cancer that affects women in North America, Europe and the Antipodes, is breast cancer [3]. The usual practice in the United Kingdom to accurately diagnose such a disease is to use three different methods: the surgeon's clinical opinion, a mammography and the cytological studies. That is why this diagnosis process is known as the triple approach [3]. Because mammographic studies are not decisive in the diagnosis of breast cancer [3, 9], an alternative confirmatory method is needed to support or reject these findings. The most common confirmatory method used in the United Kingdom for this purpose is that of fine needle aspiration of breast lesion (FNAB) [3-5]. Such a technique involves a process where a syringe sucks cells from breast lesions using a fine bore needle similar to those used for blood samples. Once this is done, these cells are transferred to a transport solution and sent to the pathology laboratory for a microscopic study carried out by a trained cytopathologist [3]. In this paper, we focus on how to build a decision support system, based on Bayesian networks, from a cytopathological database retrospectively collected by a single observer and the validation of such a system using other database prospectively collected by multiple observers.

The time it normally takes to a medical doctor to become an independent practising cytopathologist is about 5 years as a minimum. This fact can give an indication of the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 337–345, 2007.
springerlink.com

very complex learning process through which medical doctors have to pass. It is mainly for this reason that machine learning methods for decision-making may have two potential applications: to accelerate the training process of learning by providing guidance to the trainee on what features are the most important ones to look at; and to compare the final results to those of the trainee or even that of the expert so that the decision (whether the sample taken indicates a benign or a malignant output) can be made on more robust criteria. In this paper, we explore the first issue. We use 692 consecutive adequate specimens collected by a single pathologist to train (using 462 of those 692 cases) and test (using 230 of those 692 cases) 3 Bayesian network classifiers and 322 consecutive adequate specimens collected by 19 pathologists to also test that same classifiers resultant from the previous training phase. The results show the presence of a difficulty known as the interobserver variability problem [10, 13, 15], which prevents the classifiers from generalizing well enough. The interobserver variability problem refers to the situation where there is no reproducibility of the results given by different experts on the same set of cases. That is to say, in the specific case of the cytodiagnosis of breast cancer, pathologists do not agree on what cytological features are the most relevant for correctly diagnosing whether a patient has breast cancer or not given a specific dataset.

Although the inconsistencies in the two datasets prevent the decision support system presented here from performing robustly, it can be argued that, with the aid of this kind of tools (Bayesian networks), the main underlying principles, conditions, mechanisms and causes that lead to this problem could be gradually discovered. The remainder of this paper is organized as follows. In section 2 we present the material and methods used for this research. In section 3 we present the performance of three algorithms that build Bayesian network structures from a database collected by a single observer and a database collected by multiple observers. In section 4 we discuss the results given by these procedures and finally, in section 5, we present some conclusions and propose some sensible directions for future research.

2 Materials and Methods

2.1 The Databases

We used two different databases to carry out our studies: a dataset retrospectively collected by a single observer and a dataset prospectively collected by 19 different observers ranging from 5 to 20 years of experience on FNABs. The former contains 692 consecutive adequate specimens of FNAB received at the Department of Pathology, Royal Hallamshire Hospital in Sheffield during 1992-1993 [3]. The latter consists of 322 consecutive adequate FNAB specimens received at the same department during 1996-1997 [5]. In such a database the 19 pathologists were given one hour of training with images of the observable characteristics that serve to clarify definitions and a document containing the details of the features with representative colour images of each of them. 11 independent variables and 1 dependent variable form part of both datasets. The independent variables are: age, ‘cellular dyshesion’, ‘intracytoplasmic lumina', 'three-dimensionality of epithelial cells clusters', 'bipolar naked nuclei', 'foamy macrophages', nucleoli, 'nuclear pleomorphism', 'nuclear size', 'necrotic epithelial cells' and 'apocrine change'. All these variables, except age, were dichotomized taking the values of "true" or "false" indicating respectively the presence or absence of a diagnostic feature. Variable age was actually sorted into three different categories: 1 (up to 50 years old), 2 (51 to 70 years old) and 3 (above 70 years old). The dependent variable "outcome" can also take on two different values: benign or malignant (for more details on the meaning of these variables, the reader is referred to [3]). In the case of a malignant outcome, the results were confirmed by an open biopsy in both databases (where available). In the case of a benign outcome, the results were confirmed by clinical details, mammographic findings (where available) and by absence of further malignant specimens.

2.2 Bayesian Network Classification

In a classification task, given a set of unlabelled cases on the one hand, and a set of labels on the other, the problem to solve is to find a function that suitably maps each unlabelled instance to its corresponding label (class). As can be inferred, the central research interest in this specific area is the design of automatic classifiers that can estimate this function from data. The efforts of such a research have resulted in different classification methods: regression, decision trees, Bayesian networks and neural networks, among others [14]. For the tests reported here, we used 3 algorithms that build Bayesian network structures from data: PC, CBL2 and MP-Bayes [1, 6, 11]. For the lack of space, we do not provide the details of any of these 3 procedures. Instead, we only briefly describe them and refer the reader to their representative sources.

1. PC is an algorithm for constructing Bayesian network (BN) structures from data [11, 12]. It uses the G^2 statistics to carry out conditional independence tests for deciding the deletion of an arc between two nodes (random variables). It is important to mention that procedure PC is stepwise backward algorithm: it assumes that every pair of variables is connected and then deletes arcs if the test indicates so. For more details on PC, the reader is referred to [11, 12].
2. CBL2 is a constraint-based algorithm to build BN structures from data [2]. CBL2 uses mutual information and conditional mutual information tests to decide when to connect/disconnect a pair of nodes. It is a stepwise forward procedure since its initial state is an empty graph: every pair of nodes is disconnected.
3. MP-Bayes is a constraint-based algorithm to build BN structures from data which uses mutual information and conditional mutual information measures, combined with the statistics T, to add/delete an arc between a pair of nodes [6]. MP-Bayes is a stepwise forward procedure: it assumes, as the initial state, an empty graph. First, it starts adding arcs if the independence tests do not hold. Then, the procedure starts removing arcs if the conditional independence tests do indeed hold. It has been explicitly designed for behaving parsimoniously: it tries to build Bayesian network structures with the least number of arcs (see table 1).

3 Experimental Methodology and Results

The single observer dataset was randomly divided into two different sets: a training set consisting of 462 cases and a test set containing the remaining 230 cases [8]. Table 1 shows the structures resultant of running procedures PC, CBL2 and MP-Bayes on the 462 cases. Then, these structures were used to check the classification performance on the remaining 230 cases of this dataset and on the whole multiple observer dataset (322 cases). These classification results are shown in tables 2 and 3 respectively. Also, in order to visually measure the degree of agreement among experts' diagnoses, we used the 322-case dataset for constructing the correspondent Bayesian networks. Table 4 shows such networks.

Table 1. Bayesian network structures resultant from running procedures PC, CBL2 and MP-Bayes on the 462-case dataset

PC	CBL2	MP-Bayes
Bipolar → Foamy	Age → Bipolar	Bipolar → Foamy
Age → Nucleoli	Age → Outcome	Bipolar → NuclearP
Age → Outcome	ThreeD → Outcome	Age → Nucleoli
ThreeD → Outcome	Cellular → Age	Age → Outcome
Cellular → Outcome	Cellular → Outcome	Nucleoli → NuclearS
Nucleoli → ThreeD	Nucleoli → Necro	Nucleoli → Outcome
Nucleoli → Outcome	Nucleoli → Outcome	Necro → Foamy
Nucleoli → NuclearS	Necro → Foamy	NuclarP → NuclearS
Necro → Foamy	Necro → Outcome	NuclearS → Outcome
Necro → Nucleoli	Apocri → Nucleoli	Intracyto → Outcome
Apocri → Nucleoli	NuclarS → NuclearP	
Apocri → Outcome	NuclearS → Nucleoli	
NuclarS → NuclearP	NuclearS → Outcome	
NuclarS → Outcome	Intracyto → Outcome	
Intracyto → Outcome		

Table 2. Results of accuracy, sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by PC, CBL2 and MP-Bayes on the 230-case test set. 95% confidence intervals are shown in parentheses.

Tests	PC	CBL2	MP-Bayes
Accuracy	93% ± 1.68%	92% ± 1.78%	92% ± 1.78
Sensitivity	87% (79-94)	88% (81-95)	88% (81-95)
Specificity	97% (94-100)	95% (91-98)	94% (90-98)
PV+	94% (88-99)	90% (84-97)	89% (82-96)
PV-	93% (89-97)	93% (89-97)	93% (89-97)

Table 3. Results of accuracy, sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by PC, CBL2 and MP-Bayes on the 322-case test set. 95% confidence intervals are shown in parentheses.

Tests	PC	CBL2	MP-Bayes
Accuracy	80% ± 2.22%	77% ± 2.34%	85% ± 1.98%
Sensitivity	56% (48-64)	50% (42-58)	72% (64-79)
Specificity	96% (94-99)	96% (94-99)	94% (91-98)
PV+	91% (85-98)	91% (84-97)	90% (84-95)
PV-	75% (70-81)	73% (67-79)	82% (77-87)

Table 4. Bayesian network structures resultant from running procedures PC, CBL2 and MP-Bayes on the 322-case dataset

PC	CBL2	MP-Bayes
Bipolar → Foamy	Bipolar → Age	Bipolar → Foamy
Age → Bipolar	Bipolar → Outcome	Age → Bipolar
Age → Outcome	Age → Outcome	Age → NuclearP
ThreeD → Outcome	ThreeD → NuclearP	Age → Outcome
Cellular → Outcome	ThreeD → Outcome	Cellular → Outcome
Cellular → NuclearS	Foamy → Bipolar	ThreeD → Outcome
NuclearP → NuclearS	Foamy → Nucleoli	Foamy → Apocri
Nucleoli → NuclearP	Foamy → NuclearP	NuclearP → NuclearS
	Foamy → NuclearS	NuclearS → Foamy
	Foamy → Apocri	
	NuclearP → Nucleoli	
	NuclearP → Intracyto	
	NuclearP → NuclearS	
	NuclearP → Outcome	
	Cellular → Outcome	
	NuclearS → Cellular	
	NuclearS → Outcome	
	Intracyto → Outcome	

4 Discussion

According to the medical literature, all 10 defined cytological observations mentioned in section 2.1 are considered relevant for the cytodiagnosis of FNAB [3]. Furthermore, variable age is also considered relevant for such a purpose as it provides useful information for making the final diagnosis [3]. As can be seen from the results shown in table 2, the overall performance by all classifiers seems to be reasonably acceptable. That is to say, the trade-off between sensitivity and specificity appears to be adequate. However, to press the limits of the performance of these three procedures, we decided to test them on the 322-case dataset (collected by multiple observers), in order to evaluate their behavior regarding the previously mentioned interobserver variability problem. Table 3 shows a considerable reduction in the performance of such classifiers. Looking into this table more closely, we can see that MP-Bayes has the best performance in accuracy, sensitivity and PV- while PC and CBL2 have the best performance in specificity and PV+. However, the tests where MP-Bayes behaves in the best way are better than those where PC and CBL2 show the best behavior. It seems that the conservative nature of MP-Bayes, regarding the addition of arcs (underfitting), makes it less vulnerable to the important interobserver variability problem. That is to say, since the number of arcs going to variable "outcome" given by PC and CBL2 (7 arcs; see table 1) is bigger than that produced by MP-Bayes (4 arcs), meaning that the former procedures match the training data (462 cases) better than the latter, they have, in general, a poorer performance (overfitting) on the test data (322 cases), as shown in table 3. Moreover, comparing the results of tables 1 and 4 for each procedure, one can easily see that the difference within the same procedure using the 230 and 322 case test sets is far bigger in PC and CBL2 than in MP-Bayes. In sum, the results given by MP-Bayes (table 3) seem to support Occam's razor: simpler models are, in general, more useful and adequate than complex models [7].

The previous experiments using the Bayesian network approach seem also to unveil interesting characteristics in the experts' behavior that avoid the reproducibility of results among different observers. To support this hypothesis, we decided to train the three procedures presented here with the 322-case database to measure the individual differences among the cytopathologists in the codification of the variables. Table 4 helps us visualize these differences. When classifying the variables, a single observer cannot be blind to the values they have assigned to earlier variables. When a variable has an ambiguous value, the expert may then be biased by their knowledge about an earlier decision. The observed values are thus not strictly independent. In fact, the expert's own implicit knowledge about the relationships between the variables may be affecting the values recorded in the dataset. The results of the Bayesian model will thus reflect the expert's knowledge rather than (as well as) the real relationships. This indicates a subjective process where every pathologist takes into account different variables as being relevant to diagnose breast cancer. It is now easy to see why the solutions of the three procedures do not generalize well when using the database collected by a single observer (462 cases) for training purposes and the database collected by multiple observers (322 cases) for test purposes. For the case of MP-Bayes, if we compare tables 1 and 4, we can notice that it is only two variables (age and 'nuclear size') which are common for the Bayesian networks of both figures. According to these results and the results of tables 2 and 3, it seems that both the single observer and the multiple observers agree that variables age and nuclear size are the most informative variables to determine specificity and PV+, since it is the values of these tests which remain practically constant within the two datasets. Recall that cytopathologists are specially trained to keep the values of these two tests at maximum (100%). Thus, these results seem to support such specialized behavior. Furthermore, the values for sensitivity and PV- drop considerably from one database (462 cases) to another (322 cases). These results also seem to support this mentioned overspecialization.

For the case of PC, if we compare tables 1 and 4, we can notice that it is now four variables (age, 'three dimensionality', 'cellular dyshesion' and 'nuclear size') which are common for the Bayesian networks of both tables. According to these results and the results of tables of tables 2 and 3, it seems less clear (with respect to the results of MP-Bayes) in what variables the single observer and the multiple observers agree this time as being the most informative variables to determine specificity and PV+, since the specificity remains almost constant within the two datasets but PV+ does not. Because of the overfitting produced by PC (in the sense of the number of arcs), this algorithm seems to be more vulnerable to the interobserver variability since the values for sensitivity and PV- drop much more dramatically for this procedure than for MP-Bayes from one database (462 cases) to another (322 cases): from 87% to 56% in sensitivity and from 93% to 75% in PV- for PC; from 88% to 72% in sensitivity and from 93% to 82% in PV- for MP-Bayes.

For the case of CBL2, if we compare tables 1 and 4, we can notice that it is five variables (age, 'three dimensionality', 'cellular dyshesion', 'intracytoplasmic lumina' and 'nuclear size') which are common for the Bayesian networks of both tables. In contrast with PC, and according to these results and the results of tables 2 and 3, it seems that the additional variable found by CBL2 to explain the outcome ('intracytoplasmic lumina') is the key to produce almost identical values of specificity and PV+ within the two datasets. However, with the selection of these five variables, CBL2 overfits the data as well. This overfitting can be perceived in the values of sensitivity and PV-, which drop even more dramatically than those given by PC from one database (462 cases) to another (322 cases): from 88% to 50% in sensitivity and from 93% to 73% in PV- for CBL2; from 87% to 56% in sensitivity and from 93% to 75% in PV- for PC. Hence, this algorithm seems to be also more vulnerable to the interobserver variability than MP-Bayes. The results of tables 1 and 4 seem to reveal interesting characteristics in the experts' behavior about their individual differences in the codification of the variables that avoid the reproducibility of results: only age is considered by all of them as being indicative of malignancy.

In order to make their final diagnoses, cytopathologists follow a process which is not very well known to date and can only be partially explained in terms of pattern recognition with occasional use of heuristic logic [3]. It is important to notice that all the features coded in the breast cancer datasets used by the 3 procedures were coded by the expert cytopathologists who carried out most of the processing that is probably required to solve the diagnostic problem. So, if this is true, then there is little work left to any classifier that uses this dataset. In other words, the information provided in such databases is subjective rather than objective.

5 Conclusions and Future Work

We have presented a study to check the possibility of reliably constructing decision support tools, based on Bayesian networks, for the cytodiagnosis of breast cancer. The results show that it is possible to accurately build automatic classifiers from data if these data come from a single observer. However, the results suggest not constructing such classifiers from data if these data come from different observers. The Bayesian network framework allows us to visually identify the so-called interobserver variability problem: the interpretation of a sample may vary from a pathologist to another one; a situation that is reflected in different Bayesian network structures. Hence, it is necessary to carry out more research in the direction of finding the potential causes that prevent the classifiers from performing accurately. It is probable that the pathologists are taking into account more information than that portrayed by the cytological images. Thus, we will be trying to incorporate more information so that we can confirm this hypothesis.

Acknowledgements

We are very grateful to Dr. Simon S. Cross, Clinical Senior Lecturer at the Academic Unit of Pathology of the University of Sheffield, United Kingdom, who kindly provided the database used in this study. The first author thanks the CONACyT (Consejo Nacional de Ciencia y Tecnología) grant number 70356 and PROMEP (Programa de Mejoramiento del Profesorado), project number PROMEP/103.5/06/0585 for the economic support given for this research. We are also very grateful to the Macroproyecto Tecnologías para la Universidad de la Información y la Computación de la Universidad Nacional Autónoma de México for the economic support.

References

1. Cheng J, Bell, D.A., Liu, W. Learning Belief Networks from Data: An Information Theory Based Approach. in Sixth ACM International Conference on Information and Knowledge Management. 1997: ACM.
2. Cheng J and Greiner R. Learning Bayesian Belief Network Classifiers: Algorithms and Systems. in Proceedings of the Canadian Conference on Artificial Intelligence (CSCSI01). 2001. Ottawa, Canada.
3. Cross SS, Downs J, Drezet P, Ma Z and Harrison RF, Which Decision Support Technologies Are Appropriate for the Cytodiagnosis of Breast Cancer?, in Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis, A. Jain, A. Jain, S. Jain and L. Jain, Editors. 2000, World Scientific. p. 265-295.
4. Cross SS, Dube AK, Johnson JS, McCulloch TA, Quincey C, Harrison RF and Ma Z, Evaluation of a statistically derived decision tree for the cytodiagnosis of fine needle aspirates of the breast (FNAB). Cytopathology, 1998. **8**: p. 178-187.
5. Cross SS, Stephenson TJ, Mohammed T and Harrison RF, Validation of a decision support system for the cytodiagnosis of fine needle aspirates of the breast using a prospectively collected dataset from multiple observers in a working clinical environment. Cytopathology, 2000(11): p. 503-512.
6. Cruz-Ramirez Nicandro N-FL, Acosta-Mesa Hector Gabriel, Barrientos-Martinez Erandi, Rojas-Marcial Juan Efrain, A Parsimonious Constraint-based Algorithm to Induce Bayesian Network Structures from Data, in IEEE Proceedings of the Mexican International Conference on Computer Science ENC 2005, IEEE, Editor. 2005, IEEE: Puebla. p. 306-313.
7. Grunwald P, Model Selection Based on Minimum Description Length. Journal of Mathematical Psychology, 2000. **44**: p. 133-152.
8. Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. in 14th International Joint Conference on Artificial Intelligence IJCAI'95. 1995a. Montreal, Canada: Morgan Kaufmann.
9. Roberts LM, MammoNet: a Bayesian Network Diagnosing Breast Cancer, in Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis, A. Jain, A. Jain, S. Jain and L. Jain, Editors. 2000, World Scientific. p. 101-148.

10. Sidawy MK, Stoler MH, Frable WJ, Frost AR, Masood S, Miller TR, Silverberg SG, Sneige N and Wang HH, Interobserver Variability in the Classification of Proliferative Breast Lesions by Fine-Needle Aspiration: Results of the Papanicolau Society of Cytopathology Study. Diagnostic Cytopathology, 1998. **18**(2): p. 150-165.
11. Spirtes P, Glymour C and Scheines R, Causation, Prediction and Search. First ed. Lecture Notes in Statistics, ed. J. Berger, S. Fienberg, J. Gani, K. Krickeberg, I. Olkin and B. Singer. Vol. 81. 1993: Springer-Verlag. 526.
12. Spirtes P, Scheines R, Meek C and Glymour C, Tetrad II: Tools for Causal Modeling. 1994, Lawrence Erlbaum Associates, Inc.: Hillsdale, New Jersey, United States of America.
13. Verkooijen HM, Peterse JL, Schipper MEI, Buskens E, Hendriks JHCL, Pijnappel RM, Peeters PHM, Rinkes IHMB, Mali WPTM and Holland R, Interobserver variability between general and expert pathologists during the histopathological assessment of large-core needle and open biopsies of non-palpable breast lesions. Europen Journal of Cancer, 2003. **39**: p. 2187-2191.
14. Witten IH and Frank E, Data Mining: Practical machine learning tools and techniques. Second ed. 2005: Morgan Kaufmann, San Francisco, 2005.
15. Woynarowski M, Cielecka-Kuszyk J, Kaluzynski A, Omulecka A, Sobaniec-Lotowska M, Stolarczyk J and Szczepanski W, Inter-observer variability in histopathological assessment of liver biopsies taken un a pediatric open label therapeutic program for chronic HBV infection treatment. World Journal of Gastroenterology, 2006. **12**(11): p. 1713-1717.

Spatial Heart Simulation and Analysis Using Unified Neural Network

S.M. Szilágyi[1], L. Szilágyi[1,2], and Z. Benyó[2]

[1] Sapientia – Hungarian Science University of Transylvania, Faculty of Technical and Human Sciences, Târgu Mureş, Romania
szs@ms.sapientia.ro
[2] Budapest University of Technology and Economics, Dept. of Control Engineering and Information Technology, Budapest, Hungary

Abstract. This paper presents a new way to solve the inverse problem of electrocardiography in terms of heart model parameters. The developed event estimation and recognition method uses a unified neural network (UNN)-based optimization system to determine the most relevant heart model parameters. A UNN-based preliminary ECG analyzer system has been created to reduce the searching space of the optimization algorithm. The optimal model parameters were determined by a relation between objective function minimization and robustness of the solution. The final evaluation results, validated by physicians, were about 96% correct. Starting from the fact that input ECGs contained various malfunction cases, such as Wolff-Parkinson-White (WPW) syndrome, atrial and ventricular fibrillation, these results suggest this approach provides a robust inverse solution, circumventing most of the difficulties of the ECG inverse problem.

1 Introduction

The most important health problem affecting large groups of people is related to the malfunction of the heart, usually caused by heart attack, rhythm disturbances and pathological degenerations. One of the main goals of health study is to predict these kinds of tragic events, and by identifying the patients situated in the most dangerous states, to make it possible to apply a preventing therapy.

Creating a heart model is important [1], as the computer, when applying traditional signal processing algorithms recognizes lots of waves, but it does not really "understand" what is happening. To overcome this, the computer needs to know the origin and the evolvement process of the ECG signal [2]. During signal processing, if the traditional algorithm finds an unrecognizable waveform, the model-based approach is activated, which tries to estimate the causes of the encountered phenomenon (e.g. quick recognition of ventricular fibrillation) [3].

The heart is a dynamic organ and places special demands on spatial techniques. To understand its physiology and patho-physiology, not only the electrical activity and spatial distribution of its structures is important, but also their movement during cardiac cycles. The measured ECG signal is influenced during repolarization by the mechanical movement of the heart. The main goal of the inverse problem of ECG is to characterize and reconstruct cardiac electrical events from measurements. In contrast

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 346–354, 2007.
springerlink.com

to the forward problem of electrocardiography, the inverse problem does not possess a mathematically unique solution and in order to improve stability, it needs to adopt regularization techniques.

Several approaches have been explored to handle the problem of multiple solutions by using equivalent cardiac generators (such as equivalent dipole and multi-pole), heart surface isochrones [4], or epicardial potential [5]. The high sensitivity of solutions to the different disturbances forced the investigators to explore regularization techniques [6]. These methods allow a significant progress, but the different uncertainty elements of the processing hinder the potentially beneficial ECG inverse solutions from becoming a routine clinical tool at present. Body surface potential mapping (BSPM) was developed to allow an almost complete data acquisition from the body surface [7] BSPM may have a great advantage over the standard 12-lead system in different situations due to deeper accessible information. Mirvis has shown some cases of BSPM recordings that clearly demonstrate the inadequacies of the standard ECG lead sets in a variety of pathologies [7]. As we know more about the depolarization – repolarization mechanism, we can understand better the internal function of the heart.

Many interesting biomedical applications of artificial neural networks are in the area of data processing [8]. The best known neural solutions involve multilayer perceptrons, Kohonen self-organizing networks, fuzzy or neuro-fuzzy systems, genetic algorithms and the combination of various solutions within a hybrid system [9]. A typical heart modeling system applies many neural networks and chooses the best one, while discarding the rest. Most efficient approaches usually are based on the combination of many classifiers utilizing either different classifier network structures or different data preprocessing methods [10].

The support vector machine (SVM), pioneered by Vapnik [11] had to solve the main drawbacks of conventional artificial neural networks (ANNs) such as:

- Modern biological problems are high-dimensional, and if the underlying mapping is not very smooth, the linear paradigm needs an exponentially increasing number of terms with an increasing dimensionality of the input space, that means an increase in the number of independent variables. This is known as 'the curse of dimensionality';
- The real-life data generation laws may typically be far from the normal distribution and a model-builder must handle this difference in order to construct an effective learning algorithm;
- The maximum likelihood estimator (and consequently the sum-of-error-squares cost function) should be replaced by a new induction paradigm that is uniformly better, in order to model properly non-Gaussian distributions.

The SVM classifiers became quite popular due to their robustness and stability [12]. A SVM used in a heart modeling system is rigorously based on statistical learning theory and simultaneously minimizes the training and test errors. Apart from that they produce a unique globally optimal solution and hence are extensively used in diverse applications including medical diagnosis [13].

This paper presents an event recognition study performed with ECG signal analysis and 3D heart model using unified neural networks. The main purpose is to evaluate the strength and weakness of each method, and to analyze the cooperation efficiency in malfunction diagnosis.

2 Materials and Methods

2.1 Unified Neural Networks

If we focus on the two-class classification case and consider linear discriminant functions, the respective decision hypersurface in the *n*-dimensional feature space is a hyperplane that can be described as:

$$g(x) = w^T \cdot x + w_0 = 0 \tag{1}$$

where $w = [w_1,...,w_n]^T$ is known as the weight vector and w_0 as the threshold value. For a given vector x_d, if the function $g(x_d) = 0$ than x_d situates on the decision hyperplane. The distance z of a vector x from the decision hyperplane is represented as: $z = \frac{\|g(x)\|}{\|w\|}$, where $\|w\| = \sqrt{\sum_{i=1}^{n} w_i^2}$.

In a classification problem, our purpose is the optimization of vector w in such a way, that the criteria function $J(w)$ is minimized. Let ω_1 and ω_2 be the two classes that we have to separate. We assume, that this task can be performed using a linear relation. This means that there exists at least one such a w_{sol} hyperplane that fulfill the following relations:

$$w_{sol}{}^T \cdot x > 0 \text{ for } \forall x \in \omega_1 \text{, and } w_{sol}{}^T \cdot x < 0 \text{ for } \forall x \in \omega_2 . \tag{2}$$

If we design a classifier, where the desired output is $y_{des} = +1$ for $\forall x \in \omega_1$ and $y_{des} = -1$ for $\forall x \in \omega_2$, and try to modify weights in vector w in such a way that the criteria function $J(w) = (y_{des} - x^T \cdot w)^2$ is minimized than we constructed a root mean square (RMS) error based separation.

A very popular classifier algorithm is based on SVM's. The main concept incorporates the search for the 'most robust solution' vector w_{sol} that gives the maximum possible margin. The margin is represented by the minimal distance z. This means the minimization of $\|w\|$. Although these methods deliver good results in a certain noise-free environment, in biomedical simulation such sterile conditions never occur. This is caused by measurement errors and the improper estimation of un-measurable biological parameters. RMS classifiers have the following drawbacks:

- Improper solution in case of asymmetric transfer functions
- Large estimation error of the criteria function in case of border-close high dispersion (uncertain) inputs

- In a noisy environment, the criteria function may possess multiple local minimal solutions that may cause low quality results
- The white noise is 'un-learnable' so the function $J(w)$ will saturate at an uncontrollable error level.

The SVM perform a much better result in hostile environment, but did not take into consideration the topology of the input vectors as Fig. 1 presents. To overcome the above mentioned problems, we propose for classification a UNN. The main difference between UNN and the described classifiers consists in the equation of criteria function:

$$J(w)=\lambda_d\cdot\left(y_{des}-x^T\cdot w\right)^2+\lambda_{sm}\cdot f_{sm}\left(\left\|x-x_\varepsilon\right\|\right)+\lambda_m\cdot f_m(z). \tag{3}$$

that is composed by a difference error, smoothness and margin part. Coefficients λ_d, λ_{sm}, λ_m are weighting the square error and the results of smoothing and margining functions represented by f_{sm} and f_m. The norm $\left\|x-x_\varepsilon\right\|$ represents the result of equation:

$$\left\|x-x_\varepsilon\right\|=\sum\nolimits_{i=1}^{n}(x^T\cdot w-(x-\varepsilon_i)^T\cdot w)^2, \tag{4}$$

where $\varepsilon_i=[0,...,0,e_i,0,...,0]^T$. The margin value is represented by z.

Although this example illustrates a linear separation, the UNN may be applied in non-linear environment.

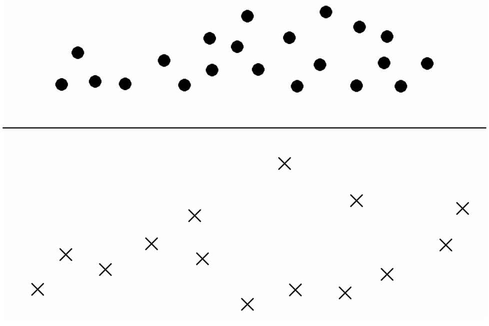


Fig. 1. Linearly separable two-class problem, resolved by an SVM-based classifier: although the circle group has a much higher density close to the separator line, the margin to the groups is the same

2.2 Study Records

The first signal resource was a 192-electrode measurement (BSPM) database (sampled at 1000 Hz with 12-bit resolution) obtained from the Research Institute for Technical Physics and Materials Science (MTA-MFA) of Budapest. These registrations contain various malfunction cases as WPW syndrome, atrial and ventricular fibrillation, flutter. In the second stage of the study we used 12-lead ECG registrations from our database. These signals were sampled at 500-1000 Hz with 12-bit resolution.

2.3 The Approach of ECG Inverse Problem

In contrast to methods that directly solve the matrix equation linking electrical sources with electrical potential fields to estimate ECG inverse solution, our approach indirectly obtains the result in terms of heart model parameters. The preliminary ECG analyzer system (PAS) is based on detailed, a priori knowledge of human anatomy and physiology, developed using an ANN, tested and validated by physicians in clinical environment (see Fig. 2).

In this study the PAS was used to obtain initial information on the site of origin of cardiac activation. The output of the ANN provides the initial heart model parameters. Then the BSPMs or 12-lead ECGs were simulated by the ECG generator unit, and the objective functions that assess the similarity between the measured and simulated signals were determined. The heart model parameters are adjusted with the aid of optimization algorithms or in certain cases physicians. The simulation procedure is performed until the objective functions satisfy the given convergence criteria. Finally the parameters are validated.

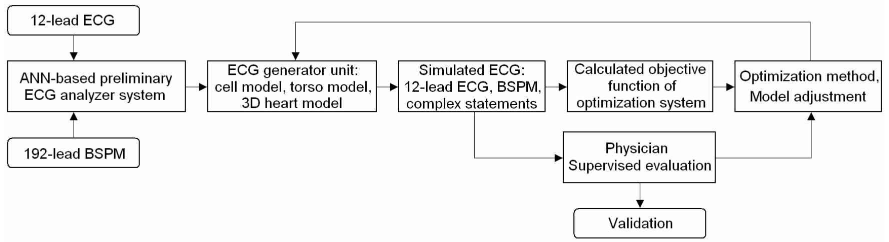

12-lead ECG
ANN-based preliminary ECG analyzer system
192-lead BSPM
ECG generator unit: cell model, torso model, 3D heart model
Simulated ECG: 12-lead ECG, BSPM, complex statements
Calculated objective function of optimization system
Optimization method, Model adjustment
Physician Supervised evaluation
Validation

Fig. 2. The schematic diagram of the heart analyzer and modeling method

2.4 ANN-Based Preliminary ECG Analyzer System

Application of a priori knowledge to reduce the searching space of heart model parameters is quite important. The PAS (see Fig. 3) was developed to determine roughly the cardiac status and state, which was then used to initialize the model parameters and decrease the searching space for the optimization system.

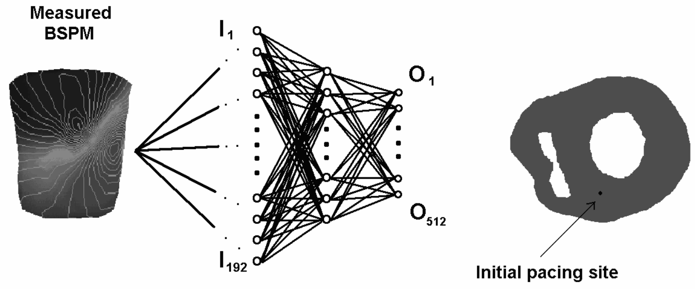


Fig. 3. The ANN-based preliminary analyzer system used to estimate the pacing site from ECG

In the present study, the PAS was developed based on a three-layer UNN as shown in Fig. 3. This network is capable of mapping the nonlinear input-output relationship, to a desired degree of accuracy. An adaptively weighted coefficient algorithm was used to train the ANN. The number of neurons in the input layer was set to 192, corresponding to the number of body surface electrodes used in the present simulation study. The hidden layer had 125 neurons that were determined heuristically. The output layer had 512 neurons, which corresponded to 32 ventricular myocardial segments of computer heart model: MS-1 to MS-32. Sixteen cardiac cellular units were selected for each of 32 myocardial segments in the ventricles, and each of these 16 x 32 = 512 sites was then paced in the forward simulation using the computer heart-torso model, generating the dataset for training the ANN.

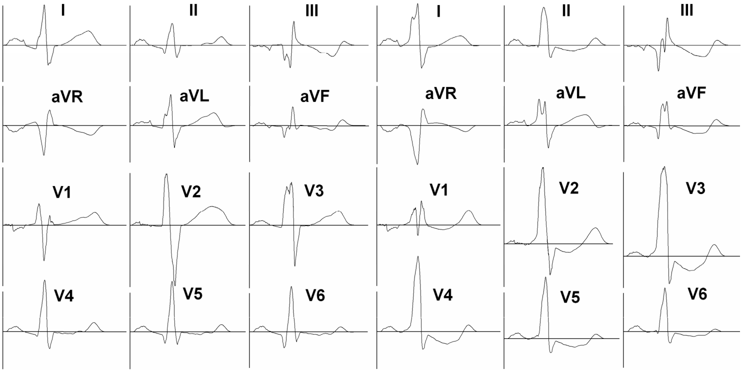


Fig. 4. The simulated ECG signal in normal and abnormal case (bypass tract)

3 Results

During the simulation, a parameter classification algorithm was applied to distinguish normal QRS complexes from abnormal ones, in order to determine the specific differences between the normal and abnormal parameter values. For normal cases the detection ratio is practically 100%. The signals presented in Fig. 4 were obtained via simulation using the initial parameter set for a normal and abnormal (bypass tract) situation.

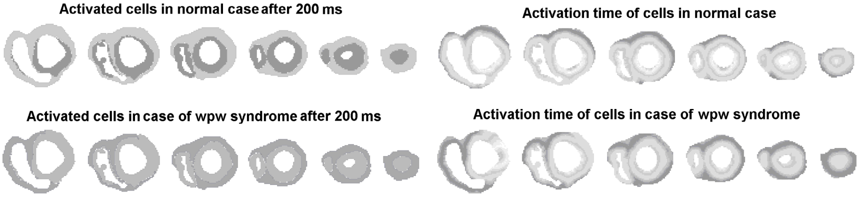

Activated cells in normal case after 200 ms
Activation time of cells in normal case
Activated cells in case of wpw syndrome after 200 ms
Activation time of cells in case of wpw syndrome

Fig. 5. The simulated depolarization in normal and abnormal case (bypass tract)

Fig. 5 presents the cell activation simulation that was effectuated and visualized after 200 ms of the rises of sinoatrial node excitation. In the right side of the image the activation time of each cell is showed. Neighbor slices are situated at 10 mm distance from each other, so totally a 5 centimeter wide ventricular tissue is visualized. Table 1 shows the efficiency of simulation for different cases. The evaluation of the simulated results was made by physicians. The performance was determined as the ratio of correct and total decisions.

Table 1. Simulation performance for normal and pathological cases

Pathological case	Decision number	Failed decisions	Performance
Normal	44	0	100.00%
Ectopic beat	21	0	100.00%
WPW syndrome	14	1	92.86%
Atrial flutter	22	1	95.45%
Atrial fibrillation	18	1	94.44%
Ventricular fibrillation	19	1	94.73%
Re-entry mechanisms	19	2	89.47%
Triggered activity	36	2	94.44%
Aberrant ventricular conduction	21	1	95.24%

4 Discussion

Table 1 reveals that the 3D heart simulation [14, 15] succeeds in most cases, such as WPW (Wolf Parkinson White) syndrome, pre-excitations, and tissue activation modeling. Re-entry mechanisms and triggered event cases represent the weak points of the simulation model. The application in practice of the model has several obstacles, which can be classified into the following groups:

- Effects of internal and external perturbations (such as environment, sympathetic and parasympathetic despondence);
- Lack of information on different elements of the model;
- Lack of technical background.

4.1 The Limitations of the Model

Several problems could be found, but the most important limitations are:

- The processes performed inside the cells are not well known, the behavior of the studied components cannot be determined with an acceptable precision;
- In critical cases, if a group of cells does not get the necessary food, it changes its behavior. A model created to simulate the normal behavior of the cell will not simulate it correctly in abnormal case;

- Because the structure of the heart differs from patient to patient, this structure is not known a priori, it has to be determined in real-time, based on the available information;
- The structure of the torso introduces the same problem. It is hard to determine the electrical conductivity and precise position of its elements.

4.2 Perturbation Phenomena and Technical Background

In case of human system identification the most important disturbing phenomena are:

- It is known, that respiration makes the heart change its shape and position. Although the motion of the heart can be tracked, it is not possible to determine from the ECG the amplitude of the motion;
- The continuous motion and displacement involves very hard problems. Because the motion has an effect on the behavior of all internal elements, the behavior of the heart will also be modified. The model has to follow the changes of the cell properties. For example: a resting man suddenly jumps out of the bed. The controlling mechanisms start their adjustment, the values of model parameters will change;
- Fever and respiration frequency can also cause alterations.

External events (the patient senses something annoying or pleasant) change the dependence between the previously measured signals, and the determined parameters. This is one of the causes why the perfect simulation of a human body is impossible.

At present, the performance of personal computers does not make possible the real-time determination of parameter values. The practical application is possible only in case of strongly parallel systems. The simplified model can be applied in real-time, but its efficiency is reduced because of the neglected parameters. The waveform of the simulated ECG in normal cases can be considered acceptable. The shape and duration of basic waves have realistic values. In case of abnormal cases the obtained waveform is not acceptable and more simulations are needed.

5 Conclusions

Regarding the fact, that computerized ECG diagnostics refer to several medical and technical problems, at the moment it cannot be applied as a standalone system. The short-term solution is the application of fuzzy systems and systems based on multi-agents, which, based on empirical information, make it possible to accomplish an adaptive advising system based on continuous transitions. If a hybrid system (neuronal-fuzzy and model-based approach, simultaneously) is built, it may become possible to learn the model via the knowledge of the traditional advising system, which after a suitable learning process, will be able to replace gradually the old system.

References

1. Thaker NV, Ferrero JM (1998) Electrophysiologic models of heart cells and cell networks. IEEE Eng Med Biol Soc Magazine, 17(5):73–83
2. MacLeod RS, Brooks DH (1998) Recent progress in inverse problems in electrocardiology. IEEE Eng Med Biol Soc Magazine, 17(1):73–83
3. Szilágyi SM (1998) Event Recognition, Separation and Classification from ECG Recordings. Proc 20th Ann Int Conf IEEE EMBS, pp 236–239
4. Cuppen JJM, van Oosterom A (1984) Model studies with inversely calculated isochrones of ventricular depolarization. IEEE Trans Biomed Eng,31:652–659
5. Guanglin L, Bin H (2001) Localization of the site of origin of cardiac activation by means of a heart-model-based electrocardiographic imaging approach. IEEE Trans Biomed Eng 48:660–669
6. Shahidi AV, Savard P, Nadeau R (1994) Forward and inverse problems of electrocardiography: modeling and recovery of epicardial potentials in humans. IEEE Trans Biomed Eng 41:249–256
7. Mirvis DM (1988) Validation of body surface electrocardiographic mapping. In Mirvis DM (ed) Body surface electrocardiographic mapping. Kluwer Acad Publ, Boston-Dordrecht-London, pp 63–74
8. Minami K, Nakajima H, Yoyoshima T (1999) Real time discrimination of the ventricular tachyarrhythmia with Fourier-transform neural network. IEEE Trans Biomed Eng 46: 179–185
9. Lagerholm M, Peterson C, Braccini G, Edenbrandt L, Sörnmo L (2000) Clustering ECG complexes using Hermite functions and self-organizing maps, IEEE Trans Biomed Eng 47:838–847
10. Osowski S, Hoai LT (2001) ECG beat recognition using fuzzy hybrid neural network, IEEE Trans Biomed Eng, 48:1265–1271
11. Osowski S, Hoai LT, Markiewicz T (2004) Support vector machine-based expert system for reliable heartbeat recognition, IEEE Trans Biomed Eng 51:582–589
12. Vapnik V (1998) Statistical learning theory. New York: Wiley
13. Smola A, Scholkopf B (1998) A tutorial on support vector regression. Royal Holloway College, Univ London, NeuroColt Tech Rep, NV2-TR-1998-030
14. Szilágyi SM, Benyó Z, Dávid L (2003) Heart model based ECG signal processing. Proc 5th IFAC Symp Modell Contr Biomed Syst, pp 213–217
15. Szilágyi SM, Benyó Z, Dávid L (2003) WPW syndrome identification and classification using ECG analysis. Proc World Congr Med Phys Biomed Eng, 4423.pdf

A Method for Creating Ensemble Neural Networks Using a Sampling Data Approach

Miguel Lopez[1,2], Patricia Melin[2], and Oscar Castillo[2]

[1] PhD Student of Computer Science in the Universidad Autonoma de Baja California, Tijuana B.C., México
danym23@aol.com
[2] Computer Science in the Graduate Division Tijuana, Institute of Technology Tijuana, B.C., Mexico
pmelin@tectijuana.mx, ocastillo@tectijuana.mx

Abstract. Ensemble Neural Networks are a learning paradigm where many neural networks are used together to solve a particular problem. In this paper, the relationship between the ensemble and its component neural networks is analyzed with the goal of creating of a set of nets for an ensemble with the use of a sampling-technique. This technique is such that each net in the ensemble is trained on a different sub-sample of the training data.

1 Introduction

The defining characteristic of an ensemble is that it involves combining a set of nets each of which essentially accomplishes the same task. The use of an ensemble can provide an effective alternative to the tradition of generating a population of nets, and then choosing the one with the best performance, whilst discarding the rest. The idea to ensemble neural networks is to find ways of exploiting instead the information contained in these redundant nets.

Combining estimators to improve performance has recently received more attention in the neural networks area, in terms of neural computing, there are two mains issues: first creation, or selection, of a set of nets to be combined in an ensemble; and second, the methods by which the outputs of the members of the ensemble are combined [2].

2 Methods for Creating Ensemble Members

There is no advantage to combining a set of nets which are identical; identical that is, in that they generalize in the same way. In principle, a set of nets could vary in terms of their weights, the time they took to converge, and even their architecture and yet constitute the same solution, since they resulted in the same pattern of errors when tested on a test set. There are many parameters which can be manipulated an efforts, to obtain a set of nets which generalize differently.

These include the following: initial conditions, training data, the topology of the nets, and the training algorithm.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 355 – 364, 2007.
springerlink.com

The main methods which have been employed for the creation of ensemble members are; changing the set of initial random weights, changing topology, changing the algorithm employed, changing the data[1].

The last one method, changing the data, is to be most frequently used for the creation of ensembles are those which involve altering the training data.

There are a number of different ways in which this can be done, which include: sampling data, disjoint training sets, boosting and adaptive resampling, different data sources, and preprocessing. These can be considerated individually, and it should be noted that ensemble could be created using a combination of two or more these techniques.

3 Sampling Data

A common approach to the creation of a set of nets for an ensemble is to use some form of sampling technique, such that each net in the ensemble is trained on a different subsample of the training data. Resampling methods which have been used for this purpose include cross-validation (Krogh and Vedelsby[2],1995) and bootstrapping(Breiman[4]), although in statistics the methods are better known as techniques for estimating the error of a predictor from limited sets of data[3].

In bagging (Breiman[5]) a training set containing N cases is perturbed by sampling with replacement (bootstrap) N times from the training set. The perturbed data set may contain repeats. This procedure can be repetead several times to create a number of different, although overlapping, data sets. Such statistical resampling techniques are particularly useful where there is a shortage of data.

Disjoint training sets, this method sampling without replacement, there is then no overlap between the data used to train different nets.

Boosting and adaptive resampling, the base of this algorithm is that training sets adaptively resampled , such that the weights in the resampling are incremented for those case which are most often misclassified [3].

4 Sampling Data Method for Creating Ensemble Neural Networks

The goal of the algorithm in the sampling data method is to find the data set of training for the creation of ensemble neural networks.

To train the network we used the Rastrigin´s function, wich has two dependent variables, and the function is defined as:

```
  Ras(p) = 20 + (p1 .^2 + p2 .^2 - 10.0 *( cos(2 *
pi .* p1) + cos(2 * pi .* p2)))
```
(1)

5 The Algorithm Sampling Data Method

1. The collected data of Rastrigin´s function are used to train the neural network.

$$P=[p(i)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots p(n)] \qquad (2)$$

2. In the process of training of the neural network the data that generate an error of greater or equal to the mean square error are stored like the new data set of training.

$$\text{If } e(i) > \text{RMSE then Psampling} = [p(i)\ldots\ldots\ldots\ldots\ldots\ldots p(n)] \quad (3)$$

3. The stored data represents the data set of sampling that is going away to use in the creation of the neural networks ensemble.

$$\text{Psampling} = [p(i)\ldots\ldots\ldots\ldots\ldots\ldots p(n)] \quad (4)$$

4. The output of the modular neural networks is evaluated by, wich is average of the output of the modules.

$$\text{Oneural\_mod} = \text{prom}[(\text{Mnn}(i)+\ldots..\text{Mnn}(n)] \quad (5)$$

5.1 Sampling Data Method

The neural networks in this study were trained using the back propagation algorithm with sigmoid activation function. During the learning phase, the output error by each data of training is verified, and if it’s greater to the mean square error, input data is stored to be used like the group of sampling data that were used for the creation of the neural networks ensemble. Rastrigin Function in figure 1 is used for the training the neural network using the data generated by the Rastrigin´s function without using sampling data method.

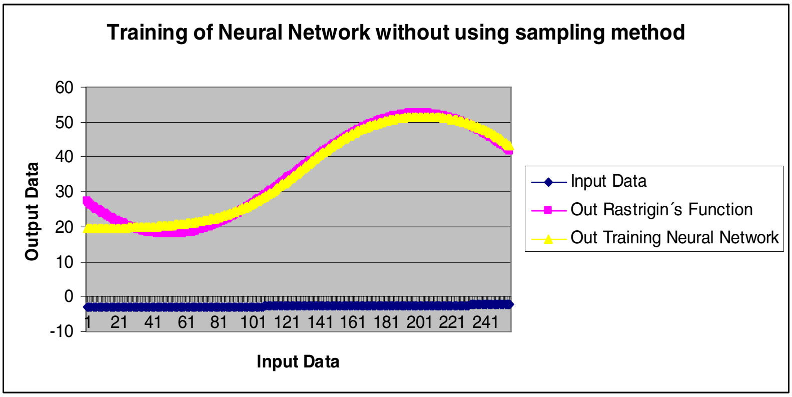


Fig. 1. Training of Neural Network without sampling data method

We show in Figure 2 the results of repeating 30 times the training of data and the average error achieved.

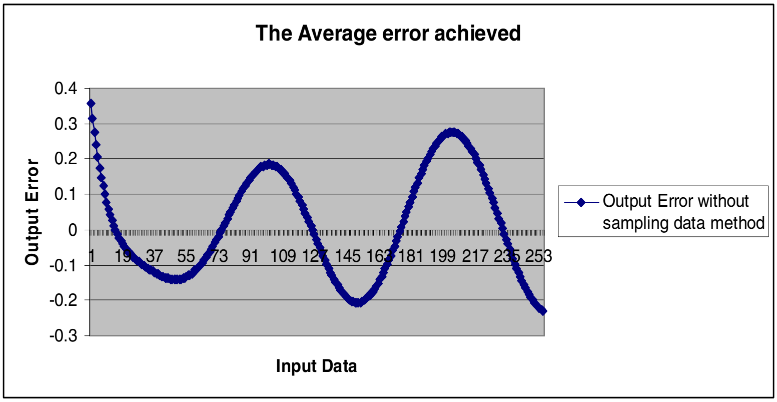


Fig. 2. The average error achieved without sampling data method

Rastrigin´s Function in figure 3 is used for training the neural network, using the data generated by Rastrigin´s function with a sampling data method.

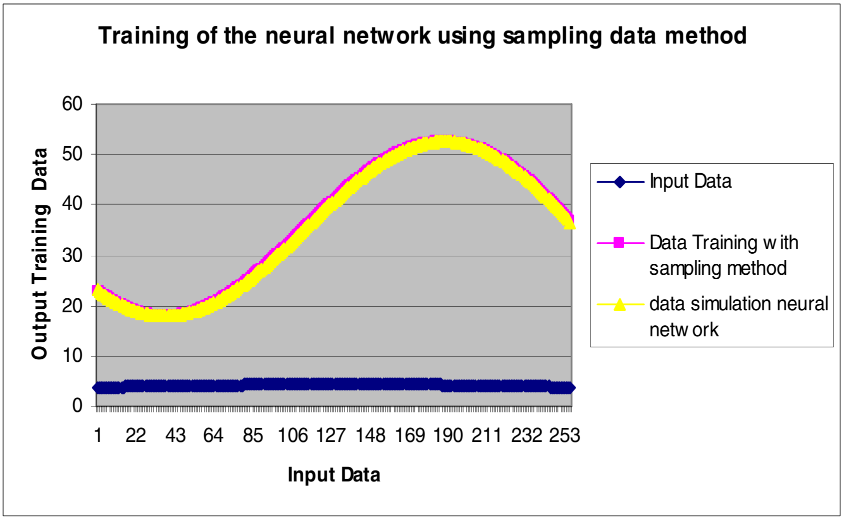


Fig. 3. Training of the neural network using sampling data method

Executing thirty times the training process and showing the average error achieved in figure 4, using data sampling method.

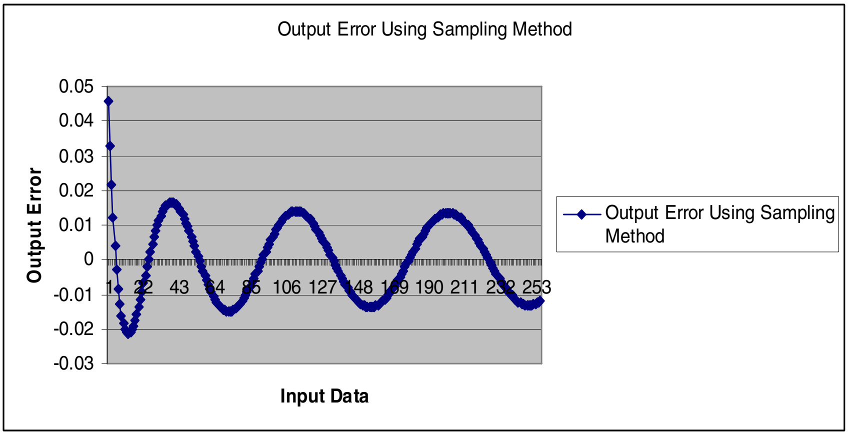


Fig. 4. The average error achieved using sampling data method

In Figure 5 we show a comparison between the Output Error using the Sampling Data Method and without in sampling data method.

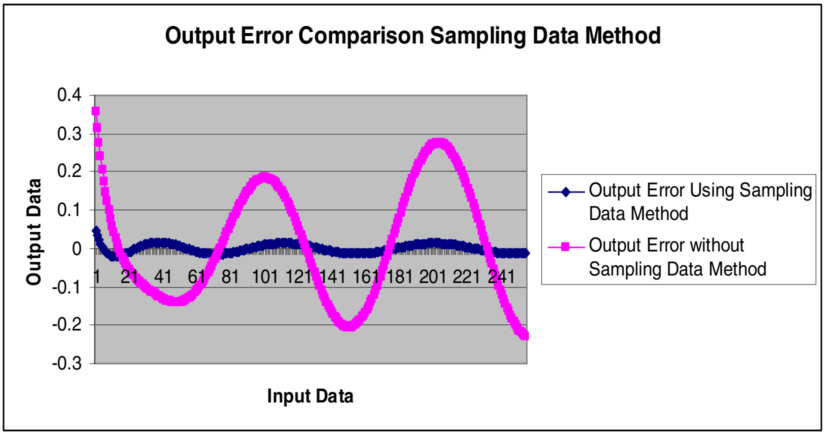


Fig. 5. Output Error Comparison sampling data method

5.2 Creating Modular Neural Networks

In the experimental results shown previously with the sampling method, for the creation of modular neural networks, we used a neural network of 1 hidden layer and 40 neurons, learning rate of .001, Rastrigin´s function, 2000 input data, training function Levenberg-Marquardt backpropagation. The First three modular neural networks created without using the sampling method are the following.

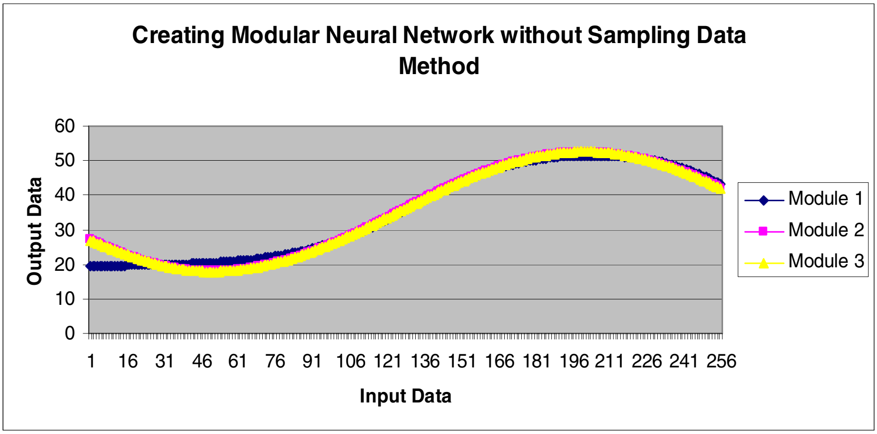


Fig. 6. Modular Neural Network without sampling data method

We show in Figure 6 the experimentals results creating modular neural networks without sampling data.

We show in Figure 7 the experimentals results creating modular neural networks using sampling data method.

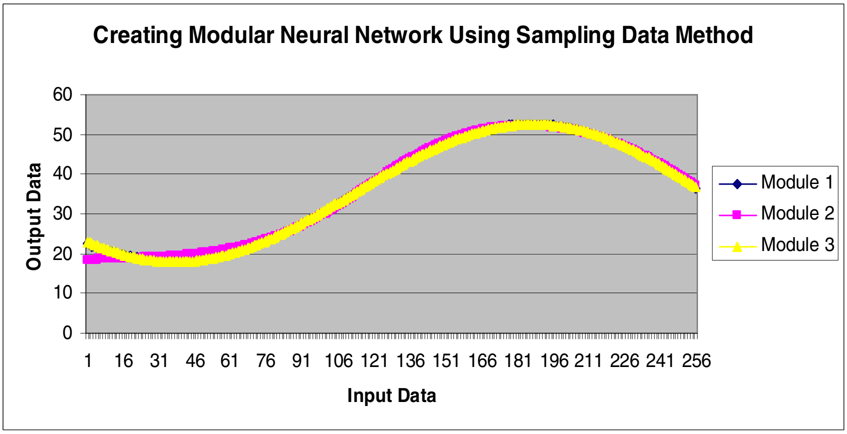


Fig. 7. Modular Neural Network using sampling data method

The previous results do not have a great difference using sampling method and without using it. Subsequently results of the output error of the modular neural networks, are presented and their average for the modular neural network.

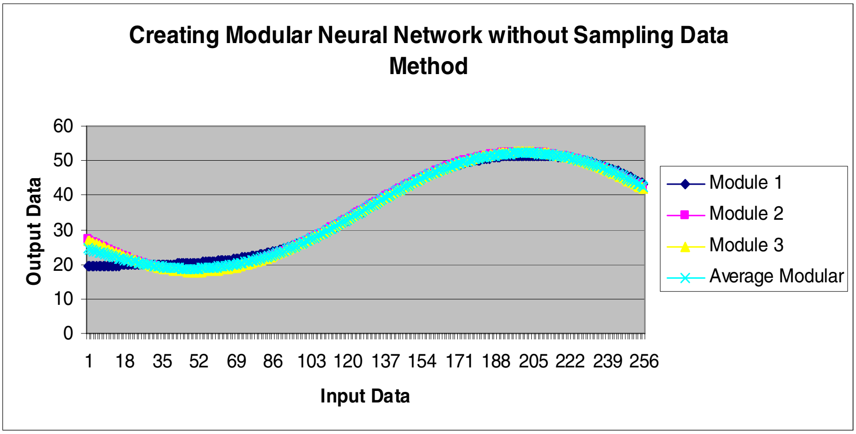


Fig. 8. Modular Neural Networks without sampling data method with modular average

We show in Figure 8 the experimental results creating modular neural network without sampling data method with modular average.

We show in Figure 9 the experimental results creating modular neural networks using sampling data method with modular average.

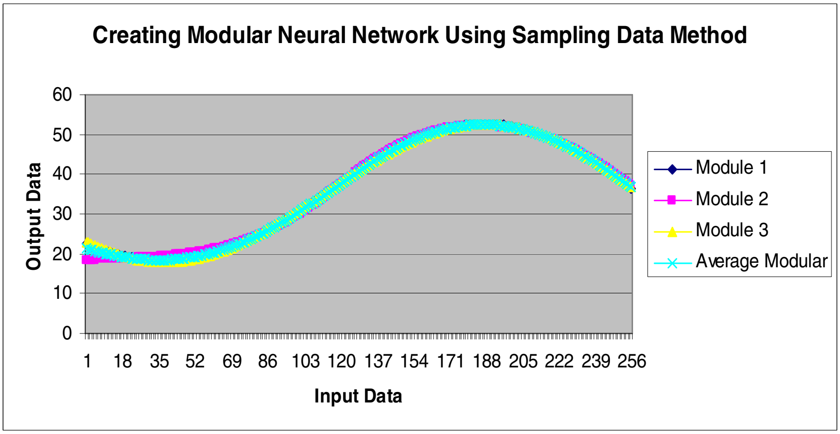


Fig. 9. Modular Neural Networks using sampling data method with modular average

We show in Figure 10 the results of the output error without the sampling data method.

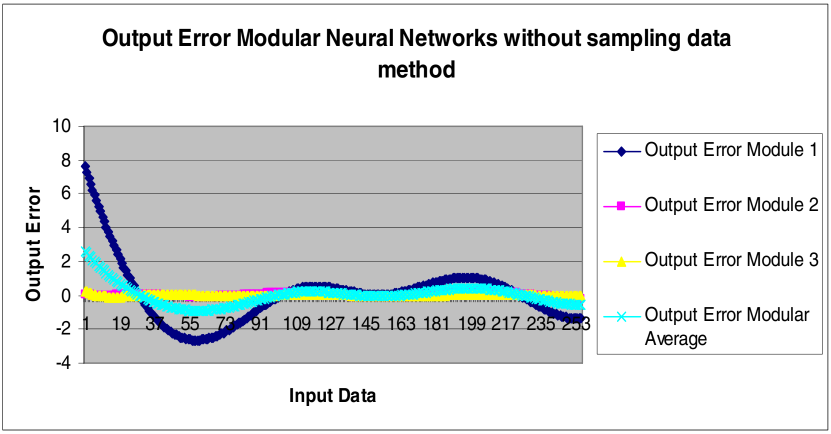


Fig. 10. Output Error of the Modular Neural Network without sampling data method

We show in Figure 11 the results the output error using sampling data method.

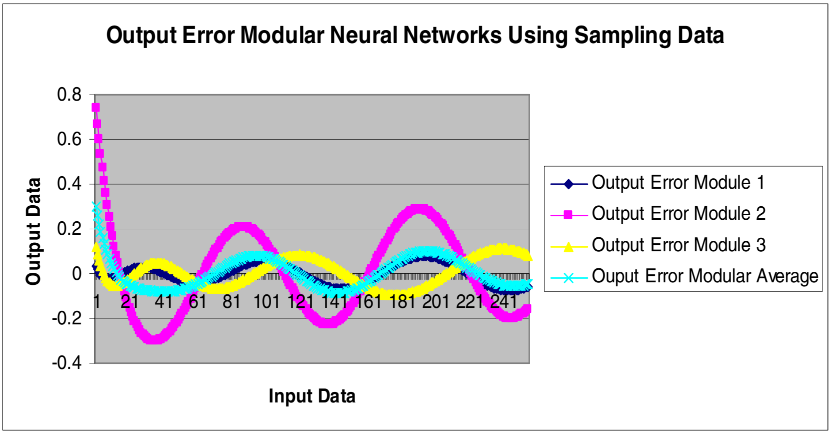


Fig. 11. Output Error of the Modular Neural Network using sampling data method

6 Conclusions

In conclusion, it's difficult to identify the best statistical sampling method to estimate the predictive accuracy of a neural network. The decision strongly depends on the complexity of the problem, and the number and variability of the available cases. Based on the experimental results obtained, we can comment that to utilize less data for the training of the modular neural networks is minimized the output error of the neural network, its necessary to prove a sampling method with different benchmark functions to establish a criterion that verifies it's operation to do a comparative with other sampling methods. It can be concluded that to a create modular neural networks is a good option to compare sampling data methods that allow the process of training modular neural network to work satisfactorily, without themselves making changes to with the architecture and parameters of the modular network neuronal.

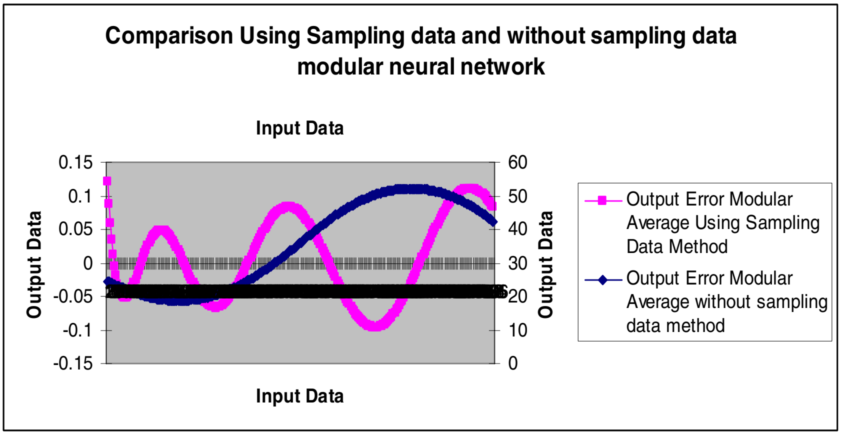


Fig. 12. Comparison using and without sampling data Modular Neural Networks

We show in figure 12 comparison results between using and without sampling data method.

Acknowledgments

We would like to thank CONACYT and DGEST for the financial support given to this research project. The PhD student in this work (Miguel Lopez) received an scholarship from CONACYT to perform the thesis work.

References

[1] The effect of data sampling on the performance evaluation of artificial neural networks in medical diagnosis, Georgia D. Tourassi, Carey E. Floyd, Med Decis Making. 1997 Apr-Jun;17(2):186-92. PMID: 9107614 [PubMed - indexed for MEDLINE].

[2] Feature Selection for Ensembles, David W. Opitz, Sixteenth National Conference on Artificial/ Intelligence (AAAI), (379-384). Orlando, FL.

[3] On combining artificial neural nets, Sharkey, A., Connection Science, Vol. 8, pp. 299-313, 1996**.**

[4] *Bagging predictors* ,Leo Breiman., Machine Learning, 24(2):123--140, 1996.

[5] Stacked Regressions ,Leo Breiman., Machine Learning 24(1): 49-64 (1996).

[6] Generating Accurate and Diverse Members of a Neural-Networks Ensemble, David W. Opitz , Jude W. Shavlik, Advances in Neural Information Processing Systems 8, D. S. Touretzky, M. Mozer, and M. Hasselmo, eds, MIT Press, Cambridge, MA, 1996, pp. 535-541.

[7] Overfitting and Diversity in Classification Ensembles based on Feature Selection, Padraig Cunningham, TCD Computer Science Technical Report, TCD-CS-2000-07.

[8] An Ensemble Neural Networks strategy for time series and spatio-temporal forecasting applications, Auroop R Ganguly, American Geophysical Union Spring Meeting, Washington, DC, Bras (2002).

[9] Face Recognition Using Ensembles of Netrworks, S. Gutta, J. Huang, B. Takacs and H. Wechsler, *icpr*, p. 50, 13th International Conference on Pattern Recognition (ICPR'96) - Volume 4, 1996. Vienna, Austria.

[10] Popular Ensemble Methods: An Empirical Study, Opitz, D. and Maclin, R. (1999), Journal of Artificial Intelligence Research, Volume 11, pages 169-198.

[11] Neural Nets and Diversity. Proceedings of the 14th International Conference on Computer Safety, Reliability and Security, Sharkey, A.J.C., Sharkey, N.E. and Chandroth, G.O., Belgirate, Italy, 1995, pp 375-389.

[12] Searching weight space for backpropagation solutions type, Sharkey, A.J.C., Sharkey, N.E., Neary, J., 1995, In L.F. Niklasson and M.B. Boden (Eds) Current Trends in Connectionism, Lawrence Erlbaum Associates, Hillsdale, N.J., pp 103-121.

[13] Generating accurate and diverse members of a neural networkensemble, D. W. Opitz and J. W. Shavlik ,in: Advances in Neural Information Processing Systems 8, D. S. Touretzky, M. Mozer, and M. Hasselmo, eds, MIT Press, Cambridge, MA, 1996, pp. 535-541.

[14] An Empirical Evaluation of Bagging and Boosting. Richard Maclin, David W. Opitz:, AAAI/IAAI 1997: 546-551.

[15] Modularity, combining and artificial neural nets Amanda.J. C. Sharkey. Connection Science, Vol. 9, No. 1, pages 3-10, 1997.

Pattern Recognition Using Modular Neural Networks and Fuzzy Integral as Method for Response Integration

Patricia Melin, Melchor Carranza, Pedro Salazar, and Rene Nuñez

Division of Graduate Studies, Tijuana Institute of Technology, Mexico

Abstract. We describe in this paper a new approach for pattern recognition using modular neural networks with a fuzzy logic method for response integration. We proposed a new architecture for modular neural networks for achieving pattern recognition in the particular case of human fingerprints. Also, the method for achieving response integration is based on the fuzzy Sugeno integral. Response integration is required to combine the outputs of all the modules in the modular network. We have applied the new approach for fingerprint recognition with a real database of fingerprints from students of our institution.

1 Introduction

Response integration methods for modular neural networks that have been studied, to the moment, do not solve well real recognition problems with large sets of data or in other cases reduce the final output to the result of only one module. Also, in the particular case of fingerprint recognition, methods of weighted-statistical average do not work well due to the nature of the fingerprint recognition problem. For these reasons, a new approach for fingerprint recognition using modular neural networks and fuzzy integration of responses was proposed in this paper.

The basic idea of the new approach is to divide a human fingerprint in to three different regions: the top, the middle and the bottom. Each of these regions is assigned to one module of the neural network. In this way, the modular neural network has three different modules, one for each of the regions of the human fingerprint. At the end, the final decision of fingerprint recognition is done by an integration module, which has to take into account the results of each of the modules. In our approach, the integration module uses the fuzzy Sugeno integral to combine the outputs of the three modules. The fuzzy Sugeno integral allows the integration of responses from the three modules of the top, middle and bottom of a human specific fingerprint. Other approaches in the literature use other types of integration modules, like voting methods, majority methods, and neural networks.

The new approach for fingerprint recognition was tested with a database of students and professors from our institution. This database was collected at our institution using a special scanner. The results with our new approach for fingerprint recognition on this database were excellent.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 365–373, 2007.
springerlink.com

2 Modular Neural Networks

There exists a lot of neural network architectures in the literature that work well when the number of inputs is relatively small, but when the complexity of the problem grows or the number of inputs increases, their performance decreases very quickly. For this reason, there has also been research work in compensating in some way the problems in learning of a single neural network over high dimensional spaces.

In the work of Sharkey [1], the use of multiple neural systems (Multi-Nets) is described. It is claimed that multi-nets have better performance or even solve problems that monolithic neural networks are not able to solve. It is also claimed that multi-nets or modular systems have also the advantage of being easier to understand or modify, if necessary.

In the literature there is also mention of the terms "ensemble" and "modular" for this type of neural network. The term "ensemble" is used when a redundant set of neural networks is utilized, as described in Hansen and Salomon [2]. In this case, each of the neural networks is redundant because it is providing a solution for the same task, as it is shown in Fig. 1.

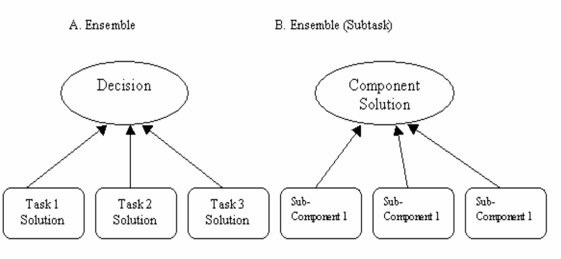


Fig. 1. Ensembles for one task and subtask

On the other hand, in the modular approach, one task or problem is decompose in subtasks, and the complete solution requires the contribution of all the modules, as it is shown in Fig. 2.

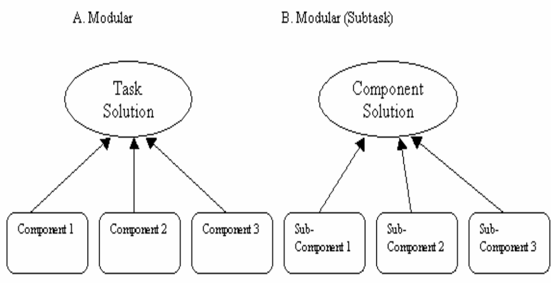


Fig. 2. Modular approach for task and subtask

2.1 Multiple Neural Networks

In this approach we can find networks that use strongly separated architectures. Each neural network works independently in its own domain. Each of the neural networks is build and trained for a specific task. The final decision is based on the results of the individual networks, called agents or experts.

One example of this is shown by Schmidt [3], in Fig. 3, where a multiple architecture is used, one module consists of a neural network trained for recognizing a person by the voice, while the other module is a neural network trained for recognizing a person by the image.

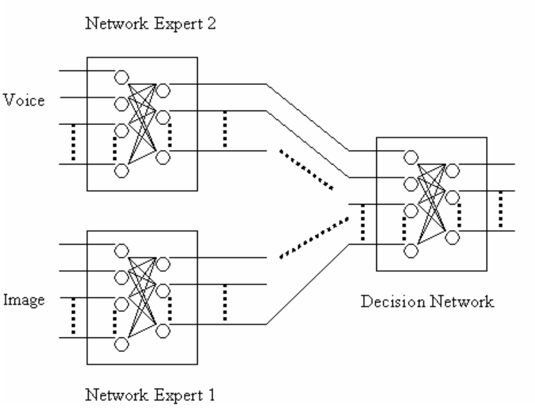


Fig. 3. Multiple networks for voice and image

The outputs by the experts are the inputs to the decision network, which is the one making the decision based on the outputs of the expert networks.

2.2 Main Architectures with Multiple Networks

Within multiple neural networks we can find three main classes of this type of networks [4]:

- Mixture of Experts (ME): The mixture of experts can be viewed as a modular version of the multi-layer networks with supervised training or the associative version of competitive learning. In this design, the local experts are trained with the data sets to mitigate weight interference from one expert to the other.
- Gate of Experts: In this case, an optimization algorithm is used for the gating network, to combine the outputs from the experts.
- Hierarchical Mixture of Experts: In this architecture, the individual outputs from the experts are combined with several gating networks in a hierarchical way.

3 Methods for Response Integration

The importance of this part of the architecture for pattern recognition is due to the high dimensionality of this type of problems. As a consequence in pattern recognition is good alternative to consider a modular approach. This has the advantage of reducing the time required of learning and it also increases accuracy. In our case, we consider dividing the images of a human fingerprint in three different regions, and applying a modular structure for achieving pattern recognition.

In the literature we can find several methods for response integration, that have been researched extensively, which in many cases are based on statistical decision methods. We will mention briefly some of these methods of response integration, in particular de ones based on fuzzy logic. The idea of using this type of methods is that the final decision takes into account all of the different kind of information available about the human face. In particular, we consider aggregation operators, and the fuzzy Sugeno integral [9].

Yager [10] mentions in his work, that fuzzy measures for the aggregation criteria of two important classes of problems. In the first type of problems, we have a set $Z=\{z_1,z_2,\ldots,z_n\}$ of objects, and it is desired to select one or more of these objects based on the satisfaction of certain criteria. In this case, for each $z_i \in Z$, it is evaluated $D(z_i)=G(A_i(z_i),\ldots,A_j(z_i))$, and then an object or objects are selected based on the value of G. The problems that fall within this structure are the multi-criteria decision problems, search in databases and retrieving of documents.

In the second type of problems, we have a set $G=\{G_1,G_2,\ldots,G_q\}$ of aggregation functions and object z. Here, each G_k corresponds to different possible identifications of object z, and our goal is to find out the correct identification of z. For achieving this, for each aggregation function G, we obtain a result for each z, Dk(z)=Gk(A1(z), A2(z), … ,An(z)). Then we associate to z the identification corresponding to the larger value of the aggregation function.

A typical example of this type of problems is pattern recognition. Where A_j corresponds to the attributes and $A_j(z)$ measures the compatibility of z with the attribute. Medical applications and fault diagnosis fall into this type of problems. In diagnostic problems, the A_j corresponds to symptoms associated with a particular fault, and G_k captures the relations between these faults.

Fuzzy integrals can be viewed as non-linear functions defined with respect to fuzzy measures. In particular, the "gλ-fuzzy measure" introduced by Sugeno [9] can be used to define fuzzy integrals. The ability of fuzzy integrals to combine the results of multiple information sources has been mentioned in previous works.

Definition 1. A function of sets $g:2^x \rightarrow (0.1)$ is called a fuzzy measure if:

1) $g(0)=0 \quad g(x)=1$
2) $g(A) \leq g(B)$ if $A \subset B$
3) if $\{A_i\} i^{\alpha} = 1$ is a sequence of increments of the measurable set then

$$\lim_{i \to \infty} g(A_i) = g\,(\lim_{i \to \infty} A_i) \tag{1}$$

From the above it can be deduced that g is not necessarily additive, this property is replaced by the additive property of the conventional measure.

From the general definition of the fuzzy measure, Sugeno introduced what is called “gλ-fuzzy measure”, which satisfies the following additive property: For every A, B $\subset$ X and A $\cap$ B = θ,

$$g(A \cup B) = g(A) + g(B) + \lambda\, g(A)g(B), \quad \text{for some value of } \lambda > -1. \tag{2}$$

This property says that the measure of the union of two disjunct sets can be obtained directly from the individual measures. Using the concept of fuzzy measures, Sugeno [9] developed the concept of fuzzy integrals, which are non-linear functions defined with respect to fuzzy measures like the gλ-fuzzy measure.

Definition 2. let X be a finite set and h:X→[0,1] be a fuzzy subset of X, the fuzzy integral over X of function h with respect to the fuzzy measure g is defined in the following way,

$$\begin{aligned} h(x) \circ g(x) &= \max_{E \subseteq X} [\, \min(\, \min_{x \in E} h(x), g(E))] \\ &= \sup_{\alpha \in [0,1]} [\min(\alpha, g(h_\alpha))] \end{aligned} \tag{3}$$

where h_α is the level set α of h,

$$h_\alpha = \{\, x \mid h(x) \geq \alpha \,\} \tag{4}$$

We will explain in more detail the above definition: h(x) measures the degree to which concept h is satisfied by x. The term min(h_x) measures the degree to which concept h is satisfied by all the elements in E. The value g(E) is the degree to which the subset of objects E satifies the concept measure by g. As a consequence, the obtained value of comparing these two quantities in terms of operator min indicates the degree to which E satifies both criteria g and min(h_x). Finally, operator max takes the greatest of these terms. One can interpret fuzzy integrals as finding the maximum degree of similarity between the objective and expected value.

4 Proposed Architecture and Results

In the experiments performed in this research work, we used 50 fingerprints that were taken with a scanner from students and professors of our Institution [11]. The images were taken in such a way that they had 198 pixels wide and 200 pixels high, with a resolution of 300x300 ppi, and with a color representation of a gray scale, some of these images are shown in Fig. 4. In addition to the training data (50 fingerprints) we did use 10 images that were obtained by applying noise in a random fashion, which was increased from 10 to 100%.

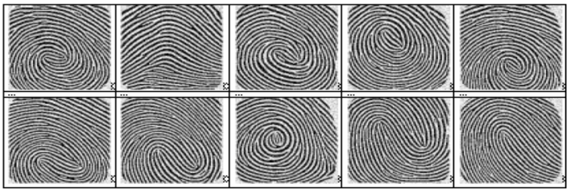


Fig. 4. Sample Images used for Training

4.1 Proposed Architecture

The architecture proposed in this work consist of three main modules, in which each of them in turn consists of a set of neural networks trained with the same data, which provides the modular architecture shown in Fig. 5.

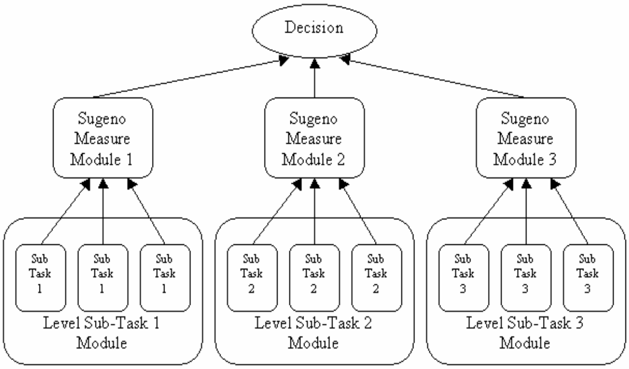


Fig. 5. Final Proposed Architecture

The input to the modular system is a complete photograph. For performing the neural network training, the images of the human fingerprints were divided in three different regions. The first region consists of the area on top, which corresponds to Sub Task 1. The second region consists of the area on the middle, which corresponds to Sub Task 2. The third region consists of the area on the bottom, which corresponds to Sub Task 3. An example of this image division is shown in Fig. 6.

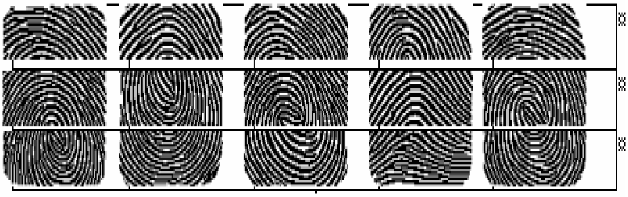


Fig. 6. Example of Image Division

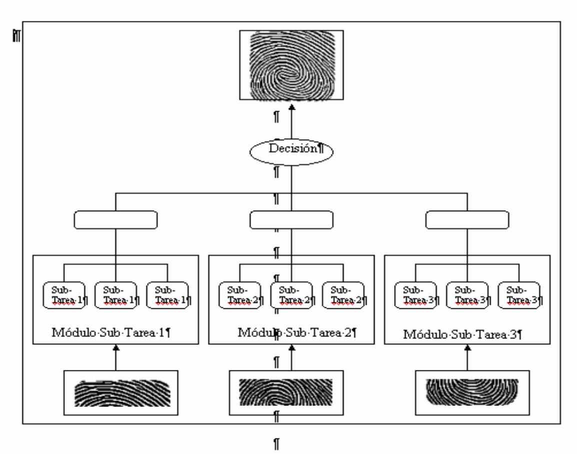


Fig. 7. Final Architecture Showing Inputs and Outputs

As output to the system we have an image that corresponds to the complete image that was originally given as input to the modular system, we show in Fig. 7 an example of this.

4.2 Description of the Integration Module

The integration modules performs its task in two phases. In the first phase, it obtains two matrices. The first matrix, called h, of dimension 3x3, stores the larger index values resulting from the competition for each of the members of the modules. The second matrix, called I, also of dimension 3x3, stores the image number corresponding to the particular index.

Once the first phase is finished, the second phase is initiated, in which the decision is obtained. Before making a decision, if there is consensus in the three modules, we can proceed to give the final decision, if there isn't consensus then we have search in matrix g to find the larger index values and then calculate the Sugeno fuzzy measures for each of the modules, using the following formula,

$$g(M_i) = h(A) + h(B) + \lambda\, h(A)\, h(B) \tag{5}$$

Where λ is equal to 1. Once we have these measures, we select the largest one to show the corresponding image.

4.3 Summary of Results

We describe in this section the experimental results obtained with the proposed approach using the 50 images as training data. We show in Table 1 the relation between accuracy (measured as the percentage of correct results) and the percentage of noise in the figures.

Table 1. Relation between the % of noise and the % of correct results

% of noise	% accuracy
0	100
10	100
20	100
30	100
40	95
50	100
60	100
70	95
80	100
90	75
100	80

In Table 1 we show the relation that exists between the % of noise that was added in a random (uniform) fashion to the testing data set, that consisted of the 50 original images, plus 200 additional images.

In Table 2 we show the reliability results for the pattern recognition system. Reliability was calculated as shown in the following equation.

$$\text{Reliability} = \frac{\text{correct results-error}}{\text{correct results}}$$

Table 2. Relation between reliability and accuracy

% errors	%reliability	%correct results
0	100	100.00
0	100	100.00
0	100	100.00
0	100	100.00
5	94.74	95.00
0	100	100.00
0	100	100.00
5	94.74	95.00
0	100	100.00
25	66.67	75.00
20	75	80.00

5 Conclusions

We showed in this paper the experimental results obtained with the proposed modular approach. In fact, we did achieve a 98.9% recognition rate on the testing data, even with an 80% level of applied noise. For the case of 100% level of applied noise, we did achieve a 96.4 % recognition rate on the testing data. The testing data included 10 images for each fingerprint in the training data. These 10 images were obtained by applying noise in a random fashion, increasing the level of noise from 10 to 100 %, to the training data. We also have to notice that it was achieved a 96.7 % of average reliability with our modular approach. These percentage values were obtained by averaging. In light of the results of our proposed modular approach, we have to notice that using the modular approach for human fingerprint pattern recognition is a good alternative with respect to existing methods, in particular, monolithic, gating or voting methods.

References

[1] A.J.C. Sharkey, Combining Artificial Neural Nets: Ensemble and Modular Multi-Nets Systems, Ed. Springer-Verlag, New York, 1998.

[2] L. K. Hansen and P. Salomon. Neural Network Ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990.

[3] S. Albrecht, A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors, 1996.

[4] Hsin-Chia Fu, Yen-Po Lee, Cheng-Chin Chiang and Hsiao-Tien Pao. Divide-and-Conquer Learning and Modular Perceptron Networks in IEEE Transaction on Neural Networks, vol. 12.

[5] Egbert J.W. Boers and Herman Kuiper. Biological Metaphors and the Design of Modular Artificial Neural Networks. Departments of Computer Science and Experimental and Theoretical Psychology at Leid University, the Netherlands. 1992.

[6] E. Ronco and P. Gawthrop. Modular neural networks: A State of the Art. Technical Report, Center for System and Control. University of Glasgow, Glasgow, UK, 1995.

[7] B. Lu and M. Ito. Task Decomposition and module combination based on class relations: modular neural network for pattern classification. Technical Report, Nagoya Japan, 1998.

[8] R. Murray-Smith and T. A. Johansen. Multiple Model Approaches to Modeling and Control. Taylor and Francis, 1997.

[9] M Sugeno. Theory of fuzzy integrals and its application, Doctoral Thesis, Tokyo Institute of Technology, 1974.

[10] R. R. Yager. Criteria Aggregations Functions Using Fuzzy Measures and the Choquet Integral, International Journal of Fuzzy Systems, Vol. 1, No. 2, December 1999.

[11] A. Quezada, Reconocimiento de Huellas Digitales Utilizando Redes Neuronales Modulares y Algoritmos Geneticos. Thesis of Computer Science, Tijuana Institute of Technology, 2004.

Part VI

Evolutionary Computation

A Differential Evolution Algorithm for Fuzzy Extension of Functions

Luciano Stefanini

University of Urbino "Carlo Bo", Urbino, Italy
LucSte@UniUrb.it

Abstract. The paper illustrates a differential evolution (DE) algorithm to calculate the level-cuts of the fuzzy extension of a multidimensional real valued function to fuzzy numbers. The method decomposes the fuzzy extension engine into a set of "nested" min and max box-constrained optimization problems and uses a form of the DE algorithm, based on multi populations which cooperate during the search phase and specialize, a part of the populations to find the the global min (corresponding to lower branch of the fuzzy extension) and a part of the populations to find the global max (corresponding to the upper branch), both gaining efficienty from the work done for a level-cut to the subsequent ones. A special version of the algorithm is designed to the case of differentiable functions, for which a representation of the fuzzy numbers is used to improve efficiency and quality of calculations. The included computational results indicate that the DE method is a promising tool as its computational complexity grows on average superlinearly (of degree less than 1.5) in the number of variables of the function to be extended.

1 Introduction

Appropriate use of fuzzy numbers in applications requires at least two features to be satisfied: (1) an easy way to represent and model fuzzy information with a high flexibility of shapes, e.g. allowing asymmetries or nonlinearities; (2) a relative simplicity and computational efficiency to perform exact fuzzy calculations or to obtain good approximations of the results. In general, the arithmetic operations on fuzzy numbers can be approached either by the direct use of the membership function (by the Zadeh extension principle [17]) or by the equivalent use of the *level-cuts* representation. By the *α-cuts* approach, it is possible to define a parametric representation of fuzzy numbers with the advantage of obtaining a wide family of fuzzy numbers (see [12]). It is well known that the fuzzy extension principle requires to solve a set of optimization problems and different heuristic methods have been proposed to obtain good solutions with a small number of function evaluations. Well known fundamental algorithms are the vertex method and its modifications (see [16] and [10]); the transformation method (see [6]) in its general or reduced versions (see [8] for an efficient implementation); a sparse grids method (see [9]). We suggest here two procedures based on the differential evolution (DE) method of Storn and Price (see [14], [15], [11]) and adapted to take into account both the nested property of *α-cuts* and the min and max problems over the same domains. In particular, we use simultaneous multiple populations that collaborate each other and specialize during the process to find all the required solutions. Computational results are reported that indicate the DE method as

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 377–386, 2007.
springerlink.com

a promising tool, as it exhibits, on average, superlinear computational complexity (of degree less than 1.5) in the number of variables.

2 Fuzzy Numbers and Fuzzy Extension Principle

We will consider fuzzy numbers and intervals, i.e. fuzzy sets defined over the field $\mathbb{R}$ of real numbers having a particular form. A general *fuzzy set* over $\mathbb{R}$ is usually defined by its membership function $\mu : \mathbb{R} \to T \subseteq [0,1]$ and a fuzzy (sub)set u of $\mathbb{R}$ is uniquely characterized by the pairs $(x, \mu_u(x))$ for each $x \in \mathbb{R}$; the value $\mu_u(x) \in [0,1]$ is the membership grade of x to the fuzzy set u. Fuzzy sets on $\mathbb{R}$ will be denoted by letters *u, v, w* and the corresponding membership functions by μ_u, μ_v, μ_w. Fundamental concepts in fuzzy theory are the *support*, the *level-sets* (or *level-cuts*) and the *core* of a fuzzy set:

Definition. *Let* μ_u *be the membership function of a fuzzy set u over* $\mathbb{R}$. *The support of u is the (crisp) subset of points of* $\mathbb{R}$ *at which the membership grade* $\mu_u(x)$ *is positive:* $\text{supp}(u) = \{x \mid x \in \mathbb{R}, \mu_u(x) > 0\}$. *For* $\alpha \in]0,1]$, *the* $\alpha -$ *level cut of u (or simply the* $\alpha - cut$ *) is defined by* $[u]_\alpha = \{x \mid x \in \mathbb{R}, \mu_u(x) \geq \alpha\}$ *and for* $\alpha = 0$ *by the closure of the support* $[u]_0 = cl\{x \mid x \in \mathbb{R}, \mu_u(x) > 0\}$. *The core of u is the set of elements of* $\mathbb{R}$ *having membership grade 1 i.e.* $core(u) = \{x \mid x \in \mathbb{R}, \mu_u(x) = 1\}$ *and we say that u is normal if* $core(u) \neq \varnothing$.

It is well-known that the *level-cuts* are "nested", i.e. $[u]_\alpha \subseteq [u]_\beta$ for $\alpha > \beta$. A particular class of fuzzy sets u is when the support is a convex set (A is said *convex* if $(1-t)x' + tx'' \in A$ for every $x', x'' \in A$ and all $t \in [0,1]$) and the membership function is quasi-concave, i.e. $\text{supp}(u)$ is convex and $\mu_u((1-t)x' + tx'') \geq \min\{\mu_u(x'), \mu_u(x'')\}$ for every $x', x'' \in \text{supp}(u)$ and $t \in [0,1]$. Equivalently, μ_u is quasi-concave if the level sets $[u]_\alpha$ are convex for all $\alpha \in [0,1]$. Finally, if the membership function is upper semi-continuous, then the level-cuts are closed.

Definition. *A fuzzy set u is a fuzzy quantity if the* $\alpha - cuts$ *are nonempty, compact intervals of the form* $[u]_\alpha = [u_\alpha^-, u_\alpha^+] \subset \mathbb{R}$. *If* $\exists \widehat{u} \in \mathbb{R}$ *such that* $core(u) = \{\widehat{u}\}$, *u is a fuzzy number and is called a fuzzy interval if* $\exists \widehat{u}^-, \widehat{u}^+ \in \mathbb{R}$, $\widehat{u}^- < \widehat{u}^+$ *such tha t* $core(u) = [\widehat{u}^-, \widehat{u}^+]$.

The "nested" property is the basis for the LU representation (L for lower, U for upper). We denote by E_F the set of fuzzy quantities.

Definition. *An LU-fuzzy quantity (number or interval) u is completely determined by any pair* $u = \left(u^-, u^+\right)$ *of functions* $u^-, u^+ : [0,1] \to \mathbb{R}$, *defining the end-points of the*

*$\alpha-cuts$, satisfying the three conditions: **(i)** $u^{-}:\alpha\rightarrow u_{\alpha}^{-}\in\mathbb{R}$ is a bounded monotonic nondecreasing left-continuous function $\forall\alpha\in]0,1]$ and right-continuous for $\alpha=0$; **(ii)** $u^{+}:\alpha\rightarrow u_{\alpha}^{+}\in\mathbb{R}$ is a bounded monotonic nonincreasing left-continuous function $\forall\alpha\in]0,1]$ and right-continuous for $\alpha=0$; **(iii)** $u_{\alpha}^{-}\leq u_{\alpha}^{+}$ $\forall\alpha\in[0,1]$.*

The support of u is the interval $[u_0^{-},u_0^{+}]$ and the core is $[u_1^{-},u_1^{+}]$. We refer to the functions $u_{(.)}^{-}$ and $u_{(.)}^{+}$ as the lower and upper branches on u, respectively. If the two branches $u_{(.)}^{-}$ and $u_{(.)}^{+}$ are continuous invertible functions then $\mu_u(.)$ is formed by two continuous branches, the left being the increasing inverse of $u_{(.)}^{-}$ on $[u_0^{-},u_1^{-}]$ and, the right, the decreasing inverse of $u_{(.)}^{+}$ on $[u_1^{+},u_0^{+}]$. There are many choices for $u_{(.)}^{-}$ and $u_{(.)}^{+}$. If we start with two decreasing shape functions $p(.)$ and $q(.)$ and with four numbers $u_0^{-}\leq u_1^{-}\leq u_1^{+}\leq u_0^{+}$ defining the support and the core of u then we can model $u_{(.)}^{-}$ and $u_{(.)}^{+}$ by $u_{\alpha}^{-}=u_1^{-}-(u_1^{-}-u_0^{-})p(\alpha)$ and $u_{\alpha}^{+}=u_1^{+}-(u_1^{+}-u_0^{+})q(\alpha)$ for all $\alpha\in[0,1]$. The simplest fuzzy quantities have linear branches: a trapezoidal fuzzy interval, denoted by $u=\langle a,b,c,d\rangle$, where $a\leq b\leq c\leq d$, has, for $\alpha\in[0,1]$, $\alpha-cuts$ $[u]_{\alpha}=[a+\alpha(b-a),d-\alpha(d-c)]$, obtaining a triangular fuzzy number if $b=c$. Consider now the extension of function $f:\mathbb{R}^n\rightarrow\mathbb{R}$ to a vector of n fuzzy numbers $u=(u_1,u_2,...,u_n)\in(E_F)^n$, with $k-th$ component $u_k\in E_F$ given by $[u_k]_{\alpha}=[u_{k,\alpha}^{-},u_{k,\alpha}^{+}]$ for *k=1,2,...,n* or $\mu_{u_k}:supp(u_k)\rightarrow[0,1]$ for *k=1,2,...,n* and denote $v=f(u_1,u_2,...,u_n)$.

The extension principle introduced by Zadeh is the basic tool for fuzzy calculus; it states that μ_v is given by

$$\mu_v(y)=\begin{cases}\sup\{\min\{\mu_{u_1}(x_1),...,\mu_{u_n}(x_x)\}\mid y=f(x_1,...,x_n)\} & \text{if } y\in Range(f)\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

where $Range(f)=\{y\in\mathbb{R}\mid\exists(x_1,...,x_n)\in\mathbb{R}^n \text{ s.t. } y=f(x_1,...,x_n)\}$.

For a continuous function $f:\mathbb{R}^n\rightarrow\mathbb{R}$, the $\alpha-cuts$ of the fuzzy extension v are obtained by solving the following box-constrained global optimization problems ($\alpha\in[0,1]$).

$$v_{\alpha}^{-}=\min\{f(x_1,x_2,...,x_n)\mid x_k\in[u_k]_{\alpha},\ k=1,2,...,n\} \tag{2}$$

$$v_{\alpha}^{+}=\max\{f(x_1,x_2,...,x_n)\mid x_k\in[u_k]_{\alpha},\ k=1,2,...,n\}. \tag{3}$$

If the function $f(x_1,x_2,...,x_n)$ is sufficiently simple, the analytical expressions for v_{α}^{-} and v_{α}^{+} can be obtained, as it is the case for many unidimensional elementary functions. For general functions, we need to solve numerically the global optimization problems (extower) and (extpper) above; general methods have been proposed and a very extended scientific literature is available. It is clear that in these cases we have only the possibility of fixing a finite set of values $\alpha \in \{\alpha_0,...,\alpha_M\}$ and obtain the corresponding v_α^- and v_α^+ pointwise; a sufficiently precise calculation requires M in the range from 10 to 100 or more (depending on the application and the required precision) and the computational time may become very high. To reduce these difficulties, various specific heuristic methods have been proposed; among others, the vertex method and its variants (see [3], [1] and [10]), the transformation method (see [6], [7], [8]), the interval arithmetic optimization with sparse grids (see [9]).

All the specific methods try to take computational advantage from the specific structure of "nested" optimizations (2)-(3) intrinsic in the properties of the $\alpha - cuts$. We will see that, at least in the differentiable case, the advantages of the LU representation appear to be quite interesting, based on the fact that a small number of points α is in general sufficient to obtain good approximations (this is the essential gain in using the slopes to model fuzzy numbers), so reducing the number of constrained *min* (2) and *max* (3) problems to be solved directly. On the other hand, finding computationally efficient extension solvers is still an open research field in fuzzy calculations.

3 Representation of LU-Fuzzy Numbers

As we have seen in the previous section, the LU representations of fuzzy numbers require to use appropriate (monotonic) shape functions to model the lower and upper branches of the $\alpha - cuts$. In this section we present the basic elements of a parametric representation of the shape functions proposed in [5] and [12] based on monotonic Hermite-type interpolation. The parametric representations can be used both to define the shape functions and to calculate the arithmetic operations by error controlled approximations.

We first introduce some models for "standardized" differentiable monotonic shape functions $p\ :[0,1] \to [0,1]$ such that $p(0) = 0$ and $p(1) = 1$ with $p(t)$ increasing on $[0,1]$ if interested to decreasing functions, we can start with an increasing function *p(.)*and simply define corresponding decreasing functions *q:[0,1]→ [0,1]* by *q(t)=1-p(t)* or *q(t)=p(φ(t))* where *φ:[0,1]→ [0,1]* is any decreasing bijection (e.g. *φ(t)=1-t*). Valid shape functions can be obtained by *p:[0,1] → [0,1]* satisfying the four Hermite interpolation conditions $p(0) = 0$, $p(1) = 1$ and $p'(0) = \beta_0$, $p'(1) = \beta_1$ for any value of the two nonnegative parameters $\beta_i \geq 0$, *i=0,1*. To explicit the parameters, we denote the interpolating function by $t \to p(t;\beta_0,\beta_1)$ for $t \in [0,1]$. We recall here two of the basic forms illustrated in [12]:

- *(2,2)-rational spline*: $p(t;\beta_0,\beta_1) = \frac{t^2+\beta_0 t(1-t)}{1+(\beta_0+\beta_1-2)t(1-t)}$;

- *mixed exponential spline*: $p(t;\beta_0,\beta_1) = \frac{1}{a}[t^2(3-2t)+\beta_0 - \beta_0(1-t)^a + \beta_1 t^a]$

where $a = 1+\beta_0+\beta_1$.

Note that in both forms we obtain a linear $p(t) = t,\ \ \forall t \in [0,1]$ if $\beta_0 = \beta_1 = 1$ and a quadratic $p(t) = t^2 + \beta_0 t(1-t)$ if $\beta_0 + \beta_1 = 2$. In order to produce different shapes we can either fix the slopes β_0 and β_1 (if we have information on the first derivatives at *t=0, t=1*) or we can estimate them by knowing values of *p(t)* in additional points. The model functions above can be adopted to represent the functions "piecewise", on a decomposition of the interval [0,1] into *N* subintervals $0=\alpha_0<\alpha_1<...<\alpha_{i-1}<\alpha_i<...<\alpha_N=1$. It is convenient to use the same subdivision for both the lower u_α^- and upper u_α^+ branches (we can always reduce to this situation by the union of two different subdivisions). In each subinterval $I_i=[\alpha_{i-1},\alpha_i]$, the values and the slopes of the two functions are

$$u^-_{(\alpha_{i-1})}=u^-_{0,i}\ ,\ u^+_{(\alpha_{i-1})}=u^+_{0,i}\ ,\ u^-_{(\alpha_i)}=u^-_{1,i}\ ,\ u^+_{(\alpha_i)}=u^+_{1,i} \tag{4}$$

$$u'^-_{(\alpha_{i-1})}=d^-_{0,i}\ ,\ u'^+_{(\alpha_{i-1})}=d^+_{0,i}\ ,\ u'^-_{(\alpha_i)}=d^-_{1,i}\ ,\ u'^+_{(\alpha_i)}=d^+_{1,i}$$

and by the transformation $t_\alpha=\frac{\alpha-\alpha_{i-1}}{\alpha_i-\alpha_{i-1}}$, $\alpha\in I_i$, each subinterval I_i is mapped into the standard [0,1] interval to determine each piece independently. Globally continuous or more regular $C^{(1)}$ fuzzy numbers can be obtained directly from the data (for example, $u^-_{1,i}=u^-_{0,i+1}$, $u^+_{1,i}=u^+_{0,i+1}$ for continuity and $d^-_{1,i}=d^-_{0,i+1}$, $d^+_{1,i}=d^+_{0,i+1}$ for differentiability at $\alpha=\alpha_i$). Let $p_i^\pm(t)$ denote the model function on I_i; we obtain $p_i^-(t)=p(t;\beta^-_{0,i},\beta^-_{1,i})$, $p_i^+(t)=1-p(t;\beta^+_{0,i},\beta^+_{1,i})$ with $\beta^-_{j,i}=\frac{\alpha_i-\alpha_{i-1}}{u^-_{1,i}-u^-_{0,i}}d^-_{j,i}$ and $\beta^+_{j,i}=-\frac{\alpha_i-\alpha_{i-1}}{u^+_{1,i}-u^+_{0,i}}d^+_{j,i}$ for j=0,1 so that, for $\alpha\in[\alpha_{i-1},\alpha_i]$ and *i=1,2,...,N*:

$$u^-_\alpha=u^-_{0,i}+(u^-_{1,i}-u^-_{0,i})p_i^-(t_\alpha)\ ,\ u^+_\alpha=u^+_{0,i}+(u^+_{1,i}-u^+_{0,i})p_i^+(t_\alpha). \tag{5}$$

The illustrated monotonic models suggest a first parametrization of fuzzy numbers on the trivial decomposition of interval [0,1], with N=1 (without internal points) and $\alpha_0=0,\alpha_1=1$. In this simple case, u can be represented by a vector of 8 components (the slopes corresponding to u_i^- are denoted by δu_i^-, etc.)

$$u=(u_0^-,\delta u_0^-,u_0^+,\delta u_0^+;u_1^-,\delta u_1^-,u_1^+,\delta u_1^+) \tag{6}$$

with $u_0^-,\delta u_0^-,u_1^-,\delta u_1^-$ for the lower branch u_α^- and $u_0^+,\delta u_0^+,u_1^+,\delta u_1^+$ for the upper branch u_α^+. On a decomposition $0=\alpha_0<\ \alpha_1<...<\ \alpha_N=1$ we can proceed piecewise. For example, a differentiable shape function requires 4(N+1) parameters

$$u=(\alpha_i;u_i^-,\delta u_i^-,u_i^+,\delta u_i^+)_{i=0,1,...,N}\text{ with} \tag{7}$$

$$u_0^-\le u_1^-\le...\le u_N^-\le u_N^+\le u_{N-1}^+\le...\le u_0^+\text{ (data)}$$

$$\delta u_i^-\ge 0,\delta u_i^+\le 0\text{ (slopes).}$$

and the branches are computed according to (5). In [5] and [12] we have analyzed the advantages of the LU representation in the computation of fuzzy expressions.

4 Differential Evolution Algorithms for Fuzzy Arithmetic

In this section we adopt an algorithmic approach to describe the application of differential evolution methods to calculate the fuzzy extension of multivariable function, associated to the LU representation of the fuzzy quantities involved. Let $v = f(u_1, u_2, ..., u_n)$ denote the fuzzy extension of a continuous function *f* in *n* variables; it is well known that the fuzzy extension of *f* to normal upper semicontinuous fuzzy intervals (with compact support) has the level-cutting commutative property (see [4]), i.e. the $\alpha - cuts$ $v_\alpha = [v_\alpha^-, v_\alpha^+]$ of v are the images of the $\alpha - cuts$ of $(u_1, u_2, ..., u_n)$ and are obtained by solving the optimization problems

$$(EP)_\alpha : \begin{cases} v_\alpha^- = \min\left\{ f(x_1, x_2, ..., x_n) \mid x_k \in [u_{k,\alpha}^-, u_{k,\alpha}^+],\ k = 1, 2, ..., n \right\} \\ v_\alpha^+ = \max\left\{ f(x_1, x_2, ..., x_n) \mid x_k \in [u_{k,\alpha}^-, u_{k,\alpha}^+],\ k = 1, 2, ..., n \right\}. \end{cases} \tag{8}$$

For simplicity, we will illustrate the case of differentiable representations (7) and differentiable function *f*.

Let $u_k = (u_{k,i}^-, \delta u_{k,i}^-, u_{k,i}^+, \delta u_{k,i}^+)_{i=0,1,...,N}$ be the LU-fuzzy representations of the input quantities for *k=1,2,…,n* and $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,...,N}$; the $\alpha - cuts$ of v are obtained by solving the box-constrained optimization problems (EP).

For each $\alpha = \alpha_i$, *i=0,1,…,N* the min and the max (EP) can occur either at a point whose components $x_{k,i}$ are internal to the corresponding intervals $[u_{k,i}^-, u_{k,i}^+]$ or are coincident with one of the extremal values; denote by $\overset{\frown}{x}_i^- = (\overset{\frown}{x}_{1,i}^-, ..., \overset{\frown}{x}_{n,i}^-)$ and $\overset{\frown}{x}_i^+ = (\overset{\frown}{x}_{1,i}^+, ..., \overset{\frown}{x}_{n,i}^+)$ the points where the min and the max take place; then $v_i^- = f(\overset{\frown}{x}_{1,i}^-, \overset{\frown}{x}_{2,i}^-, ..., \overset{\frown}{x}_{n,i}^-)$ and $v_i^+ = f(\overset{\frown}{x}_{1,i}^+, \overset{\frown}{x}_{2,i}^+, ..., \overset{\frown}{x}_{n,i}^+)$ and the slopes δv_i^-, δv_i^+ are computed (as *f* is differentiable) by

$$\delta v_i^- = \sum_{\substack{k=1 \\ \overset{\frown}{x}_{k,i}^- = u_{k,i}^-}}^{n} \frac{\partial f(\overset{\frown}{x}_{1,i}^-, ..., \overset{\frown}{x}_{n,i}^-)}{\partial x_k} \delta u_{k,i}^- + \sum_{\substack{k=1 \\ \overset{\frown}{x}_{k,i}^- = u_{k,i}^+}}^{n} \frac{\partial f(\overset{\frown}{x}_{1,i}^-, ..., \overset{\frown}{x}_{n,i}^-)}{\partial x_k} \delta u_{k,i}^+ \tag{9}$$

$$\delta v_i^+ = \sum_{\substack{k=1 \\ \overset{\frown}{x}_{k,i}^+ = u_{k,i}^-}}^{n} \frac{\partial f(\overset{\frown}{x}_{1,i}^+, ..., \overset{\frown}{x}_{n,i}^+)}{\partial x_k} \delta u_{k,i}^- + \sum_{\substack{k=1 \\ \overset{\frown}{x}_{k,i}^+ = u_{k,i}^+}}^{n} \frac{\partial f(\overset{\frown}{x}_{1,i}^+, ..., \overset{\frown}{x}_{n,i}^+)}{\partial x_k} \delta u_{k,i}^+ \tag{10}$$

If, for some reasons, the partial derivatives of *f* at the solution points are not available we can produce an estimation of the shapes δv_i^- and δv_i^+.

The idea of DE to find *min* or *max* of $\{ f(x_1, ..., x_n) \mid (x_1, ..., x_n) \in \mathrm{A} \subset \mathbb{R}^n \}$ is simple: Start with an initial "population" $(x_1, ..., x_n)^{(1)}, ..., (x_1, ..., x_n)^{(p)} \in \mathrm{A}$ of p feasible points; at each iteration obtain a new set of points by recombining randomly the individuals of the current population and by selecting the best generated elements to continue in the next generation. A typical recombination operates on a single component $j\in\{1,...,n\}$ and has the form (see [14] and [15],[2] for constraints handling) $x_j^{'}=x_j^{(r)}+\gamma[x_j^{(s)}-x_j^{(t)}]$, $\gamma\in]0,1]$ where $r,s,t\in\{1,2,...,p\}$ are chosen randomly. The components of each individual of the current population are modified to $x_j^{'}$ by a given probability q. Typical values are $\gamma\in[0.2,0.95]$ and $q\in[0.7,1.0]$.

Algorithm SPDE

```
Choose $p\approx 10n$ , $g_{\max}\approx 500$ , $q$ and $\gamma$ .
select $(x_1^l,...,x_n^l)$ , $x_k^l\in[u_{k,N}^-,u_{k,N}^+]\forall k$, $l$=1,…,2$p$ (initial population).
Evaluate $y^{(l)}=f(x_1^l,...,x_n^l), l=1,...,2p$ .
For i=N,N-1,…,0
  For g=1,2,…,g$_{\max}$
    For $l$=1,2,…,2$p$
      select $r,s,t\in\{1,2,...,2p\}$ , $j^*\in\{1,2,...,n\}$
      For j=1,2,…,n
        If $j=j^*$ or $ran(0,1)<q$
          Then $x'_j=x_j^{(r)}+\gamma[x_j^{(s)}-x_j^{(t)}]$ ;
                Ensure $u_{j,i}^-\le x'_j\le u_{j,i}^+$ (feasibility)
          Else $x'_j=x_j^{(l)}$ ;
      End For
      Evaluate $y=f(x'_1,...,x'_n)$ ;
      If $l\leq p$ and $y<y^{(l)}$ Then (min)
        substitute $(x_1,...,x_n)^{(l)}$ with $(x'_1,...,x'_n)$
      If $l>p$ and $y>y^{(l)}$ Then (max)
        substitute $(x_1,...,x_n)^{(l)}$ with $(x'_1,...,x'_n)$ ;
    End For
  End For
  Set $v_i^-=y^{l^*}=\min\{y^{(l)}\mid l=1,...,p\}$ ;
  $(\hat{x}^-_{1,i},...\hat{x}^-_{n,i})=(x_1,...,x_n)^{(l^*)}$ ;
  $v_i^+=y^{l^{**}}=\max\{y^{(l)}\mid l=1,...,p\}$ ;
  $(\hat{x}^+_{1,i},...\hat{x}^+_{n,i})=(x_1,...,x_n)^{(l^{**})}$ ;
  If i<N (prepare next level)
    select $(x_1^l,...,x_n^l)$ , $x_k^l\in[u_{k,i}^-,u_{k,i}^+]\forall k$,
    $l$=1,…,2$p$ including $\hat{x}_i^-$ and $\hat{x}_i^+$ .
End
```

Algorithm MPDE

```
Choose $p\approx 10n$ , $g_{\max}\approx 500$ , $q$ and $\gamma$ .
select $(x_1^{l,i},...,x_n^{l,i})$ , $x_k^{l,i}\in[u_{k,i}^-,u_{k,i}^+]\forall k,\forall i$ $l$=1,…,2$p$, $i=0,1,…,N$ (initial population).
Let $y^{(l,i)}=f(x_1^{l,i},...,x_n^{l,i}), l=1,...,2p, i=0,...,N$ .
Let $v_i^-=\min\{y^{(l,j)}\mid l=1,...,p, j=0,...,i\}$ ;
Let $v_i^+=\max\{y^{(l,j)}\mid l=1,...,p, j=0,...,i\}$ ;
Denote $\hat{x}_i^-$ , $\hat{x}_i^+$ where $v_i^-$ , $v_i^+$ are taken.
For g=1,2,…,g$_{\max}$
  For i=N,N-1,…,0
    For $l$=1,2,…,2$p$
      select $r,s,t\in\{1,2,...,2p\}$ , $k^*\in\{1,2,...,n\}$
      For k=1,2,…,n
        If $k=k^*$ or $ran(0,1)<q$
          Then $x'_k=x_k^{(r,i)}+\gamma[x_k^{(s,i)}-x_k^{(t,i)}]$ and
                 $x''_k=x_k^{(p+r,i)}+\gamma[x_k^{(p+s,i)}-x_k^{(p+t,i)}]$ ;
          (ensure $u_{k,i}^-\le x'_k, x''_k\le u_{k,i}^+$ feasibility)
          Else $x'_k=x_k^{(l,i)}, x''_k=x_k^{(p+l,i)}$ ;
      End For
      Evaluate $y'=f(x')$ and $y''=f(x'')$ ;
      If $y'<y^{(l,i)}$ substitute $x^{(l,i)}$ with $x'$; (min)
      If $y''>y^{(p+l,i)}$ substitute $x^{(p+l,i)}$ with $x''$; (max)
      Update $\{\hat{x}_j^-,\hat{x}_j^+,v_j^-v_j^+; j=0,...,i\}$
    End For
  End For
End For
```

To take into account the particular nature of our problem, we modify the basic procedure and examine two different strategies:

Strategy 1. start with the $(\alpha=1)-cut$ back to the $(\alpha=0)-cut$ so that the optimal solutions at a given level can be inserted into the "starting" populations of lower levels; use two distinct populations and perform the recombinations such that, during generations, one of the populations specializes to find the minimum and the other to find the maximum.

Strategy 2. use 2(N+1) populations to solve simultaneously all the box-constrained problems (EP); N+1 populations specialize for the min and the others for the max and the current best solution for level α_i is valid also for levels $\alpha_0,...,\alpha_{i-1}$.

Let $[u_{k,i}^-,u_{k,i}^+]$, *k=1,2,...,n* and $f:\mathbb{R}^n\rightarrow\mathbb{R}$ be given; we have to find v_i^- and v_i^+ according to (EP) for *i=0,1,...,N*. The slope parameters δv_i^-, δv_i^+ are computed by (9) and (10). The two strategies are implemented in algorithms SPDE and MPDE, respectively. Function $ran(0,1)$ generates a random uniform number.

5 Computational Results

The two algorithms SPDE and MPDE have been implemented using C++ and executed on a set of test functions with different dimension *n=2,4,8,16,32* as in the following table; the Rastrigin, Ackley and modified Rosenbrock functions are:

1. Rastrigin: $f(x_1,...,x_n)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$;
2. Ackley: $f=20+e-20\exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right]-\exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right]$;
3. Modified Rosenbrock: $f(x_1,...,x_n)=\sum_{i=1}^{n-1}\left[10(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$.

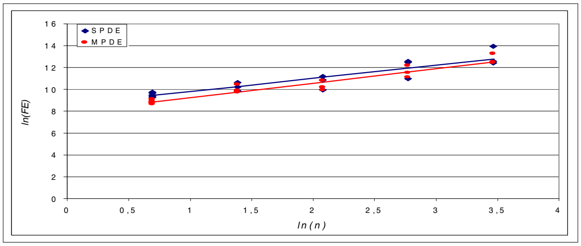
In the computations, the input fuzzy numbers are triangular and symmetric of the form $u_i=\left\langle a_i,\frac{a_i+b_i}{2},b_i\right\rangle$ with support given by the interval $[a_i,b_i]$; the applied supports are illustrated in the table.

If the extension algorithm is used in combinations with the LU-fuzzy representation for differentiable membership functions (and differentiable extended functions), then the number $N+1$ of $\alpha-cuts$ (and correspondingly of min/max optimizations) can be sufficiently small. Experiments in [5] and [12] motivated that $N=10$ is in general quite sufficient to obtain good approximations. The table reports the number of function evaluations FE_{SPDE} and FE_{MPDE} needed to the two algorithms SPDE and MPDE to reach the solution of the nested min/max optimization problems corresponding to the 11 $\alpha-cuts$ of the uniform $\alpha-$decomposition $\alpha_i=\frac{i}{10}$, *i=0,1,...,10* (*N=10* subintervals).

Table 1. Number of function evaluations for the test problems

#	*n*	*f(x₁,x₂,…,xₙ)*	*Supp(uᵢ)*	FESPDE	FEMPDE
1	2	$x_2\cos(\pi x_1)$	[0,5]×[1,5]	15400	8800
2	2	$\sqrt{(x_1-0.1)^4+(x_2-0.1)^4}$	[-1,1]×[0,2]	12760	6380
3	2	$(0.2+(x_1-2)^4+(x_2-2)^2)^{-1}$	[0,5]×[1,5]	16280	6600
4	2	$1+\frac{1}{2}x_1+\sin(2x_1-\pi/2)+2\cos(x_2)$	$[-2,2]^2$	12760	6380
5	2	$(x_1^2-x_2)^2+0.01(1-x_1)^2$	$[-2,2]^2$	12320	5720
6	2	$(1-\sqrt{x_1^2+x_2^2})\sin(\pi(x_1+\frac{1}{2}))$	[-1,1]×[-2,2]	11000	5500
7	2	$20\cos(x_1+x_2)-x_1^2-x_2^2$	$[-4,4]^2$	16720	7040
8	2	$\cos(2x_1+\sin(x_2))+\cos(x_2)$ $-0.1(x_1^2+x_2^2)$	$[-4,4]^2$	13200	8140
9	2	$20+e-20e^{-\frac{1}{20}(x_1^2+x_2^2)}-e^{\frac{1}{4}\cos(2\pi x_1)+\cos(2\pi x_2}$	$[-1,3]^2$	16280	6380
10	2	$100(x_2-x_1^2)^2+(x_1-1)^2$	$[-0.2,0.2]^2$	11440	6600
11	4	Rastrigin, n=4	$[0,3]^4$	40040	32560
12	4	Ackley, n=4	$[-1,3]^4$	27720	16280
13	4	Mod. Rosenbrock, n=4	$[-0.2,0.2]^4$	19360	18920
14	8	Rastrigin, n=8	$[0,3]^8$	72160	47520
15	8	Ackley, n=8	$[-1,3]^8$	51392	25344
16	8	Mod. Rosenbrock, n=8	$[-0.2,0.2]^8$	20416	19712
17	16	Rastrigin, n=16	$[0,3]^{16}$	292160	186560
18	16	Ackley, n=16	$[-1,3]^{16}$	255552	98560
19	16	Mod. Rosenbrock, n=16	$[-0.2,0.2]^{16}$	59136	63360
20	32	Rastrigin, n=32	$[0,3]^{32}$	1122176	560384
21	32	Ackley, n=32	$[-1,3]^{32}$	283008	252032
22	32	Mod. Rosenbrock, n=32	$[-0.2,0.2]^{32}$	250624	243584

The figure below represents the logarithm of the number of function evaluations vs the logarithm of the number *n* of arguments. It appears an almost linear relationship $\ln(FE_{SPDE})=a+b\ln(n)$ and $\ln(FE_{MPDE})=c+d\ln(n)$: the estimated coefficients are $a=8.615, b=1.20$ and $c=7.869, d=1.34$. If follows that the computational complexity of the proposed algorithms (on average for the 22 test problems) grows less than quadratically with the dimension *n* (SPDE is less efficient but grows slowly than MPDE). This is an interesting result, as all the existing methods for the fuzzy extension of functions are essentially exponential in *n*. Extended results and complete descriptions of the algorithms are discussed in [13]. The C++ source codes are available on request to the author; also a MatLab implementation is available.



References

1. H. K. Chen, W. K. Hsu, W. L. Chiang, A comparison of vertex method with JHE method, *Fuzzy Sets and Systems*, 95, 1998, 201-214.
2. K. Deb, An efficient constraint handling method for genetic algorithms, *Computer methods in applied mechanics and engineering*, 186, 2000, 311-338.
3. W. M. Dong, H. C. Shah, Vertex method for computing functions of fuzzy variables, *Fuzzy Sets and Systems*, 24, 1987, 65-78.
4. D. Dubois, H. Prade (ed), *Fundamentals of Fuzzy Sets*, Kluwer, Boston, The Handbooks of Fuzzy Sets Series, 2000.
5. M. L. Guerra, L. Stefanini, Approximate Fuzzy Arithmetic Operations Using Monotonic Interpolations, *Fuzzy Sets and Systems*, 150, 2005, 5-33.
6. M. Hanss, The transformation method for the simulation and analysis of systems with uncertain parameters, *Fuzzy Sets and Systems*, 130, 2002, 277-289.
7. M. Hanss, A. Klimke, On the reliability of the influence measure in the transformation method of fuzzy arithmetic, *Fuzzy Sets and Systems*, 143, 2004, 371-390.
8. A. Klimke, An efficient implementation of the transformation method of fuzzy arithmetic, Extended Preprint Report, 2003/009, Institut of Applied Analysis and Numerical Simulation, University of Stuttgard, Germany, 2003.
9. A. Klimke, B. Wohlmuth, Computing expensive multivariate functions of fuzzy numbers using sparse grids, *Fuzzy Sets and Systems*, 153, 2005, 432-453.
10. E. N. Otto, A. D. Lewis, E. K. Antonsson, Approximating $\alpha - cuts$ with the vertex method, *Fuzzy Sets and Systems*, 55, 1993, 43-50.
11. K. Price, An introduction to differential evolution, in D. Corne, M. Dorigo, F. Glover (ed.), *New Ideas in Optimization*, McGraw Hill, 1999, 79-108.
12. L. Stefanini, L. Sorini, M. L. Guerra, Parametric representation of fuzzy numbers and application to fuzzy calculus, *Fuzzy Sets and Systems*, 157, 2006, 2423-2455.
13. L. Stefanini, Differential Evolution Methods for the Fuzzy Extension of Functions, Working Paper Series EMS n. 103, University of Urbino, 2006.
14. R. Storn, K. Price, Differential Evolution: a simple and efficient heuristic for global optimization over continuous spaces, ICSI technical report TR-95-012, Berkeley University, 1995. Also, *Journal of Global Optimization*, 11, 1997, 341-359.
15. R. Storn, System design by constraint adaptation and differential evolution, *IEEE Transactions on Evolutionary Computation*, 3, 1999, 22-34.
16. K. L. Wood, K. N. Otto, E. K. Antonsson, Engineering design calculations with fuzzy parameters, *Fuzzy Sets and Systems*, 52, 1992, 1-20.
17. L. A. Zadeh, Fuzzy sets, *Information and Control,* 8, 1965, 338-353.

Use of Pareto-Optimal and Near Pareto-Optimal Candidate Rules in Genetic Fuzzy Rule Selection

Hisao Ishibuchi, Isao Kuwajima, and Yusuke Nojima

Department of Computer Science and Intelligent Systems, Osaka Prefecture University,
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
{hisaoi@,kuwajima@ci.,nojima@}cs.osakafu-u.ac.jp
http://www.ie.osakafu-u.ac.jp/~hisaoi/ci_lab_e

Abstract. Genetic fuzzy rule selection is an effective approach to the design of accurate and interpretable fuzzy rule-based classifiers. It tries to minimize the complexity of fuzzy rule-based classifiers while maximizing their accuracy by selecting only a small number of fuzzy rules from a large number of candidate rules. One important issue in genetic fuzzy rule selection is the prescreening of candidate rules. If we use too many candidate rules, a large computation load is required to search for good rule sets. On the other hand, good rule sets will not be found when promising fuzzy rules are not included in the candidate rule set. It is essential for the success of genetic fuzzy rule selection to use a tractable number of promising fuzzy rules as candidate rules. In this paper, we propose an idea of using Pareto-optimal and near Pareto-optimal fuzzy rules as candidate rules in genetic fuzzy rule selection. Pareto-optimality is defined by two well-known data mining criteria: support and confidence. To extract not only Pareto-optimal but also near Pareto-optimal fuzzy rules, we modify Pareto dominance using a dominance margin ε. Through computational experiments, we examine the effect of the proposed idea on multiobjective genetic fuzzy rule selection.

1 Introduction

The main advantage of fuzzy rule-based systems over other nonlinear systems such as neural networks is their linguistic interpretability [2], [3], [12]. Human users can understand fuzzy rule-based systems through linguistic interpretation of fuzzy rules. In this sense, fuzzy rule-based systems are viewed as transparent models (i.e., white-box models) while other nonlinear systems are usually black-box models. In addition to the linguistic interpretability of each fuzzy rule, various aspects are related to the interpretability of fuzzy rule-based systems (e.g., the number of fuzzy rules and the number of antecedent conditions of each fuzzy rule). Genetic fuzzy rule selection was proposed in [13] for the design of accurate and interpretable fuzzy rule-based classifiers by minimizing the number of fuzzy rules while maximizing their accuracy. A small number of fuzzy rules were selected from a large number of candidate rules to construct an accurate and interpretable fuzzy rule-based classifier. A standard single-objective genetic algorithm (SOGA) was used to optimize a weighted sum fitness function defined by an accuracy measure and a complexity measure. Genetic fuzzy rule selection was generalized to two-objective rule selection in [10] where a multiobjective genetic algorithm (MOGA) was used to search for non-dominated fuzzy rule

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 387–396, 2007.
springerlink.com

sets (i.e., non-dominated fuzzy rule-based classifiers) with respect to the accuracy and complexity measures. It was further generalized to three-objective rule selection in [11] by introducing the total number of antecedent conditions as an additional complexity measure. Currently multiobjective design of fuzzy rule-based systems is an active research area in the field of fuzzy systems [9], [14], [19], [20].

In the field of data mining, MOGAs were used to search for non-dominated rules with respect to well-known rule evaluation criteria: support and confidence. Such an MOGA-based data mining approach was first proposed in [5]. Recently a similar approach was used for multiobjective genetic fuzzy data mining [15].

One important issue in genetic fuzzy rule selection is the prescreening of candidate rules. Let N be the number of candidate rules. Any subset of the N candidate rules is represented by a binary string of length N. Thus the size of the search space is 2^N. When we have too many candidate rules (i.e., when N is too large), it is very difficult to efficiently search for good rule sets. A large computation load is required to find good rule sets in the search space of size 2^N. On the other hand, genetic fuzzy rule selection is not likely to find good rule sets when N is too small. In the application of genetic fuzzy rule selection to low-dimensional pattern classification problems with only a few attributes, we can examine all combinations of antecedent fuzzy sets to generate fuzzy rules. All the generated fuzzy rules can be used as candidate rules in genetic fuzzy rule selection. It is, however, impractical to use all the generated fuzzy rules as candidate rules for high-dimensional pattern classification problems with many attributes because the total number of possible fuzzy rules exponentially increases with the number of attributes. Thus we need a heuristic rule evaluation criterion for the prescreening of candidate rules in genetic fuzzy rule selection in its application to high-dimensional pattern classification problems [14]. Whereas various rule evaluation criteria such as confidence, support and their combinations are applicable, it is not easy to choose a single criterion because their effectiveness is problem-dependent. It is not easy to appropriately specify parameter values (e.g., the minimum support and the minimum confidence) involved in each criterion, either.

In this paper, we propose an idea of using Pareto-optimal and near Pareto-optimal rules with respect to support and confidence as candidate rules in genetic fuzzy rule selection. We modify Pareto dominance by introducing a dominance margin ε in the same manner as [18] to extract not only Pareto-optimal rules but also near Pareto-optimal rules. A similar modification method was also used to improve the performance of MOGAs under the name of ε-dominance [7], [16].

This paper is organized as follows. First we explain fuzzy rule-based classifiers in Section 2. Next we explain genetic fuzzy rule selection in Section 3. Then we examine the effect of using Pareto-optimal and near Pareto-optimal fuzzy rules as candidate rules on multiobjective genetic fuzzy rule selection in Section 4. Finally we conclude this paper in Section 5.

2 Fuzzy Rule-Based Classifiers

Let us assume that we have m training patterns $\mathbf{x}_p = (x_{p1}, x_{p2}, ..., x_{pn})$, $p = 1, 2, ..., m$ from M classes in an n-dimensional continuous pattern space $[0, 1]^n$ where x_{pi} is the attribute value of the p-th training pattern for the i-th attribute. For our pattern classification problem, we use fuzzy rules of the following form:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \qquad (1)$$

where R_q is the label of the q-th fuzzy rule, $\mathbf{x} = (x_1, x_2, ..., x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set, C_q is a class label, and CF_q is a rule weight. We denote R_q in (1) as "$\mathbf{A}_q \Rightarrow$ Class C_q" where $\mathbf{A}_q = (A_{q1}, A_{q2}, ..., A_{qn})$.

Since we usually have no *a priori* information about an appropriate granularity of the fuzzy discretization for each attribute, we simultaneously use multiple fuzzy partitions with different granularities to extract candidate fuzzy rules. In computational experiments, we use four fuzzy partitions with triangular fuzzy sets in Fig. 1. In addition to the 14 fuzzy sets in Fig. 1, we also use the domain interval [0, 1] itself as an antecedent fuzzy set in order to represent a *don't care* condition.

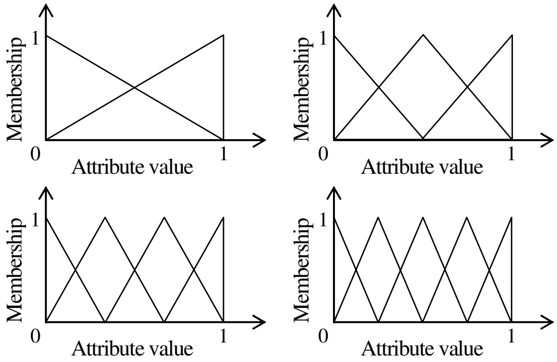


Fig. 1. Antecedent fuzzy sets used in computational experiments

Since we use the 15 antecedent fuzzy sets for each of the n attributes, the total number of combinations of the antecedent fuzzy sets is 15^n. Each combination is used in the antecedent part of the fuzzy rule in (1). The consequent class and the rule weight of each fuzzy rule can be easily specified from compatible training patterns in a heuristic manner [12]. This means that we can easily generate a large number of fuzzy rules by specifying the consequent class and the rule weight for each of the 15^n combinations of the antecedent fuzzy sets. It is, however, very difficult for human users to manually examine such a large number of fuzzy rules. It is also very difficult for human users to intuitively understand long fuzzy rules with many antecedent conditions. Thus we examine only short fuzzy rules of length $L_{\max}$ or less (e.g., $L_{\max} = 3$). This restriction on the rule length is to find fuzzy rule-based classifiers with high interpretability.

Let S be a set of fuzzy rules of the form in (1). That is, S is a fuzzy rule-based classifier. When an input pattern $\mathbf{x}_p$ is presented to S, $\mathbf{x}_p$ is classified by a single winner rule that has the maximum product of the compatibility grade and the rule weight (see [12] for various fuzzy reasoning methods for classification problems). In this paper, we use the product operator to calculate the compatibility grade.

3 Genetic Fuzzy Rule Selection

Genetic fuzzy rule selection is a two-step approach to the design of fuzzy rule-based classifiers. In the first phase, a large number of fuzzy rules are generated as candidate rules. A heuristic rule evaluation criterion is usually used for the prescreening of candidate rules [9], [11], [12], [14]. In the second phase, a genetic algorithm is used to select a small number of candidate rules.

In the field of data mining [1], two rule evaluation criteria (confidence and support) have been often used to evaluate an association rule. Fuzzy versions of these two criteria can be written as follows [12]:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \Bigg/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\,, \tag{2}$$

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\,, \tag{3}$$

where $c(\cdot)$ and $s(\cdot)$ are the confidence and the support of a fuzzy rule, respectively.

Association rule mining techniques extract all rules that satisfy the prespecified minimum confidence and support [1]. In computational experiments, we extract all fuzzy rules of length three or less that satisfy this condition. The extracted fuzzy rules are used as candidate rules in genetic fuzzy rule selection.

Let N be the number of candidate rules. A rule set S, which is a subset of the N candidate rules, is represented by a binary string of length N. Our fuzzy rule selection problem is formulated as follows [11]:

$$\text{Maximize } f_1(S)\text{, and minimize } f_2(S) \text{ and } f_3(S)\,, \tag{4}$$

where $f_1(S)$ is the number of correctly classified training patterns by S, $f_2(S)$ is the number of fuzzy rules in S, and $f_3(S)$ is the total number of antecedent conditions (i.e., total rule length) in S. It should be noted that each fuzzy rule has a different number of antecedent conditions. This is because we use *don't care* as a special antecedent fuzzy set, which is not counted in the number of antecedent conditions.

In this section, we use a standard single-objective genetic algorithm (SOGA) to maximize the following weighted sum fitness function:

$$\mathit{fitness}(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S)\,, \tag{5}$$

where w_i is a non-negative weight for the i-th objective. We use the $(\mu+\lambda)$-ES generation update mechanism with $\mu=\lambda$ in our SOGA.

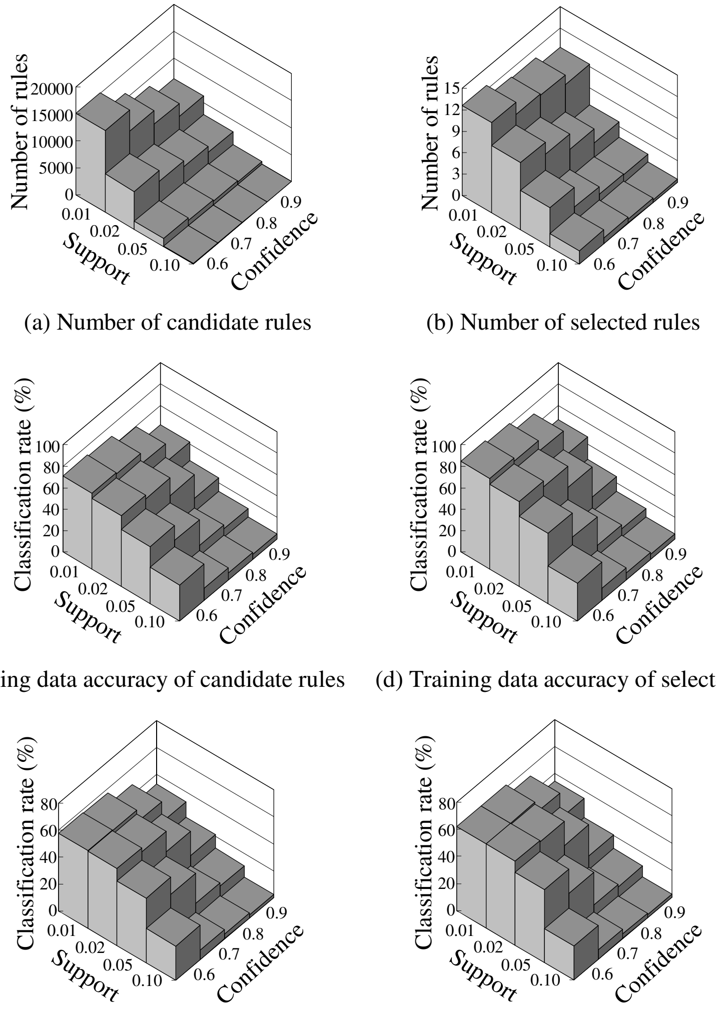
We applied genetic fuzzy rule selection to the glass data set (a nine-dimensional patterns classification problem with 214 patterns from six classes) in the UCI machine learning repository using the following parameter specifications:

Minimum confidence: 0.6, 0.7, 0.8, 0.9,
Minimum support: 0.01, 0.02, 0.05, 0.10,
Weight vector in the fitness function in (5): (10, 1, 1),
Population size: 200 (i.e., $\mu = \lambda = 200$),
Crossover probability: 0.9 (uniform crossover),
Mutation probability: 0.05 ($1 \rightarrow 0$) and $1/N$ ($0 \rightarrow 1$),
Termination condition: 1000 generations.

We examined the 4×4 combinations of the minimum confidence and support. For each combination, all the extracted fuzzy rules were used as candidate rules in the second phase where SOGA searched for the optimal rule set from the candidate rules. In our SOGA, we used biased mutation where a larger probability (i.e., 0.05) was assigned to the mutation from 1 to 0 than that from 0 to 1 for efficiently decreasing the number of fuzzy rules in each rule set. We removed unnecessary fuzzy rules from each string after mutation. That is, we removed all fuzzy rules that were chosen as winner rules for no training patterns. We performed five independent runs of the two-fold cross-validation procedure (i.e., $5 \times 2\text{CV}$).

Experimental results are summarized in Fig. 2. Fig. 2 (a) shows the number of extracted candidate rules. Their classification rates on training and test patterns are shown in Fig. 2 (c) and Fig. 2 (e), respectively. On the other hand, experimental results after genetic rule selection are shown in the right plots of Fig. 2. Only a small number of fuzzy rules were selected in Fig. 2 (b) from thousands of candidate rules in Fig. 2 (a). Training data accuracy was improved in many cases in Fig. 2 (d) by genetic rule selection from Fig. 2 (c). Test data accuracy was also improved in many cases in Fig. 2 (f) from Fig. 2 (e).

From Fig. 2 (c) and Fig. 2 (e), we can see that the accuracy of candidate rules strongly depended on the specification of the minimum confidence and support (see [4] for the learning of these parameter values). When both the minimum confidence and support were small, a larger number of candidate rules were extracted. For example, about 15000 candidate rules were generated in Fig. 2 (a) in the case of the minimum confidence 0.6 and the minimum support 0.01. Among those 15000 candidate rules, only 13 fuzzy rules were selected in Fig. 2 (b) on average. The average classification rates were improved by genetic fuzzy rule selection for both training and test data in Fig. 2. These observations demonstrate the effectiveness of genetic fuzzy rule selection. One difficulty of genetic fuzzy rule selection with a large number of candidate rules is its large computation load. Actually, our SOGA spent about one hour using a PC with Xeon 3.6 GHz CPU in the case of the minimum confidence 0.6 and the minimum support 0.01. The computation load can be significantly decreased by decreasing the number of candidate rules (i.e., by increasing the minimum confidence and support). High classification rates, however, were not obtained when the number of candidate rules was small in Fig. 2. In the next section, we discuss how we can decrease the number of candidate rules without severely degrading the accuracy of selected rules.



(a) Number of candidate rules

(b) Number of selected rules

(c) Training data accuracy of candidate rules

(d) Training data accuracy of selected rules

(e) Test data accuracy of candidate rules

(f) Test data accuracy of selected rules

Fig. 2. Experimental results using SOGA for the glass data

4 Use of Pareto-Optimal and Near Pareto-Optimal Rules

As shown in Fig. 2, good rule sets are not likely to be obtained by genetic fuzzy rule selection when the number of candidate rules is too small. On the other hand, we need a long CPU time when the number of candidate rules is large. In our former study [9], we examined the use of Pareto-optimal fuzzy rules with respect to confidence and support as candidate rules. In this section, we examine the use of not only Pareto-optimal but also near Pareto-optimal fuzzy rules.

Using a dominance margin ε, we modify Pareto dominance as in [18]: A fuzzy rule R_i is said to be ε-dominated by another fuzzy rule R_j when both the following two inequalities hold,

$$c(R_i)+\varepsilon \le c(R_j)\,,\;\; s(R_i)+\varepsilon \le s(R_j)\,, \tag{6}$$

and at least one of the following two inequalities holds:

$$c(R_i)+\varepsilon < c(R_j)\,,\;\; s(R_i)+\varepsilon < s(R_j)\,. \tag{7}$$

When a fuzzy rule R_i is not dominated by any other fuzzy rules in the sense of the ε-dominance in (6) and (7), we call R_i an ε-non-dominated rule. It should be noted that the ε-dominance with $\varepsilon = 0$ is the same as Pareto dominance.

We examined the effect of using ε-non-dominated rules as candidate rules in multiobjective genetic fuzzy rule selection by computational experiments on five data sets from the UCI machine learning repository. The two-fold cross-validation procedure was iterated five times for each data set. First we extracted fuzzy rules using the minimum confidence 0.6 and the minimum support 0.01 (0.04 for the wine data set). Among the extracted fuzzy rules, only ε-non-dominated rules were used as candidate rules. Then we applied NSGA-II [6], [8] to search for non-dominated rule sets from the candidate rules. Finally the accuracy of each of the obtained non-dominated rule sets was calculated for training and test data.

In Table 1, we show the average number of candidate rules for each value of ε. In the case of $\varepsilon = 0$, the number of candidate rules (i.e., Pareto-optimal rules) was very small. On the other hand, it is large when $\varepsilon = \infty$. In this case, all the extracted fuzzy rules were used as candidate rules. By decreasing the value of ε, we can decrease the number of candidate rules as shown in Table 1.

Table 1. Average number of generated candidate fuzzy rules

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	74	28323	35093	46941	57931	78650
Glass	163	4496	6571	11324	14850	15140
Heart C	349	9407	11835	17154	30928	102560
Iris	21	1995	2161	2555	3264	4725
Wine	43	4728	7081	15948	37915	77805

Table 2. Average CPU time of a single run of NSGA-II (minutes)

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	4.9	167.9	185.8	270.0	338.3	453.2
Glass	2.0	11.2	27.9	41.5	59.5	73.3
Heart C	7.8	31.2	48.4	78.6	130.9	266.7
Iris	0.8	4.8	9.2	10.9	10.9	15.2
Wine	3.0	11.5	26.1	60.4	104.9	136.6

The average CPU time for a single run of NSGA-II is shown in Table 2. From Table 1 and Table 2, we can see that the computation load depends on the number of candidate rules. Using a small value of ε, we can decrease the CPU time.

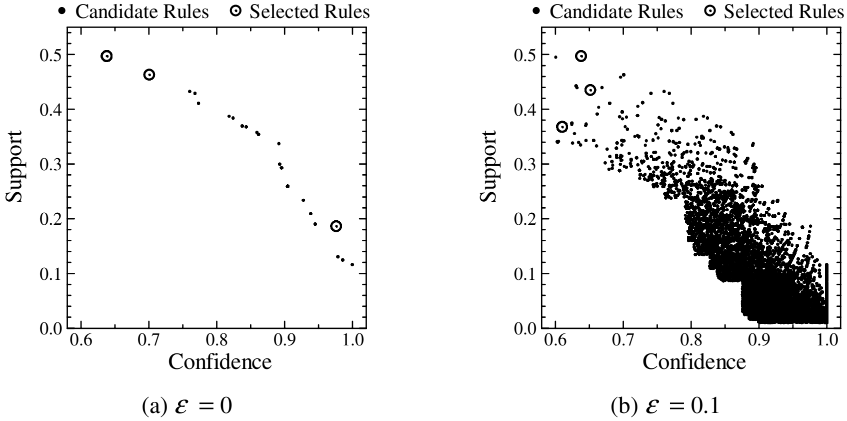
From the viewpoint of CPU time, the use of Pareto-optimal rules as candidate rules (i.e., $\varepsilon = 0$ in Table 2) is a good strategy. The training data accuracy of obtained non-dominated rule sets in this case was not necessarily good as shown in Table 3. In Table 3, we calculated the average value of the highest classification rates on training data among obtained non-dominated rule sets by each run of NSGA-II. Good rule sets were not obtained in the case of $\varepsilon = 0$ for some data sets (e.g., Glass and Heart C). An interesting observation in Table 3 is that the best training data accuracy was not always obtained from a large value of ε. For example, the best training data accuracy

Table 3. Average value of the best classification rates on training data (%)

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	97.4	99.0	99.1	99.1	99.1	99.2
Glass	74.8	79.2	81.7	83.5	84.1	84.0
Heart C	71.8	80.8	79.4	79.7	77.9	78.5
Iris	97.2	97.5	98.8	97.7	97.9	97.9
Wine	100.0	100.0	100.0	100.0	100.0	100.0

Table 4. Average value of the best classification rates on test data (%)

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	96.8	96.7	96.4	96.5	96.8	96.5
Glass	63.4	66.1	64.8	66.1	66.4	65.9
Heart C	54.8	55.4	56.4	56.3	55.2	56.5
Iris	96.6	97.0	97.0	96.5	96.8	96.9
Wine	93.8	92.7	93.3	90.4	94.9	94.9



(a) $\varepsilon = 0$ (b) $\varepsilon = 0.1$

Fig. 3. Candidate rules and selected rules for Class 1 of Heart C

was obtained from $\varepsilon = 0.01$ for Heart C in Table 3. This is because NSGA-II could not find the optimal combination of candidate rules when the number of candidate rules was too large.

In Table 4, we show the average value of the highest classification rates on test data. While the training data accuracy was severely degraded for some data sets by specifying ε as $\varepsilon = 0$, the deterioration in the test data accuracy was not so severe.

For clearly demonstrating the effect of ε, we show candidate rules and selected rules for Class 1 of Heart C by a single-run of NSGA-II in Fig. 3.

5 Conclusions

We proposed an idea of using Pareto-optimal and near Pareto-optimal rules as candidate rules in genetic fuzzy rule selection. Through computational experiments, we demonstrated that the proposed idea decreased the number of candidate rules which were generated based on the minimum confidence and support. As a result, the CPU time for rule selection was decreased. We also demonstrated that rule sets with high training data accuracy were not obtained for some data sets when we used only Pareto-optimal rules. The proposed idea improved the training data accuracy of obtained rule sets by using not only Pareto-optimal but also near Pareto-optimal rules. A future research issue is to examine other definitions (e.g., multiplicative form) of ε-dominance. Different handling of multiobjective problems such as lexicographic ordering [17] is also a future research issue.

Acknowledgement

This work was supported by Grant-in-Aid for Scientific Research on Priority Areas (18049065) and Grant-in-Aid for Scientific Research (B) (17300075).

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I.: Fast Discovery of Association Rules. In: Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park (1996) 307-328
2. Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): *Interpretability Issues in Fuzzy Modeling*. Springer, Berlin (2003)
3. Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): *Accuracy Improvements in Linguistic Fuzzy Modeling*. Springer, Berlin (2003)
4. Coenen, F., Leng, P.: Obtaining Best Parameter Values for Accurate Classification. *Proc. of 5th IEEE International Conference on Data Mining* (2005) 549-552
5. de la Iglesia, B., Philpott, M. S., Bagnall, A. J., Rayward-Smith, V. J.: Data Mining Rules using Multi-Objective Evolutionary Algorithms. *Proc. of 2003 Congress on Evolutionary Computation* (2003) 1552-1559
6. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001)

7. Deb, K., Mohan, M., Mishra, S.: Evaluating the ε-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation 14*, 4 (2005) 501-525
8. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation 6*, 2 (2002) 182-197
9. Ishibuchi, H., Kuwajima, I., Nojima, Y.: Relation between Pareto-Optimal Fuzzy Rules and Pareto-Optimal Fuzzy Rule Sets. *Proc. of 1st IEEE Symposium on Computational Intelligence in Multicriteria Decision Making* (2007) (in press)
10. Ishibuchi, H., Murata, T., Turksen, I. B.: Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems. *Fuzzy Sets and Systems 89*, 2 (1997) 135-150
11. Ishibuchi, H., Nakashima, T., Murata, T.: Three-Objective Genetics-based Machine Learning for Linguistic Rule Extraction. *Information Sciences 136*, 1-4 (2001) 109-133
12. Ishibuchi, H., Nakashima, T., Nii, M.: *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer, Berlin (2004)
13. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms. *IEEE Trans. on Fuzzy Systems 3*, 3 (1995) 260-270
14. Ishibuchi, H., Yamamoto, T.: Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining. *Fuzzy Sets and Systems 141*, 1 (2004) 59-88
15. Kaya, M.: Multi-Objective Genetic Algorithm based Approaches for Mining Optimized Fuzzy Association Rules. *Soft Computing 10* (2006) 578-586
16. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary Computation 10*, 3 (2002) 263-282
17. Luke, S., Panait, L.: Lexicographic Parsimony Pressure, *Proc. of 2002 Genetic and Evolutionary Computation Conference* (2002) 829-836
18. Reynolds, A., de la Iglesia, B.: Rule Induction using Multi-Objective Metaheuristics: Encouraging Rule Diversity. *Proc. of 2006 International Joint Conference on Neural Networks* (2006) 6375-6382
19. Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K. F.: Multi-Objective Hierarchical Genetic Algorithm for Interpretable Fuzzy Rule-based Knowledge Extraction. *Fuzzy Sets and Systems 149*, 1 (2005) 149-186
20. Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K. F.: Agent-based Evolutionary Approach for Interpretable Rule-based Knowledge Extraction. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews 35*, 2 (2005) 143-155

A Dissimilation Particle Swarm Optimization-Based Elman Network and Applications for Identifying and Controlling Ultrasonic Motors

Ge Hong-Wei[1], Du Wen-Li[1], Qian Feng[1,*], and Wang Lu[2]

[1] State-Key Laboratory of Chemical Engineering, China East China University of Science and Technology, Shanghai 200237, China
[2] College of Computer Science, Jilin University, Changchun 130012, China
fqian@ecust.edu.cn

Abstract. In this paper, we first present a learning algorithm for dynamic recurrent Elman neural networks based on a dissimilation particle swarm optimization. The proposed algorithm computes concurrently both the evolution of network structure, weights, initial inputs of the context units and self-feedback coefficient of the modified Elman network. Thereafter, we introduce and discuss a novel control method based on the proposed algorithm. More specifically, a dynamic identifier is constructed to perform speed identification and a controller is designed to perform speed control for Ultrasonic Motors (USM). Numerical experiments show that the novel identifier and controller based on the proposed algorithm can both achieve higher convergence precision and speed. In particular, our experiments show that the identifier can approximate the USM's nonlinear input-output mapping accurately. The effectiveness of the controller is verified using different kinds of speeds of constant and sinusoidal types. Besides, a preliminary examination on a randomly perturbation also shows the robust characteristics of the two proposed models.

1 Introduction

The design goal of a control system is to influence the behavior of dynamic systems to achieve some pre-determinate objectives. A control system is usually designed on the premise that an accurate knowledge of a given object and environment cannot be obtained in advance. It usually requires suitable methods to address the problems related to uncertain and highly complicated dynamic system identification. However, the majority of methods for system identification and parameters' adjustment are based on linear analysis. Normally, a large amount of approximations and simplifications have to be performed and, unavoidably, they have a negative impact on the desired accuracy. Fortunately the characteristics of the artificial neural network (ANN) approach, namely non-linear transformation and support to highly parallel operation, provide effective techniques for system identification and control [1-3].

An Ultrasonic Motor (USM) is a typical non-linear dynamic system. USMs have attracted considerable attention in many practical applications [4,5]. The simulation and control of an USM are crucial in the actual use of such systems. Following a

[*] Corresponding author.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 397–407, 2007.
springerlink.com

conventional control theory approach, an accurate mathematical model should be first set up. But an USM has strongly nonlinear speed characteristics that vary with the driving conditions and its operational characteristics depend on many factors. Therefore, performing effective identification and control in this case, is a challenging task using traditional methods based on mathematical models of the systems. The ANN approach can be applied advantageously to the specific tasks of USM's identification and control since it is capable to tackle nonlinear behaviors and it does not require any system's a prior knowledge. In the following sections, an Elman network is employed to identify and control an USM, and a novel learning algorithm based on a modified particle swarm optimization is proposed for training the Elman network.

2 Modified Elman Network

Figure 1 depicts the modified Elman neural network (ENN) which was proposed by Pham and Liu [6]. It is a type of recurrent neural network with three layers of neurons. In addition to the input nodes, the hidden nodes and the output nodes, there are also context nodes in this model. The context nodes are used only to memorize previous activations of the hidden nodes. The feedforward connections are modifiable, whereas the recurrent connections are fixed. The modified Elman network differs from the original Elman network by having self-feedback links with fixed coefficient α in the context nodes. Thus the output of the context nodes can be described by

$$x_{Cl}(k) = \alpha x_{Cl}(k-1) + x_l(k-1) \quad (l = 1,2,\cdots,n) . \tag{1}$$

where $x_{Cl}(k)$ and $x_l(k)$ are, respectively, the outputs of the lth context unit and the lth hidden unit and α ($0 \le \alpha < 1$) is a self-feedback coefficient. Assume that there are r nodes in the input layer, n nodes in the hidden and context layers, respectively, and m nodes in the output layer. Then the input u is an r dimensional vector, the output x of the hidden layer and the output x_C of the context nodes are n dimensional vectors, the output y of the output layer is m dimensional vector, and the weights W^{I1}, W^{I2} and W^{I3} are $n \times n$, $n \times r$ and $m \times n$ dimensional matrices, respectively.

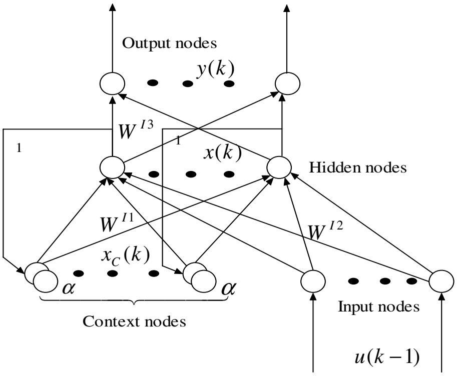


Fig. 1. Architecture of the modified Elman network

The mathematical model of the modified Elman neural network is

$$x(k)=f(W^{I1}x_C(k)+W^{I2}u(k-1))\ , \tag{2}$$

$$x_C(k)=\alpha x_C(k-1)+x(k-1)\ , \tag{3}$$

$$y(k)=g(W^{I3}x(k))\ , \tag{4}$$

where $f(x)$ is often taken as the sigmoidal function, and $g(x)$ is often taken as a linear function. Let the *k*th desired output of the system be $y_d(k)$. Define the error as

$$E(k)=\frac{1}{2}(y_d(k)-y(k))^T(y_d(k)-y(k))\,. \tag{5}$$

Differentiating E with respect to W^{I3}, W^{I2} and W^{I1} respectively, according to the gradient descent method, we obtain the following equations

$$\Delta w_{ij}^{I3}=\eta_3\delta_i^0 x_j(k) \qquad (i=1,2,\cdots,m;\ j=1,2,\cdots,n)\,, \tag{6}$$

$$\Delta w_{jq}^{I2}=\eta_2\delta_j^h u_q(k-1) \qquad (j=1,2,\cdots,n;\ \ q=1,2,\cdots,r)\,, \tag{7}$$

$$\Delta w_{jl}^{I1}=\eta_1\sum_{i=1}^{m}(\delta_i^0 w_{ij}^{I3})\frac{\partial x_j(k)}{\partial w_{jl}^{I1}} \qquad (j=1,2,\cdots,n;\ l=1,2,\cdots,n)\,, \tag{8}$$

which form the learning algorithm for the modified Elman neural network, where η_1, η_2 and η_3 are learning steps of W^{I1}, W^{I2} and W^{I3}, respectively, and

$$\delta_i^0=(y_{d,i}(k)-y_i(k))g_i'(\cdot)\,, \tag{9}$$

$$\delta_j^h=\sum_{i=1}^{m}(\delta_i^0 w_{ij}^{I3})f_j'(\cdot)\,, \tag{10}$$

$$\frac{\partial x_j(k)}{\partial w_{jl}^{I1}}=f_j'(\cdot)x_l(k-1)+\alpha\frac{\partial x_j(k-1)}{\partial w_{jl}^{I1}}\ . \tag{11}$$

Obviously, Eqs. (8) and (11) possess recurrent characteristics.

From the above dynamic equations it can be seen that the output at an arbitrary time is influenced by the past input-output. If a dynamic system is identified or controlled by an Elman network with an artificially imposed structure, and the gradient descent learning algorithm is used, this may give rise to a number of problems:

- the initial input of the context unit is artificially provided, which in turn induces larger errors of system identification or control at the initial stage.
- searches are easy to get locked into local minima.
- the self-feedback coefficient α is determined artificially or experimentally by a lengthy trial-and-error process, which induces a lower learning efficiency.
- if the network structure and weights are not trained concurrently, we have first to determine the number of the nodes of the hidden layer, which induces the network structure and weights may not obey the Kosmogorov theorem and therefore a good performance of dynamic approximation can not be guaranteed.

In order to face the above critical points, we propose a learning algorithm for the ENN based on a dissimilation particle swarm algorithm (DPSO) to enhance the identification capabilities and control performances of the models. Furthermore, a novel identifier and a controller are designed to identify and control USMs.

3 Dissimilation Particle Swarm Optimization (DPSO)

Particle swarm optimization (PSO), originally developed by Kennedy and Elberhart [7]. It is an evolutionary computation technique based on swarm intelligence. A swarm consists of individuals, called "particles", which change their positions over time. In a PSO system, particles fly around in a multi-dimensional search space. During its flight each particle adjusts its position according to its own experience and the experience of its neighboring particles, making use of the best position encountered by itself and its neighbors. The performance of each particle is measured according to a pre-defined fitness function. The mathematical abstract of PSO is as follows.

Let the ith particle in a D-dimensional space be represented as $X_i = (x_{i1},\dots,x_{id},\dots,x_{iD})$. The best previous position of the ith particle is recorded and represented as $P_i = (p_{i1},\dots,p_{id},\dots,p_{iD})$, which is also called $pbest$. The index of the best $pbest$ among all the particles is represented by the symbol g. The location P_g is also called $gbest$. The velocity for the ith particle is represented as $V_i = (v_{i1},\dots,v_{id},\dots,v_{iD})$. The concept of the particle swarm optimization consists of, at each time step, changing the velocity and location of each particle towards its $pbest$ and $gbest$ locations according to Eqs. (12) and (13), respectively:

$$V_i\ (k+1) = wV_i\ (k) + c_1 r_1\ (P_i - X_i(k))/\Delta t + c_2 r_2\ (P_g - X_i(k))/\Delta t \tag{12}$$

$$X_i\ (k+1) = X_i\ (k) + V_i\ (k+1)\Delta t \tag{13}$$

where w is the inertia coefficient which is a constant in interval [0, 1]; c_1 and c_2 are learning rates which are nonnegative constants; r_1 and r_2 are generated randomly in the interval [0, 1]; Δt is the time interval, and commonly be set as unit; $v_{id} \in [-v_{\max}, v_{\max}]$, and $v_{\max}$ is a designated maximum velocity.

The method described above can be considered as the standard particle swarm optimization, in which as time goes on, some particles become quickly inactive because they are similar to the $gbest$ and loose their velocities. In the subsequent generations, they will have less contribution to the search task for their very low global and local search activity. In turn, this will induce the emergence of a state of premature convergence, defined technically as prematurity. To improve on this specific issue, we introduce an adaptive mechanism to enhance the performance of PSO: our modified algorithm is called dissimilation particle swarm optimization (DPSO).

In our proposed algorithm, first the prematurity state of the algorithm is judged against the following conditions after each given generation. Let's define

$$\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f_i, \qquad \sigma_f{}^2 = \frac{1}{n}\sum_{i-1}^{n}(f_i - \bar{f})^2 \tag{14}$$

Where f_i is the fitness value of the i th particle, $\bar{f}$ is the average fitness of all the particles, and ${\sigma_f}^2$ is the variance, which reflects the convergence degree of the population. Moreover, we define the following indicator, τ^2:

$$\tau^2 = \frac{{\sigma_f}^2}{\max\{(f_j - \bar{f})^2, (j = 1, 2, \cdots n)\}} \tag{15}$$

Obviously, $0 < \tau^2 < 1$, if ${\sigma_f}^2 \neq 0$. If τ^2 is less than a small given threshold and the theoretical global optimum or the expectation optimum has not been found, the algorithm is considered to get into a premature convergence state.

In the case, we identify those inactive particles by use of the inequality

$$\frac{f_g - f_i}{\max\{(f_g - f_j), (j = 1, \cdots, n)\}} \leq \theta \tag{16}$$

Where f_g is the fitness of the best particle $gbest$ and θ is a small given threshold decided by the user. The parameter θ is usually taken as $(f_g - \bar{f})/10$ approximately. In this paper, τ^2 and θ are taken as 0.005 and 0.01 respectively.

Finally the inactive particles are chosen to mutate by using a Gauss random disturbance on them according to formula (17).

$$p_{ij} = p_{ij} + \beta_{ij} \qquad (j = 1, \cdots, D) \tag{17}$$

Where p_{ij} is the j th component of the i th inactive particle; β_{ij} is a random variable and follow a Gaussian distribution with zero mean and constant variance 1.

4 DPSO-Based Learning Algorithm of Elman Network

Let's denote the location vector of a particle as X, and its ordered components as self-feedback coefficients, initial inputs of the context unit and all the weights. In the proposed algorithm, a "particle" consists of two parts:

- the first part is named "head", and comprises the self-feedback coefficients;
- the second part is named "body", and includes the initial inputs of the context unit and all the weights.

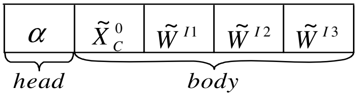


Fig. 2. Architecture of the modified Elman network

As far as the network shown in Figure 1 is concerned, the corresponding "particle" structure can be illustrated in Figure 2. There $\tilde{X}_C^0 = (x_{C,1}^0, \cdots, x_{C,n}^0)$ is a permutation of the initial inputs of the context unit, $\tilde{W}^{I1}$, $\tilde{W}^{I2}$ and $\tilde{W}^{I3}$ are the respective permutations of the expansion of weight matrices W^{I1}, W^{I2} and W^{I3} by rows. Therefore, the number of the elements in the body is $n+n\cdot n+r\cdot n+n\cdot m$. While coding the parameters, we define a lower and an upper bound for each parameter being optimized.

In the DPBEA searching process, two additional operations are introduced: namely, the “structure developing” operation and the “structure degenerating” operation. They realize the evolution of the network structure. Adding or deleting neurons in the hidden layer is judged against the developing probability p_a and the degenerating probability p_d, respectively. If a neuron is added, the weights related to the neuron are added synchronously: such values are randomly set according to their initial range. If the degenerating probability p_d passes the Bernoulli trials, a neuron of the hidden layer is randomly deleted, and the weights related to the neuron are set to zero synchronously. In order to maintain the dimensionality of the particle, the maximal number of the neurons in the hidden layer is given, and taken as 10 in this paper. The evolution of the self-feedback coefficient α in the part of the head lies on the probability p_e. The probabilities p_a , p_d and p_e are given by the following equation

$$p_a=p_d=p_e=e^{-1/N_g\cdot\gamma} \tag{18}$$

Where N_g represents the number of generations that the maximum fitness has not been changed, and is taken as 50; γ is an adjustment coefficient, which is taken as 0.03 in this paper.

The elements in the body part are updated according to Eqs. (12) and (13) in each iteration, while the element α is updated (always using Eqs. (12) and (13)) only if p_e passes the Bernoulli trials.

5 USM Speed Identification Using DPSO-Based Elman Network

In this section, we present and discuss a dynamic identifier to perform the identification of non-linear systems. We named the novel dynamic identifier as DPSO-based ENN Identifier (DPBEI). Numerical simulations have been performed using the model of DPBEI for the speed identification of a longitudinal oscillation USM [8] shown in Figure 3. Some parameters on the USM model in our experiments are taken as follows: driving frequency $27.8\,kHZ$, amplitude of driving voltage $300V$, allowed output moment $2.5kg\cdot cm$, rotation speed $3.8m/s$. The parameters on the DPSO are taken as: population scale 80, learning rates $c_1=1.9$ and $c_2=1.8$ .The Block diagram of the identification model of the motor is shown in Figure 4.

In the simulated experiments, the Elman neural network is trained on line by the DPSO algorithm. The fitness of a particle is evaluated by the reciprocal of the mean square error, namely

$$f_j(k)=\frac{1}{E_j(k)}=p\Big/\sum_{i=k-p+1}^{k}(y_d(i)-y_j(i))^2 \tag{19}$$

where $f_j(k)$ is the fitness value of the particle j at time k , p is the width of the identification window, $y_d(i)$ is the expected output at time i , and $y_j(i)$ is the actual output. The iterations continue until a termination criterion is met, where a sufficiently good fitness value or a predefined maximum number of generations is achieved in the allowed time interval. Upon the identification of a sampling step, the particles produced in the last iteration are stored as an initial population for the next sampling step: only 20% of them are randomly initialized. In fact, these stored particles are good candidate guesses for the solution for the next step, especially if the system is close to the desired steady-state. In our numerical experiments, the use of the described techniques has significantly reduced the number of generations needed to calculate an acceptable solution.

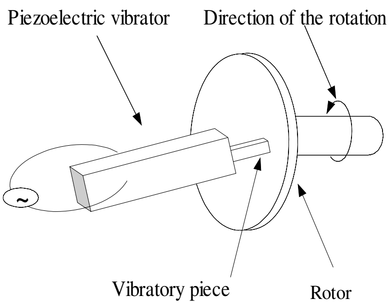


Fig. 3. Schematic diagram of the motor

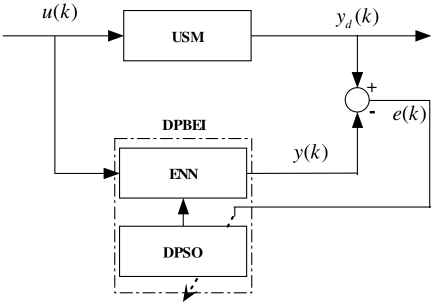
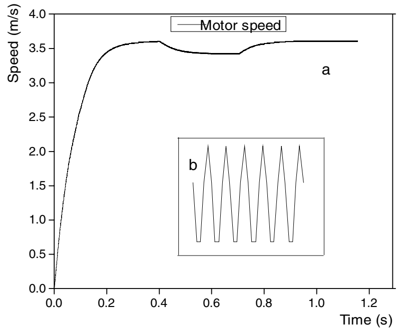


Fig. 4. Block diagram of identification model

In order to show the effectiveness and accuracy of the identification by the proposed method, a durative external moment of $1N \cdot m$ is applied in the time window [0.4, 0.7s] to simulate an external disturbance. The curve of the actual speed is shown as curve *a* in Figure 5, and curve *b* is a zoom of curve *a* at the stabilization stage. Figures 6 to 10 show the respective identification results. The proposed DPBEI model is compared with the original Elman model using the gradient descent-based algorithm.

Figure 6 shows the speed identification curves for the initial stage, with the exclusion of the first 10 sampling data. Figures 7 and 8 respectively show the speed identification curves and error curves for the disturbance stage. Figures 9 and 10 respectively show the speed identification curves and error curves for the stabilization stage. These results demonstrate the full power and potential of the proposed method. If we compare the identification errors obtained with the two methods, we see that in the gradient descent-based learning algorithm they are only less than 0.005, while in the proposed method they are about an order of magnitude smaller, i.e. less than 0.0008. In other words, the identification error of the DPBEI is about 16% that of the Elman model trained by the gradient descent algorithm.

Our simulated experiments of the identification algorithms have been carried out on a PC with Pentium IV 2.8 GHz processor and 512MB memory. There were 21000 sampling data and the whole identification time has been about 6.2 seconds. The average CPU-time for the identification of a sampling data has been about 0.3 msec. The proposed on-line identification model and strategy can be successfully used to identify highly non-linear systems.



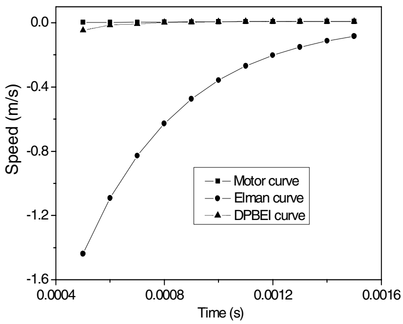
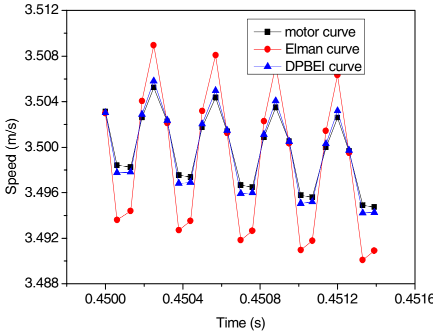


Fig. 5. Actual Speed curve of the USM **Fig. 6.** Speed identification curves at initial stage



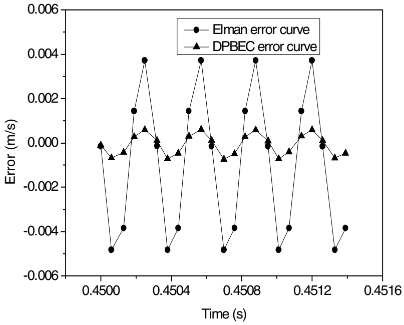
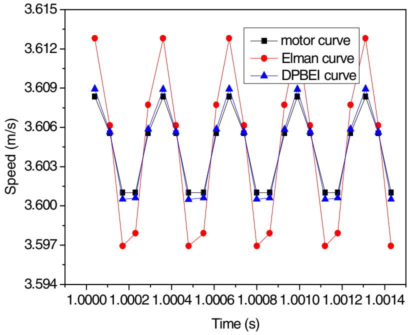


Fig. 7. Identification curves at disturbance stage **Fig. 8.** Error curves at disturbance stage



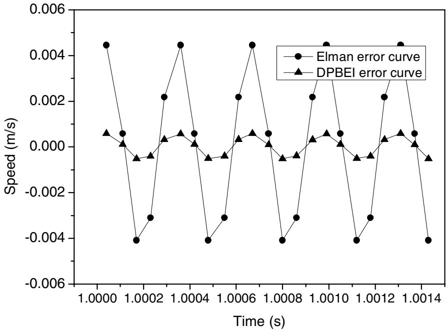


Fig. 9. Identification curves at stabiliztion stage **Fig. 10.** Error curves at stabiliztion stage

6 USM Speed Control Using the DPSO-Based Elman Network

In this section, we present and discuss a novel on-line controller specially designed to control the USM using the DPSO-based Elman network, which we name DPSO-based ENN Controller (DPBEC). The optimized control strategy is illustrated in Figure 11.

In the developed DPBEC, the Elman network is trained on line by the DPSO algorithm, and the driving frequency is taken as the control variable. The fitness of a particle is evaluated computing the deviation of the control result over the expected result from a desired trajectory.

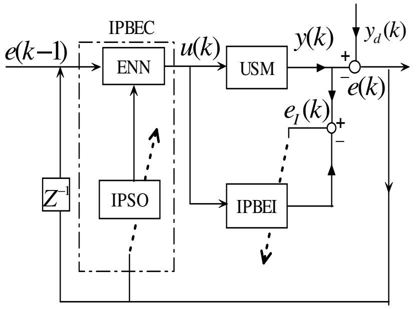


Fig. 11. Block diagram of the control system

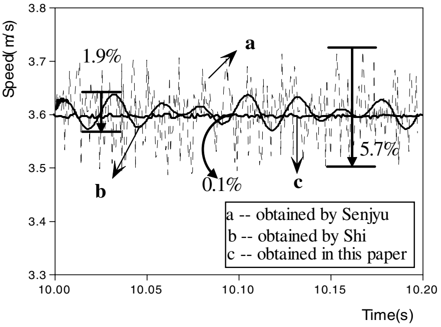


Fig. 12. Comparison of speed control curves

Figure 12 shows the USM speed control curves using three different control strategies when the control speed is taken as $3.6m/s$. The dotted line a represents the speed control curve based on the method presented by Senjyu et al.[9], the solid line b represents the control curve using the method presented by Shi et al.[10] and the solid line c represents the control curve using the proposed method. Simulation results show that the stable speed control curves obtained by the three methods possess different fluctuation behaviors. From Figure 12 it can be seen that the amplitude of the speed fluctuation using the proposed method is significantly smaller at the steady state than the other methods. The fluctuation degree is defined as

$$\zeta = (V_{\max} - V_{\min}) / V_{ave} \times 100\% \tag{20}$$

where $V_{\max}, V_{\min}$ and V_{ave} represent the maximum, minimum and average values of the speeds. As reported in Figure 12, the maximum fluctuation values when using the methods proposed by Senjyu and Shi are 5.7% and 1.9% respectively, whereas they are reduced significantly to 0.1% when using the method proposed in this paper.

The speed control curves of the referenced values varying with time are also examined to further verify the control effectiveness of the novel method. Figure 13 shows the speed control curves, where the reference speed follows a sinusoidal behavior.

For the sake of verifying preliminarily the robustness of the proposed control system, we examine the response of the system when an instantaneous perturbation is added into the control system. Figure 14 shows the speed control curve when the driving frequency is subject to an instantaneous perturbation (5% of the driving frequency value) at time = 6 seconds. From the figure it can be seen that the control model possesses a rapid adaptive behavior against the randomly instantaneous perturbation on the frequency of the driving voltage.

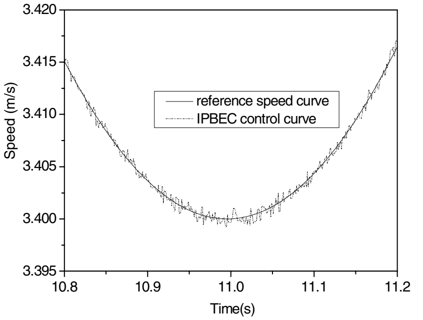


Fig. 13. Control curves with varied speeds

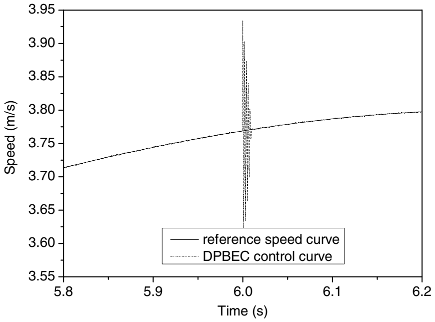


Fig. 14. Speed control for instantaneous disturbance

7 Conclusions

The proposed learning algorithm for Elman neural networks based on the modified PSO overcomes some known shortcoming of ordinary gradient descent methods, namely (1) their sensitivity to the selection of initial values and (2) their propensity to lock into a local extreme point. Moreover, training dynamic neural networks by DPSO does not need to calculate the dynamic derivatives of weights, which reduces significantly the calculation complexity of the algorithm. Besides, the speed of convergence is not dependent on the dimension of the identified and controlled system, but is only dependent on the model of neural networks and the adopted learning algorithm. In this paper, we have described, analyzed and discussed an identifier DPBEI and a controller DPBEC designed to identify and control ultrasonic motor on line. When the system is disturbed by an external noise, it can learn on line and adapt in real-time to the nonlinearity and uncertainty. Our numerical experiments show that the designed identifier and controller can achieve both higher convergence precision and speed. Besides, the preliminary examination on a random perturbation also shows the robust characteristics of the two models. The methods described in this paper can provide effective approaches for non-linear dynamic systems identification and control.

Acknowledgment

The first three authors are grateful to the support of the National “973” Plan (2002CB3122000), the National “863” Plan (20060104Z1081), the National Science Fund for Distinguished Young Scholars (60625302), the Major State Basic Research Development Program of Shanghai (05DJ14002), and the Natural Science Foundation of Shanghai (05ZR14038).

References

1. Hayakawa T., Haddad W.M., Bailey J.M. and Hovakimyan N.: Passivity-based neural network adaptive output feedback control for nonlinear nonnegative dynamical systems. IEEE Transactions on Neural Networks, (2005), 16(2): 387-398.
2. Sunar M., Gurain A.M.A. and Mohandes M.: Substructural neural network controller. Computers & Structures, (2000), 78(4): 575-581.

3. Wang D. and Huang J.: Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. IEEE Transactions on Neural Networks, (2005), 16(1): 195-202.
4. Sashida T. and Kenjo T.: An Introduction to Ultrasonic Motors. Oxford: Clarendon Press, (1993).
5. Hemsel T. and Wallaschek J.: Survey of the present state of the art of piezoelectric linear motors. Ultrasonics, (2000), 38(1-8): 37–40.
6. Pham D.T. and Liu X.: Dynamic system modeling using partially recurrent neural networks. J. of Systems Engineering, (1992), 2: 90-97.
7. Kennedy J. and Eberhart R.: Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press. Perth, Australia, (1995), 4: 1942–1948.
8. Xu X., Liang Y.C., Lee H.P., Lin W.Z., Lim S.P. and Lee K.H.: Mechanical modeling of a longitudinal oscillation ultrasonic motor and temperature effect analysis. Smart Materials and Structures, (2003), 12(4): 514-523.
9. Senjyu T., Miyazato H., Yokoda S. and Uezato K.: Speed control of ultrasonic motors using neural network. IEEE Transactions on Power Electronics, (1998), 13(3): 381-387.
10. Shi X.H., Liang Y.C., Lee H.P., Lin W.Z., Xu X. and Lim S.P.: Improved Elman networks and applications for controlling ultrasonic motors. Applied Artificial Intelligence, (2004), 18(7): 603-629.

A Cultural Algorithm for Solving the Set Covering Problem

Broderick Crawford[1,2], Carolina Lagos[1], Carlos Castro[2], and Fernando Paredes[3]

[1] Pontificia Universidad Católica de Valparaíso, PUCV, Chile
[2] Universidad Técnica Federico Santa María, Valparaíso, Chile
[3] Universidad Diego Portales, Santiago, Chile

Abstract. In this paper we apply a new evolutive approach for solving the Set Covering Problem. This problem is a reasonably well known NP-complete optimization problem with many real world applications. We use a Cultural Evolutionary Architecture to maintain knowledge of Diversity and Fitness learned over each generation during the search process. Our results indicate that the approach is able to produce competitive results in compare with other approximation algorithms solving a portfolio of test problems taken from the ORLIB.

Keywords: Set Covering Problem, Cultural Algorithms, Genetic and Evolutionary Computation.

1 Introduction

Set partitioning problem (SPP) and set covering problem (SCP) are two types of problems that can model several real life situations [6, 3]. In this work, we solve some benchmarks of SCP with a new evolutive approach: cultural algorithms [12, 13, 14].

Cultural algorithms are a technique that incorporates knowledge obtained during the evolutionary process trying to make the search process more efficient. Cultural algorithms have been successfully applied to several types of optimization problems [4, 9]. However, nobody had proposed a cultural algorithm for SCP.

This paper is organized as follows: In Section 2, we formally describe SCP using mathematical programming models. In section 3 we present the cultural evolutionary architecture. In sections 4 y 5 we show the population space and the belief space considered to solve SCP with cultural algorithms maintaining diversity and fitness knowledge. In Section 6, we present experimental results obtained when applying the algorithm for solving some standard benchmarks taken from the ORLIB [1]. Finally, in Section 7 we conclude the paper and give some perspectives for future research.

2 Problem Description

SPP is the NP-complete problem of partitioning a given set into mutually independent subsets while minimizing a cost function defined as the sum of the costs associated to each of the eligible subsets [3]. In the SPP linear programming formulation we are

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 408–415, 2007.
springerlink.com

given a m rows and n columns incidence matrix $A = (a_{ij})$ in which all the matrix elements are either zero or one. Additionally, each column is given a non-negative real cost c_j. We say that a column j can cover a row i if $a_{ij} = 1$. Let J denotes the set of the columns and x_j a binary variable which is one if column j is chosen and zero otherwise. The SPP can be defined formally as follows:

$$Minimize \quad f(x) = \sum_{j=1}^{n} c_j x_j \tag{2.1}$$

$$subject \ \ to \sum_{j=1}^{n} a_{ij} x_j = 1 \qquad i = 1,...,m. \tag{2.2}$$

$$x_j \in \{0,1\} \qquad j = 1,...,n \tag{2.3}$$

These constraints enforce that each row is covered by exactly one column. The SPP has been studied extensively over the years because of its many real world applications. The SCP is a SPP relaxation (replacing the equation 2.2 by 2.4). The goal in the SCP is to choose a subset of the columns of minimal weight which covers every row. The SCP can be defined formally using constraints to enforce that each row is covered by at least one column as follows:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i = 1,...,m. \tag{2.4}$$

3 Cultural Algorithm

The cultural algorithms were developed by Robert G. Reynolds [12–14] as a complement to the metaphor used by evolutionary algorithms that are mainly focused on natural selection and genetic concepts. The cultural algorithms are based on some theories which try to model cultural as an inheritance process operating at two levels: a *micro-evolutionary level*, which consists of the genetic material that an offspring inherits from its parent, and a *macro-evolutionary level*, which is the knowledge acquired by the individuals through generations. This knowledge, once encoded and stored, it serves to guide the behavior of the individuals that belong to a population.

Considering that evolution can be seen like an optimization process, Reynolds developed a computational model of cultural evolution that can have applications in optimization [4, 9]. He considered the phenomenon of double inheritance with the purpose of increase the learning or convergence rates of an evolutionary algorithm. In this model each one of the levels is represented by a space. The micro-evolutionary level is represented by the population space and the macro-evolutionary level by the belief space.

The population space can be adopted by anyone of the paradigms of evolutionary computation, such as the genetic algorithms, the evolutionary strategies or the evolutionary programming. In all of them there is a set of individuals where each one has a set of independent characteristics with which it is possible to determine its fitness. Through time, such individuals could be replaced by some of their descendants, obtained from a set of operators (crossover and mutation, for example) applied to the population.

The belief space is the "store of the knowledge" acquired by the individuals along the generations. The information in this space must be available for the population of individuals. There is a protocol of communication established to dictate rules about the type of information that it is necessary to interchange between the spaces. This protocol defines two functions: *acceptance*, this function extracts the information (or experience) from the individuals of a generation putting it into the belief space; and *Influence*, this function is in charge "to influence" in the selection and the variation operators of the individuals (as the crossover and mutation in the case of the genetic algorithms). This means that this function exerts a type of pressure according to the information stored in the belief space. Such interactions can be appreciated in Figure 1 [13, 8].

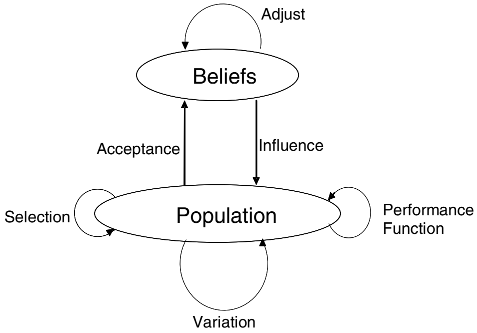


Fig. 1. Spaces of a Cultural Algorithm

3.1 Types of Knowledge

In the belief space of a cultural evolution model there are different types of knowledge: Situational, Normative, Topographic, Historical or Temporal, and Domain Knowledge. According to Reynolds and Peng [14, 11] these types conform a complete set, that is any other type of knowledge can be generated by means of a combination of two or more of the previous types of knowledge.

The pseudo-code of a cultural algorithm is shown in Figure 2 [8]. Most of the steps of a cultural algorithm correspond with the steps of a traditional evolutionary algorithm. It can be clearly seen that the main difference lies in the fact that cultural algorithm use a belief space. In the main loop of the algorithm, we have the update of the belief space, at this point the belief space incorporates the individual experiences of a select group of members with the acceptance function, which is applied to the entire population.

```
Generate the initial population (Population Space)
Initialize the Belief Space
Evaluate the initial population
Repeat
    Update the Belief Space (Acceptance function)
    Apply the variation operators (considering Influence function)
    Evaluate each child
    Perform selection
Until the end condition is not satisfied
```

Fig. 2. Pseudo-code of a Cultural Algorithm

4 Population Space

In the design and development of our cultural algorithm solving SCP we considered in the population space a genetic algorithm with binary representation. An individual, solution or chromosome is an n-bit string, where a value 1 in the bit indicates that the column is considered in the solution and zero in another case (value in j-bit corresponds to value of x_j in the linear programming model). The initial population was generated with n selected individuals randomly with a repair process in order to assure the feasibility of the individuals. For the selection of parents we used binary tournament and the method of the roulette. For the process of variation we used the operator of basic crossover and the fusion operator proposed by Beasley and Chu [2], for mutation we used interchange and multibit. For the treatment of unfeasible individuals we applied the repairing heuristic proposed by Beasley and Chu too. In the replacement of individuals we use the strategy steady state and the heuristic proposed by Lozano et al. [10], which is based on the level of diversity contribution of the new offspring. The genetic diversity was calculated by the Hamming distance, which is defined as the number of bit differences between two solutions. The fitness function is determined for:

$$f_i = \sum_{j=1}^{n} c_j s_{ij} \tag{4.1}$$

Where S is the set of columns in the solutions, s_{ij} is the value of bit (or column) j in the string corresponding to individual i and c_j is the cost of the bit.

The main idea is try to replace a solution with worse fitness and with lower contribution of diversity than the one provided by the offspring. In this way, we are working with two underlying objectives simultaneously: to optimize the fitness and to promote useful diversity.

5 Belief Space

In a cultural algorithm, the shared belief space is the foundation supporting the efficiency of the search process. In order to find better solutions and improve the convergence speed we incorporated information about the diversity in the belief space. We stored in the belief space the individual with better fitness of the current generation and the individual who delivers major diversity to the population, which will be considered leaders in the space of beliefs. With this type of knowledge situational, each of the new individuals generated try to follow a leader stored in the space of beliefs.

5.1 Initialize the Belief Space

A situational-fitness knowledge procedure selects from the initial population the individual with better fitness, which will be a leader in the situational-fitness space of beliefs. A situational-diverse knowledge procedure selects from the initial population the most diverse individual of the population, which will be a leader in the situational-diverse space of beliefs.

5.2 Apply the Variation Operators

In this work, we implemented the influence of situational-fitness knowledge in the operator of crossover. The influence initially appears at the moment of the parental selection, the first father will be chosen with the method of binary tournament and the second father will be the individual with better fitness stored in the space of beliefs. In relation with the influence of situational-diverse knowledge in the operator of crossover, this procedure works recombining the individual with better fitness of every generation with the most diverse stored in the space of beliefs, with this option we expect to deliver diversity to the population.

5.3 Update the Belief Space

The updating the situational belief space procedure implies that the situational space of beliefs will be updated in all generations of the evolutionary process. The update of the situational space of beliefs consists in the replacement of the individuals by current generation individuals if they are better considering fitness and diversity.

6 Experiments and Results

The following tables present the results (cost obtained) when applying different operators in our algorithm for solving SCP benchmarks from ORLIB [1]. The first two columns present the problem code and the best known solution for each instance. The next column shows the cost from a Genetic Algorithm using the basic proposal described in 4 not considering diversity. The next column presents the best cost obtained when applying our Cultural Algorithm. The following columns show the results applying Ant System (AS) and Ant Colony System (ACS) taken from [5] and Round, Dual-LP, Primal-Dual, Greedy taken from [7]. The algorithm has been run with the following parameters setting: size of the population (n) =100, size of the tournament (t) =3, number of generations (g) =30, probability of crossing (pc) =0.6 and probability of mutation (pm) =0.2. The algorithm was implemented using ANSI C, GCC 3.3.6, under Microsoft Windows XP Professional version 2002.

Table 1. Cost obtained with different algorithms. Genetic and Cultural Algorithms using Fusion Crossover and Mutation Interchange.

Problem	**Opt.**	**GA**	**CA**	**AS**	**ACS**	**Round**	**Dual-LP**	**Primal-Dual**	**Greedy**
SCP41	429	506	462	473	463	429	505	521	463
SCP42	512	609	582	594	590	*	*	*	*
SCP48	492	560	549	524	522	*	*	*	*
SCP51	253	298	296	289	280	405	324	334	293
SCP61	138	172	156	157	154	301	210	204	155
SCP62	146	162	162	169	163	347	209	232	170
SCP63	145	170	164	161	157	*	*	*	*
SCPa1	254	319	263	*	*	592	331	348	288
SCPb1	69	102	95	*	*	196	115	101	75
SCPc1	227	260	260	*	*	592	331	348	288

Table 2. Cost obtained with different algorithms. Genetic and Cultural Algorithms using Fusion Crossover and Mutation MultiBit.

Problem	**Opt.**	**GA**	**CA**	**AS**	**ACS**	**Round**	**Dual-LP**	**Primal-Dual**	**Greedy**
SCP41	429	448	448	473	463	429	505	521	463
SCP42	512	642	603	594	590	*	*	*	*
SCP48	492	609	548	524	522	*	*	*	*
SCP51	253	380	309	289	280	405	324	334	293
SCP61	138	162	155	157	154	301	210	204	155
SCP62	146	188	171	169	163	347	209	232	170
SCP63	145	178	176	161	157	*	*	*	*
SCPa1	254	253	303	*	*	592	331	348	288
SCPb1	69	101	87	*	*	196	115	101	75
SCPc1	227	308	254	*	*	592	331	348	288

Table 3. Cost obtained with different algorithms. Genetic and Cultural Algorithms using Basic Cossover and Mutation MultiBit.

Problem	**Opt.**	**GA**	**CA**	**AS**	**ACS**	**Round**	**Dual-LP**	**Primal-Dual**	**Greedy**
SCP41	429	513	460	473	463	429	505	521	463
SCP42	512	668	635	594	590	*	*	*	*
SCP48	492	598	562	524	522	*	*	*	*
SCP51	253	349	299	289	280	405	324	334	293
SCP61	138	164	161	157	154	301	210	204	155
SCP62	146	181	161	169	163	347	209	232	170
SCP63	145	198	193	161	157	*	*	*	*
SCPa1	254	363	269	*	*	592	331	348	288
SCPb1	69	107	95	*	*	196	115	101	75
SCPc1	227	329	260	*	*	592	331	348	288

Table 4. Cost obtained with different algorithms. Genetic and Cultural Algorithms using Basic Crossover and Mutation Interchange.

Problem	Opt.	GA	CA	AS	ACS	Round	Dual-LP	Primal-Dual	Greedy
SCP41	429	519	446	473	463	429	505	521	463
SCP42	512	627	569	594	590	*	*	*	*
SCP48	492	642	636	524	522	*	*	*	*
SCP51	253	303	295	289	280	405	324	334	293
SCP61	138	179	151	157	154	301	210	204	155
SCP62	146	183	162	169	163	347	209	232	170
SCP63	145	198	196	161	157	*	*	*	*
SCPa1	254	426	300	*	*	592	331	348	288
SCPb1	69	97	95	*	*	196	115	101	75
SCPc1	227	238	277	*	*	592	331	348	288

7 Conclusions

In this paper we have introduced the first proposal to solve SCP with cultural algorithms. The main idea of the cultural algorithms is to incorporate knowledge acquired during the search process, integrating inside of an evolutionary algorithm the so called belief space. The objective is to do more robust algorithms with greater rates of convergence. Our computational results confirm that incorporating information about the diversity of solutions we can obtain good results in the majority of the experiments. Our main conclusion from this work is that we can improve the performance of genetic algorithms considering additional information in the evolutionary process. Genetic algorithms tends to lose diversity very quickly, in order to deal with this problem, we have shown that maintaining diversity in the belief space we can improve the computational efficiency. Evolutionary algorithms can be seen as two phase process. The first phase explores for good solutions, while the second phase exploits the better solutions. Both phases appear to require different diversity. Then, in order to improve further the performance of our algorithm, we are now exploring the possibilities for employing knowledge about diversity from the belief space only in the first generations of the population space. In this way, considering that population diversity and selective pressure (that gives individuals with higher fitness a higher chance of being selected for reproduction) are inversely related, we also plan to work on a nondeterministic use of the two communication functions (accept and influence). The main idea behind this approach is an adaptive solving way that it evaluates the tradeoffs between exploration and explotation during the process.

References

1. J. E. Beasley. Or-library: distributing test problem by electronic mail. Journal of Operational Research Society, 41(11):1069–1072, 1990. Available at http://people.brunel.ac.uk/mastjjb/jeb/info.html.
2. J. E. Beasley. and P. C. Chu. A genetic algorithm for the set covering problem. European Journal of Operational Research, 94(2):392–404, 1996.

3. P. C. Chu. and J. E. Beasley. Constraint handling in genetic algorithms: The set partitioning problem. Journal of Heuristics, 4(4):323–357, 1998.
4. C. A. Coello. and R. Landa. Constrained Optimization Using an Evolutionary Programming-Based Cultural Algorithm. In I. Parmee, editor, Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002), volume 5, pages 317–328, University of Exeter, Devon, UK, April 2002. Springer-Verlag.
5. B. Crawford. and C. Castro. Integrating lookahead and post processing proceures with aco for solving set partitioning and covering problems. In L. Rutkowski, R. Tadeusiewicz. L. A. Zadeh, and J. Zurada, editors, ICAISC, volume 4029 of Lecture Notes in Computer Science, pages 1082–1090. Springer, 2006.
6. T. Feo and M. Resende. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, 8:67–71, 1989.
7. F. C. Gomes, C. N. Meneses, P. M. Pardalos, and G. V. R. Viana. Experimental analysis of approximation algorithms for the vertex cover and set covering problems. Comput. Oper. Res., 33(12):3520–3534, 2006.
8. R. Landa. and C. A. Coello. Optimization with constraints using a cultured differential evolution approach. In GECCO ’05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pages 27–34, New York, NY, USA, 2005. ACM Press.
9. R. Landa. and C. A. Coello. Use of domain information to improve the performance of an evolutionary algorithm. In GECCO ’05: Proceedings of the 2005 workshops on Genetic and evolutionary computation, pages 362–365, New York, NY, USA, 2005. ACM Press.
10. M. Lozano. F. Herrera. and J. R. Cano. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. In Proceedings of the 8th Online World Conference on Soft Computing in Industrial Applications, September 2003.
11. B. Peng. Knowledge and population swarms in cultural algorithms for dynamic environments. PhD thesis, Detroit, MI, USA, 2005. Adviser-Robert G. Reynolds.
12. R. Reynolds. An introduction to cultural algorithms. In Third Annual Conference on Evolutionary Programming, pages 131–139, 1994.
13. R. G. Reynolds. Cultural algorithms: theory and applications. In New ideas in optimization, pages 367–378, Maidenhead, UK, England, 1999. McGraw-Hill Ltd., UK.
14. R. G. Reynolds. and B. Peng. Cultural algorithms: Modeling of how cultures learn to solve problems. In ICTAI, pages 166–172. IEEE Computer Society, 2004.

Integration of Production and Distribution Planning Using a Genetic Algorithm in Supply Chain Management

Byung Joo Park, Hyung Rim Choi*, and Moo Hong Kang

Departement of Management Information Systems, Dong-A University
840 Hadan-dong, Saha-gu, Busan 604-714, Korea
{a967500, hrchoi, mongy}@dau.ac.kr

Abstract. In SCM (supply chain management), planning and operation includes the mutually related partial problems. These problems should be dealt with integrated manner on the whole aspect. The core partial problems in the SCM are production planning and distribution planning. As these problems are mutually related, they should be dealt with simultaneously in an integrated manner. This study develops a genetic algorithm that can integrate the production planning and distribution planning in the SCM.

Keywords: Supply chain management, Genetic algorithm, Production planning, Distribution planning.

1 Introduction

SCM (supply chain management) is a business strategy to raise the efficiency of a supply chain by integrating and managing the flows of materials/goods and information ranging from raw materials to customers. The problems related to the planning and operation of an SCM are correlated in the parts of supply, production, distribution, and inventory in the SCM. To solve these problems, the need for a new approach, which can integrate and manage all those problems from an overall perspective, becomes greater. In other words, the approach is not to optimize the problems in decision making in various sectors in sequence, but to simultaneously optimize all problems by using an integrated model. As a typical study related to integrated management in SCM, Thomas & Griffin [9] categorized a supply chain into several integrated models by operation; that is, into integrated models of supplier/buyer, production/distribution, and inventory/distribution, and explained each model in detail. However, integrated planning of all functions in supply chain is a tough task in reality. The reason is that those functions have different goals within mutually complex relationships, and conflicting relationships exist between goals and cost. Studies on the partial integration of those functions have recently been carried out. Partial integration is conducted with production/distribution, production/

* Corresponding author.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 416–426, 2007.
springerlink.com

inventory/distribution, supply/purchase, supply/production, and inventory/ distribution. The core problems in SCM are production and distribution planning. The problem of production planning is to decide which manufacturer will produce the ordered goods, and when, and how many/much to meet customer needs. The problem of distribution planning is to find a channel to deliver goods from a manufacturer to a distributor or a customer. These problems are mutually dependent, and therefore, they should be simultaneously handled in an integrated manner.

MIP (mixed integer programming) and a simulation technique were often used for the integration of production/distribution planning in SCM [3, 5]. However, when the size of problems becomes bigger, and more constraints exist, it is impossible to obtain an optimal solution using a mathematical model. Accordingly, a multi-level technique was developed, based on a mathematical model or a heuristic technique in each stage of the supply chain. Meanwhile, the simulation technique can describe various real and detailed situations, but it is difficult to search for an optimal value; as a result, the simulation technique takes much time and effort to analyze results. Byrne & Bakir [1] studied a hybrid algorithm, which mixed a mathematical model and a simulation model to solve the integration problem of production and distribution planning of multiple products and periods. They demonstrated the usefulness of the hybrid method, and also showed that the solutions obtained from an analytic model could not be accepted in the real world. Kim & Kim [6] expanded the idea of Byrne & Bakir [1] for adequate production planning by using a mathematical model and a simulation model. Erenguc et al. [4] discussed production/distribution planning to check the matters to be considered for optimization of production/distribution, when supplier, a factory, and distribution shops are separated. They combined an analytical model and a simulation model to integrate all the stages in the supply chain. Some researcher applied GA (genetic algorithm) for integrated planning in SCM. Syarif et al. [8] presented a spanning tree-based GA to select the manufacturer, distributor, and distribution network that can meet customer’s needs with minimum cost.

This study develops a GA that can solve the integration problem of production and distribution in an SCM.

2 Integration of Production and Distribution Planning in SCM

2.1 Definition of the Integration Problem

In SCM of this study, raw materials are supplied from multiple suppliers to multiple manufacturers, and various products produced by each manufacturer are supplied to various distributors. Here, a distributor can be understood as a logistics warehouse delivering finished goods from a manufacturer to a retailer or customer. We assume that each manufacturer can produce various products, and can produce all the products ordered within each period. Suppliers are also assumed to be able to supply raw materials or parts in each period. Many distributors exist in integrated planning of the SCM. These distributors receive orders from many customers, and collect those orders and give orders to manufacturers. In this study, demand in the concerned periods is assumed to be known for several periods in the near future. Each distributor can control inventory cost and cost of orders through adjustment of order quantity, and can be supplied products with minimum transport cost and production cost, due to selection of an adequate manufacturer. Manufacturers can control production preparation cost and inventory cost through adjustment of production output at each period, and can be supplied materials and parts at minimum prices and transport cost with a selection of an adequate supplier. The objective function in the integrated planning problem is to minimize total cost of the cost of order, inventory cost, production preparation cost, production cost of manufacturers, and the transport cost between manufacturers and distributors and between suppliers and manufacturers.

2.2 Mathematical Model for Integrated Planning in SCM

Indices

i : Number of suppliers (i = 1, 2, ..., I), j: Number of manufacturers (j = 1, 2, ..., J)
p: Number of distributors (p = 1, 2, ..., P), q: Number of products (q = 1, 2, ..., Q)
t : Number of period (t = 1, 2, ..., T)

Parameters

DE_{pqt} : Demand of product q from distributor p in period t
CO_{jq} : Production cost per unit of product q at manufacturer j
SE_{jq} : Production preparation cost for product q at manufacturer j
OR_{pq} : Cost of order for product q at distributor p
IC_{jq} : Inventory cost per unit of product q at manufacturer j
TL_{jpq} : Transport cost per unit of product q from manufacturer j to distributor p
ST_{ijq} : Supply cost per unit of raw material of product q from supplier i to manufacturer j
IL_{pq} : Inventory cost per unit of product q at distributor p

Variables

N_{jqt} : Production output for product q of manufacturer j in period t
Q_{pqt} : Quantity of order for product q of distributor p in period t
M_{jpqt} : Quantity of supply for product q from manufacturer j to distributor p in period t
I_{jqt} : Inventory for product q of manufacturer j in period t
R_{pqt} : Inventory for product q of distributor q in period t
S_{ijqt} : Quantity of supply of raw material for product q from supplier i to manufacturer j in period t
W_{jqt} : If product q is produced at manufacturer j in period t, 1 otherwise, 0
O_{pqt} : If product q is ordered from distributor p in period t, 1 otherwise, 0

$$\textit{Minimize} \quad TC = \sum_{j=1}^{J}\sum_{q=1}^{Q}\sum_{t=1}^{T} CO_{jq} \cdot N_{jqt} + \sum_{j=1}^{J}\sum_{q=1}^{Q}\sum_{t=1}^{T} SE_{jq} \cdot W_{jqt} + \sum_{j=1}^{J}\sum_{q=1}^{Q}\sum_{t=1}^{T} IC_{jq} \cdot I_{jqt}$$
$$+ \sum_{j=1}^{J}\sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{t=1}^{T} TL_{jpq} \cdot M_{jpqt} + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{q=1}^{Q}\sum_{t=1}^{T} ST_{ijq} \cdot S_{ijqt}$$
$$+ \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{t=1}^{T} IL_{pq} \cdot R_{pqt} + \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{t=1}^{T} OR_{pq} \cdot O_{pqt} \qquad (1)$$

Subject to

$$Q_{pqt} - R_{pqt} + R_{pq,t-1} = DE_{pqt} \qquad \text{for all p, q, t} \tag{2}$$

$$\sum_{j=1}^{J} M_{jpqt} - Q_{pqt} = 0 \qquad \text{for all p, q, t} \tag{3}$$

$$\sum_{p=1}^{P} M_{jpqt} + I_{jqt} - I_{jq,t-1} - N_{jqt} = 0 \quad \text{for all j, q, t} \tag{4}$$

$$\sum_{i=1}^{I} S_{ijqt} - N_{jqt} = 0 \qquad \text{for all j, q, t} \tag{5}$$

$$N_{jqt} - M \cdot N_{jq,t-1} - M \cdot W_{jqt} \le 0 \quad \text{for all j, q, t} \tag{6}$$

$$Q_{pqt} - M \cdot O_{pqt} \le 0 \qquad \text{for all p, q, t} \tag{7}$$

$$W_{jqt},\ O_{pqt} = \{0, 1\} \qquad \text{for all j, q, t} \tag{8}$$

$$M_{jpqt},\ I_{iqt},\ N_{jqt},\ S_{ijqt},\ Q_{pqt},\ R_{pqt} \ge 0 \quad \text{for all i, j, p, q, t} \tag{9}$$

$$M : Big\ Number,\ \ R_{pq,0} = 0,\ \ I_{jq,0} = 0$$

Formula (1) minimizes the total cost of cost of order, production cost, production preparation cost, inventory cost in a manufacturer and a distributor, and transport cost from a distributor to a manufacturer, and from a manufacturer to a distributor. In (2), the demand of a distributor is met by quantity of orders and inventory in the distributor. (3) demonstrates that the quantity of supply from a manufacturer to a distributor is the same as the quantity of orders of distributor. (4) demonstrates that the quantity of supply from a manufacturer to a distributor must be met by production output and inventory in the manufacturer. In (5), production output of each manufacturer equals to the total supply quantity from suppliers. (6) and (7) are related to production preparation cost and cost of order. When a product which was not produced in the previous period is produced in the current period, production preparation cost is generated, and cost of order is generated for every order. (8) is the condition of the binary integer, and (9) shows non-negative condition.

3 Design of Genetic Algorithm

GA are the most well-known and robust methods. GA deals with coding of the problem using chromosomes as representative elements, instead of decision making valuables handled in an exact method. In the integration problem of production and distribution planning handled in this study with one chromosome, the chromosome has to represent all the components in the process of production and distribution planning, and then the pattern of production and distribution planning should be able to be decided.

3.1 Representation of a Chromosome

The most important stage in designing a GA is to represent solution with chromosome. A chromosome should be able to reflect the features of a problem and represent them properly, and produce a more suitable solution for the objective function, through an evolutionary process, by a genetic operator. This study uses the chromosome representation suitable for the integrated planning of suppliers, manufacturers and distributors. When the demand information for each distributor in each period is collected, chromosome representation needs to be conducted to decide the period and manufacturer to order. In the chromosome composed of four rows in Fig. 1, the first row is the representation of the chromosome to decide a distributor's order quantity. The second row is the representation to decide a manufacturer. The third row is to decide production output of a manufacturer at each period and the fourth row is to select a supplier to supply a raw material.

1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1
1	1	2	1	2	2	1	2	2	1	2	1	1	2	1	2	2	1	1	1
1	1	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0
1	2	1	2	2	1	1	2	1	2	2	2	1	1	2	1	2	2	1	1

Fig. 1. Representation of a chromosome

To determine order quantity at each period with regard to each distributor, the binary representation is used as presented by Dellaert [2]. And in order to set the manufacturer to produce the ordered product, values within the number of eligible manufacturers need to be randomly generated. If a specific manufacturer is not able to produce a specific product, the GA has been designed to generate random number for manufacturers within the number of manufacturers in which the concerned manufacturer was excluded. After the manufacturers are decided, each manufacturer decides its production period. To decide this, binary representation is also used. Through this process, when the production period and output are decided for each manufacturer, suppliers who will supply the required raw materials should be selected. For the supplier selection, the values within the number of possible suppliers are randomly generated as shown in the decision of manufacturers above. When the chromosome is composed, the total genes are generated as follows: ((number of distributors × number of products × number of period) + (number of manufacturers × number of products × number of period)) × 2. For example, if there are two distributors, two manufacturers, two suppliers, and two products ordered by each distributor, respectively, and periods are divided into five, the representation of the chromosome is exhibited as Fig. 1. A chromosome has 80 genes, and the reason for this is to simultaneously decide the period to order, production period, and a manufacturer and a supplier. In the representation of a chromosome to decide a distributor's period to order, 1 is represented so that an order can always be placed in the first period for each product. The reason is that no inventory was assumed in the previous period. In the decision regarding production periods and outputs of a manufacturer, 1 is represented so that production can always be conducted in the first period for each product. When no inventory was assumed in the previous period, it is to make a solution that can be executed.

3.2 Analysis of a Chromosome

The chromosome represented in Fig. 1 is interpreted as demonstrated in Fig. 2 so as to select the quantity of order and period to order, a manufacturer to produce and output quantity, and a supplier to supply raw materials. To decide the quantity of order and period to order, a binary representation method was used. If the value of the gene represented in each period is 1, the quantity of order equivalent to the demand quantity in the concerned period is ordered, and if not, the order is not ordered. For instance, the first gene of the chromosome has the value of 1 in Fig. 2, this means that distributor 1 has to order for product 1 in period 1 equivalent to demand quantity. For the gene value of 0, it is interpreted that order is simultaneously placed in the previous period, whose gene value is 1. In Fig. 2, the value of the third gene in the first row is 0. This means that an order is not placed in period 3, but is placed in period 2 added with demand quantity of period 3. Therefore, the quantity of 230 is ordered in period 2. The distributor holds 100 units of product 1 in period 2. Accordingly, additional inventory cost is generated. After the quantity of order and period to order are decided. Manufacturer producing the order quantity has to be decided, through the second row of the chromosome in Fig. 2. Manufacturers can be decided by the genes represented as the manufacturer's number. For example, the first gene value is 1 in the second row. This means that manufacturer 1 produces ordered quantity 120 for product 1 which has been ordered by distributor 1. Each manufacturer's production period and output is decided through the third row of the chromosome. The chromosome of the third row uses the same binary representation and interpretation methods which were used to decide the quantity of an order and period to order. For an instance, the fourth gene in the third row is 1. This means producing order quantity of 180 in period 4. Meanwhile, the gene value is 0 in the period 5, and this means producing order quantity of 330 in period 4 added with quantity of 150 of period 5. In the third row, the chromosome representation of product 2 of manufacturer 1 is interpreted in a slightly more complex manner. In period 1 and 2, there was no output that manufacturer has to produce, but the representation of the chromosome is 1. In this case, output is interpreted as 0, when there is nothing to actually produce. However, when production is not conducted in the remaining periods, the output is decided by adding all the output in those periods. While reading the chromosome for allocation of production output, in the period that is represented as 1, the output needed in the period and following periods with 0 is produced. In the chromosome representation in Fig. 2, in the case that only period 1 and 2 are represented as 1 among five periods, it is interpreted that output of 170 is produced in period 2, which is needed in the following periods. In period 1, it is interpreted that no production is made, because there is no required output. The chromosome in the fourth row is to select a supplier. This also indicates a supplier's number in the same way used to select a manufacturer above. The first gene 1 in the fourth row means that manufacturer 1 receives raw materials required for the production of the quantity of 120 from supplier 1.

	Distributor 1										Distributor 2									
	Product 1					Product 2					Product 1					Product 2				
Period	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Demand Quantity	120	130	100	100	80	60	50	50	80	50	200	180	180	200	150	30	50	40	40	40
Chromosome	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1
Quantity of Order	120	230	0	180	0	110	0	50	80	50	200	560	0	0	150	80	0	80	0	40
Chromosome	1	1	2	1	2	2	1	2	2	1	2	1	1	2	1	2	2	1	1	1
	Manufacturer 1										Manufacturer 2									
	Product 1					Product 2					Product 1					Product 2				
Production Quantity	120	230 560	0	180	150	0	0	80	0	50 40	200	0	0	0	0	110 80	0	50	80	0
Chromosome	1	1	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0
Output	120	790	0	330	0	0	170	0	0	0	200	0	0	0	0	240	0	0	80	0
Chromosome	1	2	1	2	2	1	1	2	1	2	2	2	1	1	2	1	2	2	1	1

Fig. 2. Interpretation of chromosome representation

3.3 Crossover and Mutation

This study applied the crossover which can inherit parents' chromosome order and type properly. Once parent 1 and parent 2 are selected, this crossover randomly generates number 1 and 2, which indicate parents, as many as the length of chromosome genes in order to inherit genes from one parent of the two parents. The length of the chromosome corresponds to the longer chromosome between the result of (number of distributors × number of products × number of period) and the result of (number of manufacturers × number of products × number of period). And then, a new chromosome is generated by inheriting all the genes within the column from the parent who corresponds to the random number. This process is conducted again by exchanging parent 1 and parent 2. Between two children generated in the above mentioned manner, one, which has better value of objective function, is sent to the next generation. An example of crossover is given in Fig. 3.

Parent 1

1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1
1	1	2	1	2	2	1	2	2	1	2	1	1	2	1	2	2	1	1	1
1	1	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0
1	2	1	2	2	1	1	2	1	2	2	2	1	1	2	1	2	2	1	1

Random number

2	**2**	**1**	**2**	**2**	**1**	**1**	**2**	**1**	**1**	**2**	**1**	**2**	**1**	**1**	**2**	**1**	**2**	**2**	**1**

Parent 2

1	0	1	0	0	1	1	0	1	1	1	0	1	0	1	1	1	1	0	1
2	1	1	2	1	2	1	1	1	2	1	1	2	2	1	1	2	2	1	2
1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	1	0	0	0	1
2	1	2	2	1	2	1	2	1	2	1	2	1	2	2	1	1	2	2	1

Offspring 1

1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	0	1	0	1
2	1	2	2	1	2	1	1	2	1	1	1	2	2	1	1	2	2	1	1
1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0
2	1	1	2	1	1	1	2	1	2	1	2	1	1	2	1	2	2	2	1

Fig. 3. An example of crossover

This study uses a mutation method in which genes of all the products in relation to a distributor and a manufacturer in each period are mutually exchanged with the genes of another distributor and manufacturer. When multi-distributors and manufacturers exist, genes are mutually exchanged by randomly selecting distributors and manufacturers. An example of mutation is given in Fig. 4.

1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1
1	1	2	1	2	2	1	2	2	1	2	1	1	2	1	2	2	1	1	1
1	1	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0
1	2	1	2	2	1	1	2	1	2	2	2	1	1	2	1	2	2	1	1

1	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1
2	1	1	2	1	2	2	1	1	1	1	1	2	1	2	2	1	2	2	1
1	0	0	1	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	0
2	2	1	1	2	1	2	2	1	1	1	2	1	2	2	1	1	2	1	2

Fig. 4. An example of mutation

3.4 Selection Method and Objective Function

As a selection method, a seed selection is used [7]. A seed, which corresponds to a father, is randomly selected among excellent individual group that are within top rank in a population. The mother individual is randomly selected from the entire group. These are used as parents, and they are returned to their own group, so that they can be selected again. The next generation is newly formed by using a selection method from the current generation and by using genetic operators. The next generation is formed by generating new individuals that are equal to the number of individuals in the previous population. After that, bad individuals are replaced by good ones in elitism size by applying elitism.

Integrated planning for minimum cost in SCM means an efficient operation for distributors, manufacturers, and suppliers. When a chromosome is represented in a permutation form, we read the genes in chromosome row by row from left to right in order to get the value of objective function. The total cost of the following is calculated: cost of order, inventory cost, transport cost from a manufacturer to a distributor according to the selection of the manufacturer, production preparation cost and inventory cost according to output of a manufacturer at each period, and the price of raw materials supplied by a supplier and transport cost from a supplier to a manufacturer, according to the selection of a supplier. The object is to perform an integrated production and distribution planning with minimum total cost.

3.5 Revision of Chromosome Representing Method

In the generation stage of the chromosome in the first row to decide a distributor's order quantity at each period, and the chromosome in the third row to decide production output at each period, it was designed to always generate 1 in period 1. The reason is to make a solution that can be executed because it is assumed that there is no inventory in the previous period of period 1. However, in Fig. 2, the random value 1 in the period 1 and 2 for the product 2 of manufacturer 1 was generated, although there is no quantity of order to produce in the corresponding period. In this case, the quantity of 170 is produced in period 2, and therefore, additional inventory expense is generated during one period. Therefore, the expense needs to be reduced. In this study, for the initial period with quantity of order and production for product, the random value has been revised to 1 for order and production. This leads to the reduction of unnecessary inventory expense in a distributor and manufacturer. This revision process is undertaken before sending the individual to the next generation, after conducting genetic operations for the generation of the next generation. An example of revised chromosome is given in Fig. 5.

	Manufacturer 1									
	Product 1					Product 2				
Production Quantity	120	230 560	0	180	150	0	0	80	0	50 40
Chromosome	1	1	0	1	0	1	1	**0**	0	0
Output	120	790	0	330	0	0	170	0	0	0
Chromosome	1	2	1	2	2	1	1	2	1	2

	Manufacturer 1									
	Product 1					Product 2				
Production Quantity	120	230 560	0	180	150	0	0	80	0	50 40
Chromosome	1	1	0	1	0	1	1	**1**	0	0
Output	120	790	0	330	0	0	0	170	0	0
Chromosome	1	2	1	2	2	1	1	2	1	2

Fig. 5. An example of revised chromosome representing method

4 Performance Evaluation

In this chapter, the performance of proposed GA is exhibited through a comparison with the optimal solution. First, experiments were conducted to decide the values of parameters of GA. As the values of the parameters used in this study, the size of population was 1000, number of generations 200, seed range for selection 300 and size of elitism 100, and mutation rate and cross rate is 0.1, and 0.9, respectively. To generate a problem for evaluation, demand quantity was generated in the unit of 10 between 50 and 200, and the order cost of a distribution center, and inventory cost per period at a factory and a distribution center were set 100 and 5 respectively. Transport cost from a factory to a distribution center was set between 3 and 5, and transport cost from a supplier to a manufacturer was generated between 10 and 15 at integer value. Production cost at each manufacturer was generated between 35 and 40. Table 1 compares the optimal solution obtained in the small size problem with the results obtained through GA proposed in this study. To obtain the optimal solution of a mathematical model, the ILOG CPLEX 10.0 package was used. The GA was developed based on Java 2 Standard Edition 1.5. The solutions could not be obtained using ILOG in the problems bigger than problem 5 in Table 1. The results are the best solutions obtained in 100 repeated experiments.

Table 1. Results of performance evaluation for proposed GA

Problem	Problem Size					Optimal Solution	GA	Optimality (%)
	Distributor	Product	Manufacturer	Supplier	Period			
1	2	2	2	2	5	96,630	96,630	100
2	3	2	2	2	5	160,090	160,090	100
3	2	2	3	2	5	96,630	96,630	100
4	3	5	3	3	5	506,394	507,460	99.8
5	4	5	4	4	5	603,997	605,280	99.8

A comparison between a case using chromosome revision and a case using the initial chromosome representation was made in bigger problems. Using a Pentium Ⅳ 3.2GHz (Memory: DDR2 521MB), the experiment was repeated 100 times. These solutions are exhibited in Table 2. Through the experiment result, we can verify that it produces better solutions in both the best solution and average solution, when a method of chromosome revision which can reduce inventory expenses of a distributor and manufacturer is used.

Table 2. Results of performance evaluation for proposed GA in bigger problem

Problem	Problem Size					GA			GA (Chromosome Revision)		
	D	P	M	S	P	Best	Average	Time (sec.)	Best	Average	Time (sec.)
1	6	4	7	5	5	745,040	745,837	106	744,700	745,462	107
2	7	5	8	4	5	1,100,900	1,102,461	127	1,100,530	1,101,532	127
3	7	7	10	6	7	2,133,080	2,138,908	201	2,129,890	2,133,677	204
4	8	4	7	5	5	1,024,870	1,026,139	111	1,024,810	1,025,780	112
5	8	6	9	5	7	2,096,110	2,098,904	276	2,095,140	2,097,227	277
6	9	5	8	4	5	1,445,330	1,447,573	173	1,445,290	1,446,542	173
7	9	7	10	7	7	2,746,790	2,752,964	315	2,742,460	2,747,181	317
8	10	10	10	10	10	6,209,990	6,226,548	461	6,196,380	6,209,590	465

5 Conclusion

In this paper, we proposed a new GA for integration of production and distribution planning in supply chain. Unlike the conventional method to solve various SCM problems in sequence, the new method is to simultaneously solve the production and distribution planning problem as integrated problem. The efficiency of the GA presented in this paper has been proved through performance evaluation in the problems with various sizes. In this experiment, an optimal solution which approximates 100% efficiency was obtained in test problem. The time to obtain such result took less than 20 seconds. This proves the performance of GA. Fast speed and simple structure of the GA makes it to be easily applied in real world.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (The Regional Research Universities Program/Research Center for Logistics Information Technology).

References

1. Byrne MD, Bakir MA (1999) Production Planning using a Hybrid Simulation-analytical Approach. International Journal of Production Economics 59:305-311
2. Dellaert N, Jeunet J, Jonard N (2000) A GA to solve the General Multi-level Lot-sizing Problem with Time Varying Costs. International Journal of Production Economics 68:241-257

3. Dhaenens-Flipo C, Finke G (2001) An Integrated Model for Industrial Production-Distribution Problem. IIE Transaction 33:705-715
4. Erenguc SS, Simpson NC, Vakharia AJ (1999) Integrated Production/Distribution Planning in Supply Chains. European Journal of Operational Research 115:219-236
5. Fumero F, Vercellis C (1999) Synchronized Development of Production, Inventory and Distribution Schedules. Transportation Science 33:330-340
6. Kim B, Kim S (2001) Extended Model of a Hybrid Production Planning Approach. International Journal of Production Economics 73:165-173
7. Park BJ, Choi HR, Kim HS (2003) A Hybrid GA for Job Shop Scheduling Problems. Computers and Industrial Engineering 45:597-613
8. Syarif A, Yun YS, Gen M (2002) Study on Multi-stage Logistic Chain Network: A Spanning Tree-based GA Approach. Computers & Industrial Engineering 43:299-314
9. Thomas DJ, Griffin PM (1996) Coordinated Supply Chain Management. European Journal of Operational Research 94:1-15

Bacteria Swarm Foraging Optimization for Dynamical Resource Allocation in a Multizone Temperature Experimentation Platform

Mario A. Muñoz, Jesús A. López, and Eduardo Caicedo

Grupo de Investigación Percepción y Sistemas Inteligentes
Universidad del Valle, Cali, Colombia, South America
{andremun,jesuslop,ecaicedo}@univalle.edu.co

Abstract. In this work, an algorithm based on the Bacteria Swarm Foraging Optimization was used for the dynamical resource allocation in a multiple input/output experimentation platform. This platform, which mimics a temperature grid plant, is composed of multiple sensors and actuators organized in zones. The use of the bacteria based algorithm in this application allows the search the best actuators in each sample time. This allowed us to obtain a uniform temperature over the platform. Good behavior of the implemented algorithm in the experimentation platform was observed.

1 Introduction

The natural selection tends to eliminate animals with poor "foraging strategies" and favor the propagation of the animals with successful ones, since they have more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or redesigned. This evolutionary principle has led the scientists in the field of "foraging theory" to model the foraging activity as an optimization process. The balance between the energy intake and the time spent in its search has been "engineered" into what is called an "optimal foraging policy". Optimization models are also valid for "social foraging" where groups of animals cooperate to forage [1].

The E. Coli bacterium is one of this individuals that form groups for social foraging. This bacterium is probably the best understood microorganism, whose entire genome has been sequenced. Also, the E. Coli is capable of reproduce by division and occasionally transfer gene sequences from one to another. The bacterium moves through its medium by two states, the tumble and the run, that allows it to search for food and avoid noxious substances. The knowledge of the characteristics and behavior, known as chemotaxis, of the E. Coli and its interactions between each other and the environment, has allowed the development of the *Bacteria Swarm Foraging Optimization (BFSO)* algorithm [2].

In this paper, an algorithm based on the Bacteria Swarm Foraging Optimization is used to dynamically allocate the time of ignition of an actuator in a Multizone Temperature Experimentation Platform (MuTEP) and to achieve a uniform temperature over a particular area. The MuTEP, presented in [3], is a multiple input / output plant that emulates the workings of a system designated to control the temperature over a surface.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 427–435, 2007.
springerlink.com

The paper is organized as follows. First we show the main concepts of the BSFO algorithm. Next we describe the experimentation platform used to proof our algorithm. Later we continue with the explanation of the bacteria algorithm for dynamical task allocation in the experimentation platform. Finally we show some results when the algorithm was applied to the platform.

2 The Bacteria Swarm Foraging Optimization

Suppose that we need to find the minimum of $J(\theta), \theta \in R^p$, when we do not have a deterministic description of $J(\theta)$ or its gradient. This problem becomes a non gradient optimization problem, where the ideas from bacteria foraging can be used. Suppose that θ is the position of the bacteria and $J(\theta)$ represent the environment conditions, with $J(\theta) < 0$, $J(\theta) = 0$ and $J(\theta) > 0$ represents that the bacteria location is a nutrient rich, neutral or noxious environment, respectively. Basically, the chemotaxis is a foraging behavior where bacteria tries to climb up the nutrient concentration, avoid noxious substances and search for ways out of neutral media by a random walk.

A chemotactic step *j* is defined as a tumble followed by a tumble or a tumble followed by a run, a reproductive step *k* is defined as the selection of the fittest in the population and its splitting, and a elimination–dispersal event *l* as the selection of random individuals and its relocation in a new random position. Then,

$$P(j,k,l) = \theta_i(j,k,l) \mid i = 1,2,\ldots S \qquad (1)$$

are the positions of each member of the *S* bacteria population at *j-th* chemotactic step, *k-th* reproductive step and *l-th* elimination and dispersion event. Then $J(i,j,k,l)$ is the location cost of the *i-th* bacteria $\theta_i(j,k,l) \in R^p$, and N_c as the bacteria's life time in chemotactic steps. To represent a tumble, a length unit in a random direction y (j) is generated:

$$\theta_i(j+1,k,l) = \theta_i(j,k,l) + C(i) \cdot \psi(j) \qquad (2)$$

where $C(i)$ is the size of the step at the direction $\psi(j)$. If in $\theta_i(j+1,k,l)$, the value of $J(i,j+1,k,l)$ is less than in $\theta_i(j,k,l)$, then a new step is taken until a maximum of N_s, making this cycle a chemotactic step.

After N_c chemotactic steps, a reproduction step is taken. For the reproduction, the healthiest bacteria are split and the others are eliminated, maintaining a constant population. After N_{re} reproduction steps, a dispersion and elimination event is made, where each bacteria is subject to relocation with a probability p_{ed}. After N_{ed} dispersion and elimination, the algorithm ends. The population size *S* is restricted to an even number, so the population can be easily kept constant.

3 The Multizone Temperature Experimentation Platform

The Bacteria algorithm was tested in a Multizone Temperature Experimentation Platform (MuTEP) [3], which was composed of two parts, a process stage, and a data acquisition stage.

The process stage, shown in the figure 1, is an emulation of a planar temperature grid. This is a system that exhibits effects that are difficult to model, especially strong interactions between zones. Therefore, it requires the use of particular control strategies. This type of system is mainly used in the semiconductor industry for the elaboration of crystals and the generation of photo resistive layers, processes that require maintaining a constant surface temperature [4, 5]. The shape of the grid was selected because of its symmetry and the difficulty in raising the edges temperature.

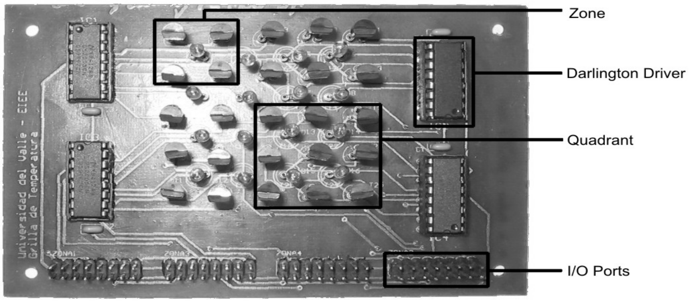


Fig. 1. Process stage of the Multizone Temperature Experimentation Platform (MuTEP)

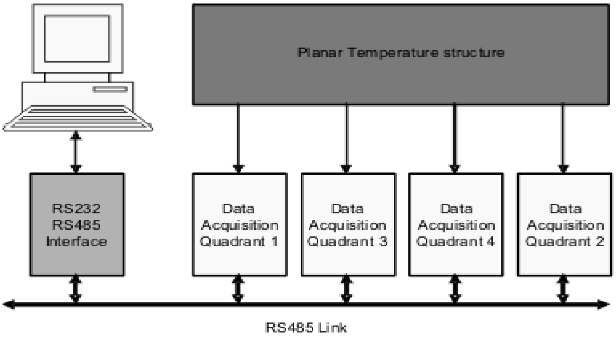

Planar Temperature structure
RS232
RS485
Interface
Data Acquisition Quadrant 1
Data Acquisition Quadrant 3
Data Acquisition Quadrant 4
Data Acquisition Quadrant 2
RS485 Link

Fig. 2. System Architecture of the Multizone Temperature Experimentation Platform (MuTEP) for management form a single computer

The data acquisition system used was composed of four modules based in a low cost micro controller. Each module controlled a quadrant of the process and enabled communication with a master computer, which contained the management and control algorithm. The basic structure of the system is shown in Fig. 2.

4 Bacteria Algorithm for Dynamical Resource Allocation

We design an algorithm using the BSFO algorithm as base to develop an strategy similar to the shown in [6, 7]. This strategy finds the zone with the lowest temperature and assigns the resource to that zone. To assign the resource implies to turn on the bulb of the zone and, as consequence, the temperature of the zone is increased. This is a centralized approach were a single computational agent takes a destination based in global knowledge.

The searches done by the BSFO are in continuous spaces. In this application, the search space corresponds to the surface defined by the platform, and the bacteria will move between the actuators depending of its food concentration.

The objective function used in our proofs was to obtain a maximum uniform temperature over the process surface using a limited amount of actuators (four maximum in our proofs) over a period of time. The amount of actuators depends of the numbers of bacteria (agents) used in the algorithm. We did proofs using two and four bacteria, because of the limitation of the population size to even numbers. This produce that in each sample time are turn on two or four bulbs.

In our algorithm the bacteria are placed at random at the beginning of the experiment. Then it will have a limited number of attempts per chemotaxis step to find a lower cost position, and will be able to move one position in a sample time. The surface cost is directly related to the surface temperature, making the food sources the location with lower temperature. If a bacteria is unable to find a better position after its attempts, can be eliminated or left in place with a probability of $p_{ed} = 0.1$ (low to avoid random search), to avoid the lockup of the bacteria in one position. Only the nutrients action was used, and there is a restriction in the location of the agents, so they do not use the same position. With these considerations, the developed algorithm can be presented as follows [8]:

1. The bacteria are located at random inside the search space in the first generation.
2. The current cost surface is calculated by the actual value obtained from the sensors in the platform.
3. A random tumble is generated and the next position is find. If the next position exceeds the edges, it will be bounded to the nearest position.
4. A run cost is calculated by the difference between the new position in the current surface and the current position in the previous surface.
5. If the difference value is negative, the bacteria are displaced to the new position and its health is updated. If the difference value is positive, steps 3 to 5 are repeated until the search attempts are exhausted.
6. If no better position than the current is found, a dispersion event is performed. If $rand < p_{ed}$, where *rand* is a random number using uniform distribution, the bacteria is relocated in a random place in the surface and its health reset, otherwise, the bacteria stays in its position and its health stays the same.
7. Steps 3 to 6 are repeated for all the bacteria.
8. If the number of samples per reproductive step are achieved, a reproduction step is performed where the healthiest bacteria are split in two and the offspring is placed in the lowest temperature position beside the parent.

5 Results

For the execution of our algorithm different sizes of the population were used. First, taking in account that each bacteria used turns on an actuator, the maximum size of the population is four bacteria. The experiments were carried on using populations of two and four bacteria. The results for two bacteria are shown in the figure 3 and for four bacteria in the figure 4, during a 1800 samples experiment, with $N_{re} = 10$. The figures show that with four agents a higher temperature was achieved than with two agents. The actuators use pattern show that the corner and edge positions where preferred over the central ones. Nevertheless, with the use of the random dispersion, the central positions where used. The higher actuator use in the central part of the structure indicates that there are variations in the way the structure behaves.

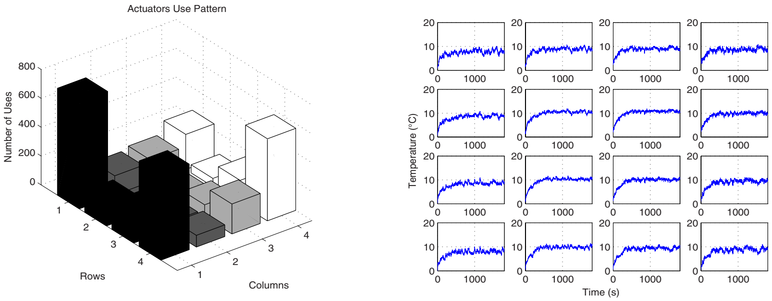


Fig. 3. Results of the experiment obtained with two bacterias: (a-left) shows the number of times an actuator is used in the experiment (b-right) shows the zones' temperatures over the experiment

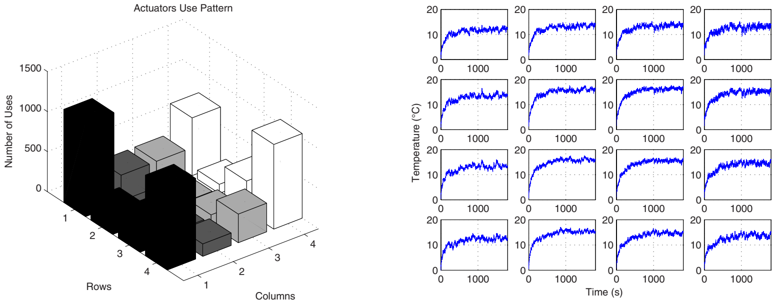


Fig. 4. Results of the experiment obtained with four bacterias: (a-left) shows the number of times an actuator is used in the experiment (b-right) shows the zones' temperatures over the experiment

For the results evaluation, a series of parameters where used to check if the obtained temperature surface gets the control goal [8], during a 5 experiment series. These are:

- Maximum average temperature ΔT: Corresponds to the maximum variation of the average surface temperature. With $t_p(t)$ the average temperature at 0 and $t_p(t_{fin})$ the average temperature at the end of the experiment ΔT is defined by the equation 3 The results are shown in the figure 5.

$$\Delta T = t_p(t_{fin}) - t_p(t) \tag{3}$$

- Settling time t_{est} : Corresponds to the settling time taken to achieve the maximum average temperature. The results are shown in the figure 5.
- Settling temperature spreading σ: Correspond to the error of the surface to the average temperature. The results are shown in the figure 6.
- Spreading percentage %σ: The comparision of the sperading and the achived average tempeature is allowed. The results are shown in the figure 6.
- Control action average *CA*: Corresponds to the average number of actuators used in a sample. The identification of the control effort used to raise the temperature is allowed and calculated by the equation 4, where $u_j(i)$ corresponds to the control action value of the *j–th* actuator at the time *i*. The results are shown in the figure 7.

$$CA = \frac{1}{t_{fin}} \sum_{i=0}^{t_{fin}} \sum_{j=1}^{L} u_j(i) \tag{4}$$

The results showed that, while the temperature spreading percentage is lower with the bacteria algorithm than with a simple random and sequential selection, and even with an ant algorithm presented in [9], is not better than the simple strategies shown in [6, 7]. The achieved average temperature stays around the values for the number of

Table 1. Evaluation Parameters of the Experiment

Method	ΔT	t_{est}	σ	%σ	CA
Sequential	5,1808	699	1,2579	24,28	1,0000
Random	4,8320	388	1,1621	24,05	0,9339
Simple	5,2275	609	0,3568	6,83	1,0000
Distributed	12,4800	482	0,5748	4,61	3,1189
1 Ant	4,4483	534	0.7532	16.93	1,0000
2 Ants	8,2744	831	1,2089	14,61	1,9872
4 Ants	13,5954	448	1,6402	12,06	3,7740
2 Bacteria	8,3389	482	0,8289	9,94	1,9456
4 Bacteria	13,5789	797	1,3190	9,71	3,6989

agents used. Because of the elimination–dispersal of the agents, some may fall in a filled position, making the *CA* decrease. Finally, the settling time varies depending in the environmental conditions, making this value not very useful in the comparison.

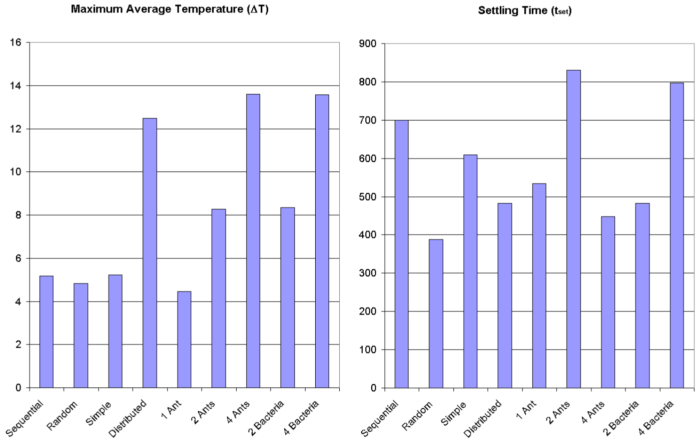


Fig. 5. Maximum average temperature Δ*T* (left) and Settling time t_{est} (right) with the bacteria algorithm and the algorithms described in [3, 9]. The graphics shows that an increase of the population achieves a higher temperature, while the settling time stays around a determinated range.

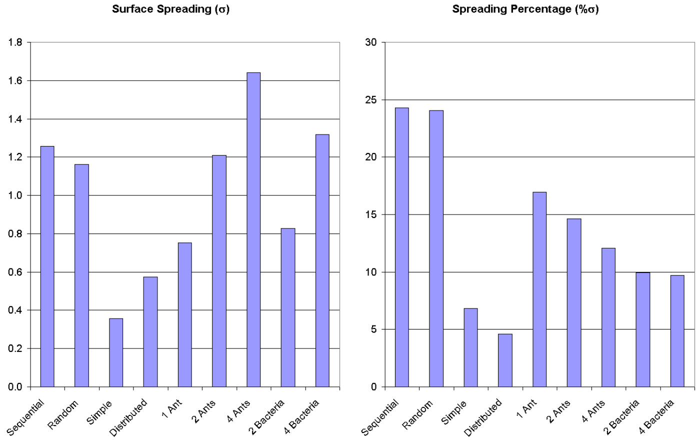


Fig. 6. Settling temperature spreading σ (left) and Spreading percentage %σ (right) with the bacteria algorithm and the algorithms described in [3, 9]. The graphic shows that an increase of the bacteria population increases the temperature spreading.

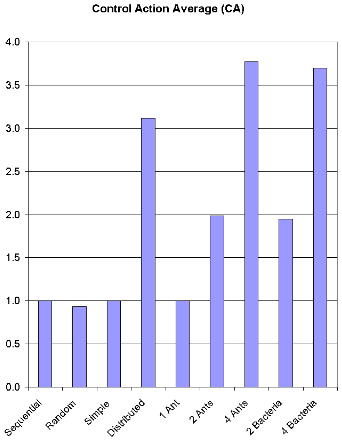


Fig. 7. Control action average *CA* with the bacteria algorithm and the algorithms described in [3, 9]

6 Conclusions

In this paper, the implementation and test of a bacteria based algorithm for dynamical resource allocation in the MuTEP platform was presented. This work represents the first step toward the implementation of a complex intelligent controller. The simulated bacteria presented an intelligent behavior that allowed the achievement of good results, although they are not optimal, they use the most adequate actuators.

The experiment showed that at a large population the temperature variation is increased. The analysis of the results showed that the bacteria algorithm allowed a shorter establishing time than the algorithms showed in [3].

Further work will include the construction of a model based controller for each zone or for the whole system, using a bacteria based algorithm. The development of this controller will include restrictions in the number of actuators used, magnitude of the control action, and continuous operation different from the on-off approximation used in this experiment.

Acknowledgments

We thank Professor Kevin Passino for inspiring this work and his graduate students Nicanor Quijano, Jorge Finke, and Alvaro Gil for their valuable discussions. Part of this work has been supported by Colciencias and Universidad del Valle through a graduate research fellowship awarded to Jesús A. López, and a graduate research scholarship awarded to Mario A. Muñoz, through grant No. 1106–11–17707.

References

[1] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 22(3):52–67, 2002.

[2] K. M. Passino. Distributed optimization and control using only a germ of intelligence. 2000.

[3] M. A. Muñoz, J. A. López, and E. F. Caicedo. Implementation of a distributed control experimentation platform. In IEEE 2005 Conference on Industrial Electronics and Control Applications, 2005.

[4] C. D. Schaper, K. A. El-Awady, and A. E. Tay. Spatially programmable temperature control and measurement for chemically amplified photoresist processing. In K. Dang A. J. Toprac, editor, Procedures SPIE – Process, Equipment and Materials Control in Integrated Circuit Manufacturing V, volume 3882, pages 74–79, 1999.

[5] K. El-Awady, C. D. Schaper, and T. Kailath. Temperature cycling and control system for photosensitive materials processing. Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures, 21(4):1459–1465, 2003.

[6] N. Quijano. Experiments and technologies for decentralized temperature control. Master's thesis, The Ohio State University, 2002.

[7] N. Quijano, A. E. Gil, and K. Passino. Experiments for dynamic resource allocation, scheduling and control. IEEE Control Systems magazine, pages 63–79, 2005.

[8] M. A. Muñoz. Asignación dinámica de recursos con técnicas bio inspiradas para un sistema de control de temperatura mimo. Technical report, Universidad del Valle, 2005.

[9] M. A. Muñoz, J. A. López, and E. F. Caicedo. Ant colony optimization for dynamical resource allocation in a multizone temperature experimentation platform. In 2006 Electronics, Robotics and Automotive Mechanics Conference, volume 1, pages 137–142, 2006.

An Ant Colony Optimization *plug-in* to Enhance the Interpretability of Fuzzy Rule Bases with Exceptions

P. Carmona[1] and J.L. Castro[2]

[1] University of Extremadura, Department of Computer Science, Industrial Eng. School, Badajoz E-06071, Spain
[2] University of Granada, Department of Computer Science and A.I., Computer and Telecommunications Eng. School, Granada E-18071, Spain

Abstract. Usually, fuzzy rules contain in the antecedent propositions that restrict a variable to a fuzzy value by means of an *equal-to* predicate. We propose to improve the interpretability of fuzzy models by extending the syntax of their rules. With this aim, on one hand, new predicates are considered in the rule antecedents and, on the other hand, rules can be associated with exceptions that modify the output of those rules in a region of their covered input space. The method stems from an initial fuzzy model described with the usual fuzzy rules and uses an ACO algorithm to search the optimal set of extended rules that describes this model.

Keywords: Fuzzy modeling, exceptional rules, ACO algorithm, interpretabity.

1 Introduction

Despite the interpretability is one of the distinctive features of fuzzy models, it has often been underestimated in the search for an accuracy improvement. Nevertheless, in the last years, research efforts have been redirected to preserve and enhance the interpretability power of this type of models in order to obtain a good interpretability-accuracy trade-off [3].

One way to improve the interpretability of a fuzzy model consists of trying to identify general rules, so that each rule covers the highest number of examples [4]. It allows to reduce the number of rules in the model and the complexity of the rule (i.e., the number of elements needed to represent the rule). Besides, by extending the syntax of the rules with more predicates than the usual *equal-to* predicate the rule compactness can be increased. Moreover, recently the authors proposed the use of exceptions to further increase the interpretability of models described with fuzzy rules [1,2].

However, finding the optimal set of such general rules with exceptions is not an easy task. Here, it is proposed to search for the best combination of general rules with exceptions that describes an initial fuzzy model, which is mainly a combinatorial problem.

In order to solve it, an ant colony optimization (ACO) algorithm [5] is proposed, an optimization technique based on the emergent behavior that rises from the cooperative search of a set of agents called ants. These ants communicate among them in an indirect way by means of a shared memory that emulates the pheromones that real ants deposit along the paths they trace between the nest and the food. For further readings, see [6].

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 436–444, 2007.
springerlink.com

Next section introduces the syntax of the general rules with exceptions. In Section 3 the different aspects of the ACO algorithm are detailed. In Section 4, some experimental results are provided to show the suitability of the proposal. Finally, conclusions can be found in Section 5.

2 Single, Compound and Exceptional Rules

We consider systems with n input variables $\mathbf{X}=\{X_1,...,X_n\}$ defined over the universe of discourse $\mathrm{U}=\mathrm{X}_1\times...\times\mathrm{X}_n$, and one output variable $\mathbf{Y}$ defined over the universe of discourse V. The fuzzy domain of X_i is denoted as $\widetilde{X}_i=\{LX_{i,1},...,LX_{i,p_i}\}$, where p_i is the number of values associated with the X_i and $LX_{i,j}$ represents both the membership function and the label of the jth value. Analogously, $\widetilde{Y}=\{LY_1,...,LY_q\}$ is the output fuzzy domain.

Usually, the fuzzy rules contain in their antecedent a premise for each input variable which associates that variable with a label from its corresponding fuzzy domain. We refer to this type of rules as *single rules*:

$$R^{i_1...i_n}_{LY^i} : LX_{1,i_1},...,LX_{n,i_n} \rightarrow LY^i . \tag{1}$$

In order to improve the compactness of the fuzzy rules, it is possible to extend their syntax both by associating more than one label to each variable in the antecedent of the rule and by using other predicates different from the *equal-to* predicate. We call *compound rules* to such type of rules:

$$R^i : X_1\, op_1\, SX_1^i, \ldots, X_n\, op_n\, SX_n^i \rightarrow LY^i \tag{2}$$

where $op_i \in \{=,\leq,\geq,\div\}$ ($\div$ means *between*), and where each SX_j^i is one label if $op_i \in \{\leq,\geq\}$, two labels if op_i is $\div$, and a set of labels associated disjunctively in the another case.

Moreover, the compactness of the fuzzy model can be further improved by introducing the concept of *exceptional rules*. An exceptional rule is a compound rule associated with another one –the *excepted rule*– that excludes the latter from being fired in a region of its coverage of the input space, being the exceptional rule fired instead. This idea was introduced in [1,2] to reduce the number of rules needed to describe a model.

For example, given a 2-input/1-output system with the fuzzy domain $\{NL,NS,Z,PS,PL\}$ for all the variables, the excepted and exceptional rules

$$R^1 : X_2 \leq \{NS\} \rightarrow NL, \text{excepting if } X_1 \geq \{PS\}, X_2 = \{NS\} \rightarrow NS$$

equal to the 3 compound rules shown below and comprise 10 single rules:

$$\begin{aligned} R^1 &: X_2 = \{NL\} \rightarrow NL, \\ R^2 &: X_1 \leq \{Z\}, X_2 = \{NS\} \rightarrow NL \\ R^3 &: X_1 \leq \{PS\}, X_2 = \{NS\} \rightarrow NS \end{aligned}$$

3 The ACO Algorithm

In this section, we present an ACO algorithm to improve the interpretability of fuzzy models described initially with single rules. It searches for the best transformation of this set of initial rules (SIR) into a set of compound rules (SCR) possibly with exceptional rules associated to them.

The whole fuzzy modeling process must comprise a first stage to identify the SIR. However, we focus on the second stage, that translates the SIR into a SCR equivalent in the sense that the rules in the latter positively cover all the rules in the former. A compound rule positively cover an initial rule when the input space covered by the compound rule includes the one covered by the initial rule and both rules have the same consequent.

With this aim, an ant can take two types of steps: *inclusion* steps, that add a rule from the SIR to the SCR, or *amplification* steps, that extend the coverage of a rule from the SCR by adding a label to one of its premises.

Since each compound rule stems from a specific initial rule, it will be identified with the order number of the initial rule from which it stems from. In short, R_i is the compound rule that stems from the *i*th initial rule.

3.1 Construction Graph

Ants builds solutions traversing a *construction graph* whose nodes are parts of the whole solution. In our proposal, this graph has two types of edges which allow an ant to make an inclusion or an amplification step. An amplification step, denoted by <*i*,*j*,*k*>, represents the addition of the *k*th label, $LX_{j,k}$, to the *j*th premise in the antecedent of R^i. For the homogeneity of the notation, an inclusion step, denoted by <*i*,0,0>, represents the inclusion of the *i*th rule from the SIR into the SCR, giving the rule R^i.

3.2 Building a Solution

In order to build a solution, firstly an ant is randomly located in an initial rule, so that this rule is included in the SCR of the ant (each ant builds its own SCR). Subsequently, the ant selects one step among the *feasible* transitions from its state.

Briefly, the idea is to amplify the rules in the SCR and to designate rules as exceptions of others if the latter involve the former during their amplification. Once a rule is designated as exceptional, it still can be amplified, then extending the exception of its excepted rule. A rule can only be exception of one rule in the SCR, and this rule can not be simultaneously exception of another rule (that is, nested exceptions are not allowed).

Specifically, a feasible step can be described as follows:[1]

- If it is an inclusion step, it is feasible when the initial rule to be included in the SCR is not positively covered by any of the rules in the SCR yet.
- If it is an amplification step of a compound rule R^i:
  - if R^i is not an exceptional rule, it is feasible when both:
    - the amplification zone (AZ) –i.e., the new regions of the input space covered by the rule after the amplification– covers some uncovered region without initial rules or positively covers some initial rule (already covered or not), and
    - each overlapping zone of R^i with any other rule R^j already in the SCR equals to the coverage of the overlapped rule R^j and, either the consequent of this rule equals to the consequent of R^i or they differ and R^j does not cover negatively any initial rule[2].
  - If R^i is an exceptional rule of R^j, it is feasible if the coverage of R^j includes the AZ and the AZ does not cover negatively any initial rule.

After each amplification step, if the AZ overlaps with any other rule in the SCR, each overlapped rule with a consequent equal to the consequent of the amplified rule will be eliminated –the former subsumes in the latter– and each overlapped rule with a consequent different from the one of the amplified rule will become an exceptional rule of the amplified rule.

3.3 Heuristic Information

The heuristic information provides a way to guide the search to paths containing (a priori) promising steps. The proposed heuristic function attends to both the interpretability of the rule formed by the step –the extension of the coverage achieved with the step– and the rule accuracy –the number of initial rules positively covered by the compound rule after the step.

The proposed heuristic function will be computed dynamically during the evaluation of each feasible transition. The reason relies on the observation that the AZ of a specific amplification step depends on which other amplifications have been previously made over the same rule. Thus, the heuristic function can provide different values for the same step in different solutions or, in general, for the same step placed in different locus along a path. Then, the heuristic value of a step must be evaluated just over the SCR that precedes this step. The heuristic function is the following:

$$\eta = \gamma \frac{Reg_{cov}}{Reg_{max}} + (1-\gamma)\frac{Reg_{+}/(Reg_{-}+1)}{Reg_{max}}. \tag{3}$$

The above function measures both aspects outlined above: interpretability and accuracy of the rule. The parameter γ controls the influence of each of both in the final

[1] The trivial constraint of a step already made by the ant is assumed.

[2] Otherwise, it will be eventually needed to associate an exceptional rule to R^j and, since R^j (in conflict with R^i) will become an exceptional rule of R^i after the amplification step, nested exceptions will be formed.

value. Reg_{cov} is the number of input regions of the fuzzy grid covered by the AZ, Reg_+ is the number of initial rules positively covered by the AZ, Reg_- is the number of initial rules negatively covered by the AZ and Reg_{max} is the maximum number of input regions covered by the AZ when amplifying the variable considered in the step.

That is, if the step is <*i*,*j*,*k*>, then $Reg_{max} = \prod_{l \neq j} p_l$. When the step is an inclusion step, since there is no variable in amplification, Reg_{max} is the maximum number of input regions covered by an AZ, i.e., $Reg_{max} = \prod_l p_l \big/ \min_l p_l$. Moreover, in order to favor the amplification of existing compound rule over the inclusion of new ones and after some experimental results, (3) is divided by 2 when dealing with an inclusion step.

3.4 Memoristic Information

The memoristic information is the information shared by all the ants that stores the experience collected by them about the goodness of the selected steps. Therefore, memoristic information refers to pheromone trails.

The amount of pheromone deposited in an edge depends on the interpretability degree of the solution. The accuracy of the final model is not taken into account, since this model always positively covers all the initial rules (which are the same for all the solutions constructed by all the ants).

The first idea that comes to mind is to deposit in each edge involved in the path of the ant an amount of pheromone proportional to the interpretability of the final fuzzy model. However, this has some drawbacks that make it unsuitable. Firstly, the final rule base could be composed by some rules with a high interpretability and others with a low one, and an even reward for all the steps in the path, with no regard about which rule they contribute to form, seems to be unfair. Secondly, the path can contain steps that are not actually necessary to obtain the final rule base. This is the case of steps involved in the construction of rules that eventually subsumed in other rules and were deleted. No pheromone should be deposited in these edges, since they correspond to "dumb" steps.

Due to this, it has been adopted a strategy for the deposit of pheromone slightly different to the standard one. Instead of evaluating the interpretability of the rule base as a whole, each rule that composes it is separately evaluated, and the steps involved in the construction of that rule (but not the "dumb" steps) are reinforced proportionally to its interpretability.

Specifically, each R^i that describes the final model is inspected, its interpretability value is calculated and an amount of pheromone proportional to this value is deposited in every step of the form <i,·,·> involved in the path. In this sense, both excepted and exceptional rules are equally dealt.

A key issue in the pheromone deposit strategy is the definition of the interpretability of a compound rule. Regarding this, a compound rule is considered as a set of syntactic elements and a measure is defined that evaluates the complexity of these elements. In particular, only the elements in the antecedents of the rules are taken into account, since all the rules considered here have a label in their consequents. Since each variable in the antecedent of a rule is associated with a set of labels and with the aim to consider different relational operators, the following cases considered:

- A set of labels which contains all the fuzzy domain, $\tilde{X}_j$, is associated with no premise.
- A single label, $LX_{i,k}$, is associated with the *equal-to* operator, resulting in the premise $X_j=\{LX_{j,k}\}$.
- A set of consecutive labels which includes the leftmost one of the fuzzy domain, $\{LX_{j,1},...,LX_{j,k}\}$, is associated with the *less-or-equal-to* operator, resulting in the premise $X_j \leq \{LX_{j,k}\}$.
- A set of consecutive labels which includes the rightmost one of the fuzzy domain, $\{LX_{j,k},...,LX_{j,p_j}\}$, is associated with the *greater-or-equal-to* operator, resulting in the premise $X_j \geq \{LX_{j,k}\}$.
- A set of consecutive labels which does not include neither the rightmost nor the leftmost one of the fuzzy domain, $\{LX_{j,k_1},...,LX_{j,k_2}\}$, is associated with the *between* operator, resulting in the premise $X_j \div \{LX_{j,k_1},LX_{j,k_2}\}$.
- A set of non-consecutive labels, $\{LX_{j,k_1},...,LX_{j,k_1}\}$, is associated with the *equal-to* operator, resulting in the premise $X_j = \{LX_{j,k_1},...,LX_{j,k_1}\}$.

Next, for each premise P_j, a complexity cost is assigned as

$$c(P_j) = \begin{cases} 0 & \text{if } \textit{void} \text{ premise} \\ 3 & \text{if the premise is } X_j = \{LX_{j,k}\},\ X_j \leq \{LX_{j,k}\}, \\ & \qquad\qquad\qquad \text{or } X_j \geq \{LX_{j,k}\} \\ 4 & \text{if the premise is } X_j \div \{LX_{j,k_1}, LX_{j,k_2}\} \\ l+2 & \text{if the premise is } X_j \div \{LX_{j,k_1},...,LX_{j,k_l}\}. \end{cases} \tag{4}$$

Then, the complexity of a rule is defined as

$$C(R^i) = \min\left(1, \frac{E + \sum_j c(P_j)}{\sum_j p_j + 1}\right), \tag{5}$$

where E is the number of exceptional rules associated with R^i and where p_j+1 is the maximum complexity of a premise. This measure considers each link of R^i with an exceptional rule as an additional syntactic element.

Finally, the interpretability measure of a rule is defined as

$$I(R^i) = \delta(1 - C(R^i)) + (1-\delta)\frac{Reg_{cov}}{Reg_{max}} \tag{6}$$

where Reg_{cov} is the number of input regions of the fuzzy grid covered by R^i but the ones covered by its exceptional rules, Reg_{max} is the total number of input regions (i.e., $Reg_{max} = \prod_l p_l$), and $\delta \in [0,1]$ weights the influence of both factors. The second factor tries to promote those rules that cover the input space in a high degree, since it is hoped that these rules help to describe the model more easily.

At this point, it must be noted that a deposit of pheromone dependent only on (6) does not gather the moment when the steps of the solution should be taken, though the appropriateness of a step depends on the steps previously taken. Therefore, it should be desirable to make the amount of deposited pheromone also dependent on the locus of each step along the path. That way, the steps located near the start of the path are reinforced and it favors that the first edges are traveled first in subsequent paths.

Due to this, the amount of pheromone deposited on the edge associated to a step *s*:<i,·,·> included in the path of an ant *k* is calculated as

$$\Delta\tau_s^k = I(R^i) \times \frac{L-l+1}{L}, \tag{7}$$

where *L* is the length –number of steps– of the path and *l* is the locus in the path of the step <*i*,0,0> that included the *i*th initial rule in the SCR.

The pheromone evaporation is implemented as usually by using the evaporation rate parameter $\rho \in (0,1]$, and affects to all the edges in the construction graph. Thus, the pheromone update for each possible step *s* is

$$\tau_s = (1-\rho)\tau_s + \sum_{k=1}^{M} \Delta\tau_s^k, \tag{8}$$

where *M* is the number of ants and $\Delta\tau_s^k$ =0 when (7) is not applicable.

Additionally, a measure of the global interpretability of the rule base is needed to provide the best solution at the end of the algorithm. Regarding this, it must be noted that the second factor in (6) is a heuristic information more than an indicator of the interpretability of the final model. Due to it, this factor is discarded, and, we base the global interpretability of the rule base only on the complexity of their rules. Since the interpretability is inversely proportional to that complexity, the maximum complexity of a rule base is integrated into the measure to transform the complexity level of the rule base on an interpretability measure. We consider that a rule base has the maximum complexity when it has the maximum number of rules (i.e., the number of initial rules, *NIR*) and each of them has the maximum complexity (that is, 1). Therefore the measure is defined as

$$I(RB) = \frac{NIR - \sum_i C(R^i)}{NIR}. \tag{9}$$

Finally, the initial amount of pheromone deposited in each edge is often calculated as a function of the quality degree of solutions obtained from some fast method. Here, it has been defined as the average interpretability of the compound rules obtained from the method proposed in [4], which identifies a fuzzy model from a set of examples.

4 Experimental Results

In order to analyze the capability of the algorithm to find good descriptions by using fuzzy compound rules with exceptions, it was compared with the two methods proposed by the authors in [2] which also identify fuzzy compound rules with exceptions. These methods firstly identify a set of single rules and secondly try to improve this initial rule base by amplifying the rules and by including exceptions into the rules. The method LMSFRWE+ is an improvement of LMSFRWE that uses a greedy strategy to select the most promising amplification among a set of candidates.

The ACO algorithm was applied to the same functions used in [2]:

$$f_1 : [-1,1]\times[-1,1] \rightarrow [-1,1]; f_1(x_1,x_2) = (x_1 + x_2)/2$$
$$f_2 : [-1,1]\times[-1,1] \rightarrow [-1,1]; f_2(x_1,x_2) = 1 - | x_1 - x_2 |$$
$$f_3 : [0,3]\times[0,3] \rightarrow [0,1]; f_3(x_1,x_2) = (\sin({x_1}^2)\cdot e^{-x_1} + \sin({x_2}^2)\cdot e^{-x_2} + c_1)/c_2$$

where c_1=0.2338 and c_2=0.8567 restricts the values of f_3 to [0,1].

The modeling processes in [2] were carried out with 3 training set sizes: 20, 50, and 100 randomly generated examples. For each algorithm and training set size, 100 runs were done. Since the input of the proposed ACO algorithm is a set of single rules, it was applied to each initial rule base obtained in the first stage of both methods in [2]. Equally spaced partitions with 7 triangular membership functions were used for all the domains.

Table 1. Interpretability sesults

		Training set size								
Function	Method	20			50			100		
f1	LMSFRWE	12.4	(9.2)	[4.6]	17.8	(4.5)	[3.6]	19.9	(1.6)	[3.3]
	LMSFRWE+	11.0	(8.7)	[4.8]	16.6	(5.4)	[3.7]	19.1	(1.9)	[3.3]
	ACO algorithm	8.9	(1.2)	[3.7]	13.0	(2.0)	[3.3]	13.2	(3.6)	[3.4]
f2	LMSFRWE	14.3	(9.2)	[4.3]	23.0	(6.2)	[3.2]	28.3	(2.1)	[2.7]
	LMSFRWE+	13.0	(9.4)	[4.5]	21.7	(7.8)	[3.3]	27.5	(3.3)	[2.8]
	ACO algorithm	10.6	(0.9)	[3.4]	18.1	(1.6)	[3.1]	21.9	(3.0)	[3.5]
f3	LMSFRWE	12.7	(8.7)	[4.5]	18.5	(5.7)	[3.6]	20.0	(2.0)	[3.3]
	LMSFRWE+	11.1	(9.0)	[4.8]	16.3	(6.1)	[3.8]	18.6	(2.5)	[3.4]
	ACO algorithm	9.5	(0.9)	[3.9]	14.4	(1.4)	[3.7]	15.9	(1.9)	[3.7]

Besides, in the ACO algorithm the original Ant System version proposed in [5] was implemented, which applies a random proportional rule in order to select each step and whose pheromone deposit mechanism is run once the solution is completed.

The parameters were set as M=5, ρ=0.1, γ=0.25, and δ=0.75. For the random proportional rule, the parameters α=1 and β=0.5 were used. The number of cycles was bounded to NC_{max}=20 and a stagnation condition was satisfied and the corresponding run finished if all the ants in a cycle constructed the same SCR.

Table 1 summarizes the interpretability results, which are averaged values over the 100 runs. They are depicted as the averaged number of rules describing the model (but the exceptions), the averaged number of exceptions in parenthesis, and the averaged number of labels in the antecedent of the rules in brackets. It can be observed that the ACO algorithm clearly improves the interpretability obtained by the other methods.

5 Conclusions

The interpretability of fuzzy models is one of their key features. Due to it, this feature must be preserved and enhanced instead of being relegated to a secondary place. With this aim, an ACO algorithm has been proposed to improve the interpretability of an initial fuzzy model described with single rules. This goal is achieved by searching good descriptions of the fuzzy model by means of compound rules and by adding exceptional rules to them. The construction graph allows to represent each solution as the successive addition of labels to the premises in the antecedent (rule amplifications) and the possible association of rules as exceptions of other rules.

The experimental results indicate that good combinations of compound rules with exceptions emerge from the cooperative behavior of ants and from the heuristic proposed to guide the ants toward interesting solutions.

References

1. Carmona P, Castro JL, Zurita JM (2004) FRIwE: Fuzzy rule identification with exceptions. IEEE Trans Fuzzy Syst, 12(1):140–151
2. Carmona P, Castro JL, Zurita JM (2004) Learning maximal structure fuzzy rules with exceptions. Fuzzy Sets Syst, 146(1):63–77
3. Casillas J, Cordón O, Herrera F, Magdalena L (eds) (2003) Accuracy Improvements in Linguistic Fuzzy Modelling, vol 129 of Studies in Fuzziness and Soft Computing. Springer-Verlag, Heidelberg, Germany
4. Castro JL, Castro-Schez JJ, Zurita JM (1999) Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. Fuzzy Sets Syst, 101:331–342
5. Dorigo M, Colorni A, Maniezzo V (1996) The ant system: Optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B, 26(1):29–41
6. Dorigo M, Stützle T (2004) Ant Colony Optimization. MIT Press

Performance Improvement of the Attitude Estimation System Using Fuzzy Inference and Genetic Algorithms

Min-Soo Kim

Vehicle Dynamics Research Team, Korea Railroad Research Institute
360-1 Woram-dong, Uiwang-si, Gyeonggi-do, Korea
Ms_kim@krri.re.kr

Abstract. This paper describes the development of a closed-loop attitude estimation system for determining attitude reference for vehicle dynamics using fuzzy inference and Genetic Algorithms (GAs). By recognizing the situation of dynamic condition via fuzzy inference process, each parameter of the estimator of the attitude estimation system is determined online adaptively under varying vehicle dynamics. For this solution scheme, fuzzy rules and reasoning method are consider based on the error signal of the gyro and accelerometer and the magnitude of dynamic motion, and the input gains of the fuzzy systems and the position of the membership function are optimized based on the GAs. Computer simulations based on the real test data of a vehicle are used in the study to assess the system performance with the proposed fuzzy-GAs estimation method.

Keywords: Attitude Estimation System, Fuzzy Inference, Genetic Algorithm.

1 Introduction

Research and development in the advanced control technologies of vehicles related to the railway, the aviation, and the like have been interesting topics from the viewpoint of the improvement of riding comfort and running safety, reducing maintenance cost and noise, and so on. The advanced control system of vehicles is made up of several control units and sensors for each actuator to determine the dynamics of the whole vehicle. Especially, an attitude measuring system is indispensable for the control system all the time, and it needs to be as accurate as possible [1]~[4].

The roll, pitch and yaw rate of the vehicle are measured using rate gyros with respect to its body axis system. As a physical instrument, rate gyros also carry some errors such as axis misalignment, fixed bias, drift bias, fixed scale factor errors, asymmetric scale factor error, and so on. The bias drift would be the most serious and deteriorate the accuracy of an attitude estimation system. Therefore many researchers have been interested in looking at ways to replace costly inertial-grade rate gyros of the attitude estimation system with low cost inertial sensors to minimize drift or bias with an estimating filter [5][6]. In order to reduce cost and increase reliability, we propose attitude estimation system with fuzzy inference and Genetic Algorithms (GAs) to replace traditional expensive gyro systems.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 445–454, 2007.
springerlink.com

The idea of fuzzy sets was introduced by Zadeh in 1965 [7]. Currently one of the more active areas of fuzzy logic applications is control system. Fuzzy control systems are rule-based systems, which have a set of fuzzy IF-THEN rules represents a control decision mechanism to adjust the effects of certain causes coming from the system [8][9]. We take simplified fuzzy reasoning method in the defuzzifier and utilize the reduced rule fuzzy controller as our attitude reference systems.

GAs were invented by John Holland and developed by him and his students and colleagues [10][11]. GAs are a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operate on a population of potential solutions by applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that created from just as in the natural adaptation. This is repeated until some condition is satisfied, for example, a number of generations or improvement of the best solution. In this paper, an optimized closed-loop attitude estimation system based on fuzzy inference and GAs for the vehicle using low-cost solid-state inertial sensors will be derived. For choosing the optimal values, we utilize on-line scheduling method with fuzzy inference and GAs.

This paper is organized as follows. First, genetic algorithm is briefly introduced in Section 2. Conventional attitude estimator on gyros and accelerometers and fuzzy-GAs attitude estimator are described in section 3, and simulation is developed in section 4. Finally, we close the paper with a brief conclusion in Section 5.

2 Genetic Algorithms

The characteristics of GAs are the following:

- GAs are parallel-search procedures that can be implemented on parallel processing machines for massively speeding up their operations.
- GAs are applicable to both continuous and discrete optimization problems.
- GAs are stochastic and less likely to get trapped in local minima, which inevitably are present in any practical optimization application.
- GAs have flexibility both structure and parameter identification in complex models such as neural networks and fuzzy inference systems.

In each generation, the GAs construct a new population using genetic operators such as crossover and mutation, that is, members with higher fitness values are more likely to survive and to participate in mating crossover operations. After a number of generations, the population contains members with better fitness values in which this is analogous to Darwinian models of evolution by random mutation and natural selection. GAs are something referred to as methods of population based optimization that improves performance by upgrading entire populations rather than individual members.

The outline of the basic GA is the following:

Step 1: Initialize a population with randomly generated individuals and evaluate the fitness value of each individual.

Step 2:

(1) Select two members from population with probabilities proportional to their fitness value.

(2) Apply crossover with a probability equal to the crossover rate.

(3) Apply mutation with a probability equal to the mutation rate.

(4) Repeat (1) to (4) until enough members are generated to form the next generation.

Step 3: Repeat steps 2 and 3 until a stopping criterion is met.

Fig. 1 shows how to produce the next generation from the current one.

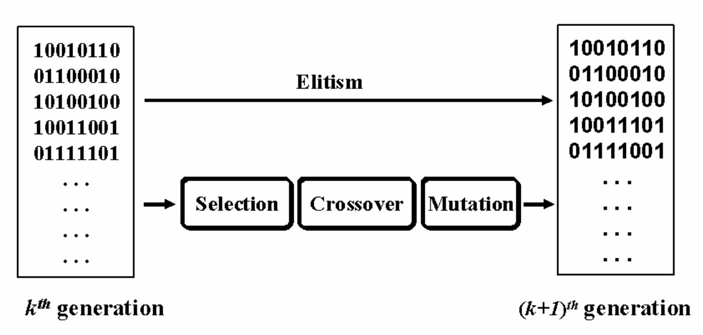


Fig. 1. Procedure of the next generation

We use multiple population method to optimize several variables of controller in QFT with real valued representation for the individuals and stochastic universal sampling for selection function. Multiple populations are a routine for exchanging individuals between subpopulations. This method improves the quality of the results obtained using GAs compared to the single Population GA. The transfer of individuals between subpopulations is implemented and a single scalar is used to determine the amount of migration of individuals from on subpopulation to another.

The first step in a GA is to create an initial population consisting of random individuals. Creation of a real-valued initial population produces a matrix containing uniformly distributed random values in its elements.

The offspring of a pair of two parents are computed through recombination procedures. Mutation of real-valued populations mutates each population with given probability and returns the population after mutation. Generally, mutation rate μ has within the range [0, 1]. The mutation of a variable x_{new} from x_{old} is computed as follows.

$$x_{new} = x_{old} + \delta M \, d_{\text{var}} \tag{1}$$

where δ specifies the normalized mutation step size, and mutation matrix M produces an internal mask table determining which variable to mutate and the sign for adding +1 or –1 with equal probability based on μ, d_{var} means the half range of the variables domain.

Crossover performs migration of individuals between subpopulations in current population when it uses multiple populations.

The pseudo code for the generational loop of the multi-population GA is shown as:

```
Initialize GA parameters;
Initialize population $P(t)$ ;
$t$ =0;
Evaluate initial population $P(t)$ ;
While Termination criterion is not satisfied
   Assign fitness values to whole population;
   Select individuals from population;
   Recombine selected individuals;
   Mutate offspring;
   Calculate objective function for offspring;
   Insert best offspring into population $P(t)$ ;
   $t = t+1$ ;
   Exchange individual between subpopulations at specific $t$ ;
End;
```

GA parameters for initialization consist of mutation rate μ_{mut}, maximum number of generations N_{gen}, insertion rate μ_{ins} which specifies that the individuals produced at each generation are reinserted into the population, the number of subpopulations N_{sub}, migration rate μ_{mig} that migrates between subpopulations with probability μ_{mig}, the number of individuals of each subpopulation N_{ind}.

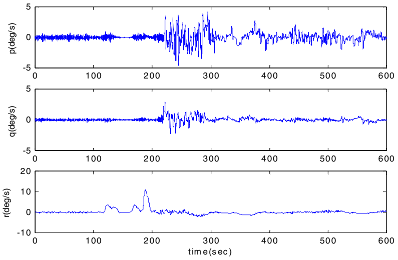
In this paper, we use each parameter in multiple population genetic algorithms with μ_{mut} =1/number of variable, N_{gen}= 150, μ_{ins} = 90 [%], N_{sub} =200, μ_{mig} = 20 [%], and N_{ind} =1000.

3 Attitude Estimation Systems

3.1 Attitude Determination from Inertial Sensors

During all the scheme investigation process in this paper, a set of test data is used [14]. The following figures show time histories of the parameter. All parameters of approximately 600 seconds (10Hz sampling rate) were collected with the solid-state low cost rate gyro and accelerometer. The measurement of angular rates (p, q, r) and accelerations (a_x, a_y, a_z) from rate gyros and accelerometers are shown in Fig. 2 and 3, respectively. Fig. 4 shows attitude angles obtained from the on-board vertical gyro used as the reference standard.

Attitude Euler angles (ϕ, θ, ψ) can be obtained from open-loop integration of equation (2) using first order Euler method of integration and can also be obtained from the accelerometer in equation (3).



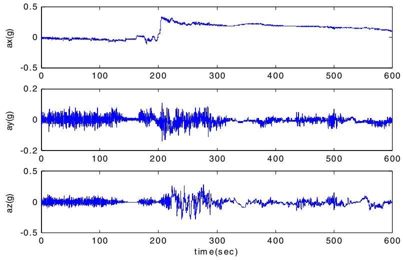


Fig. 2. Test data of the rate gyro measurements

Fig. 3. Test data of the accelerometer measurements

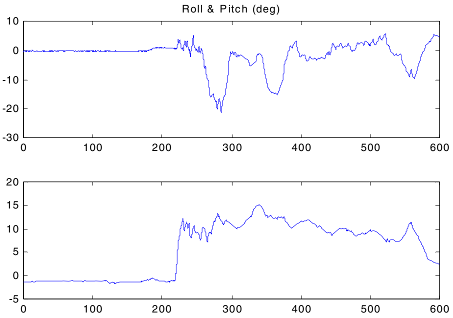


Fig. 4. Attitude angles from the on-board vertical gyro (reference standard)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & w & -v \\ -w & 0 & u \\ v & -u & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} -r^2-q^2 & pq-r^2 & pr+\dot{q} \\ pq+\dot{r} & -p^2-r^2 & rq-\dot{p} \\ pr-\dot{q} & rq+\dot{p} & -q^2-p^2 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + g \begin{bmatrix} \sin\theta \\ -\cos\theta\sin\phi \\ -\cos\theta\cos\phi \end{bmatrix} \tag{3}$$

where (p , q , r) is the roll, pitch and yaw rate measured using rate gyros, (ϕ , θ , ψ) are Euler angles, (u , v , w) are linear velocity components and (p_x , p_y , p_z) are accelerometer coordinates along each axis in the body frame with its origin at the center of gravity of the vehicle.

But, the Euler method of integration shows how the bias errors cause the attitude angles to deviate, and the instability of the integration which drifts as a function of time unless corrected, and the attitude from accelerometers tend to follow the reference output approximately in steady state flight, but large errors and poor reliability result in transient and high dynamic environment [14].

3.2 Conventional Attitude Estimation System

The role of the attitude estimator is to compare attitude angles resulting from the integration of the gyros with the attitude angle products from the accelerometers. The error e_ϕ between $\hat{\phi}_{gyro}$ and ϕ_{accel} is fedback through a proportional and integral controller with a pair of estimator gains α_1^ϕ and α_2^ϕ. A similar procedure could also apply to the pitch channel.

The mathematical relation of the estimated attitude ($\hat{\phi}_{gyro}$), the attitude products from the accelerometers (ϕ_{accel}), and the attitude rate products from the gyros ($\dot{\phi}_{gyro}$) in Laplace form is as follow:

$$\hat{\phi}_{gyro} = \frac{1}{s}\dot{\phi}_{gyro} + \frac{\alpha_2^\phi}{s}\left(\phi_{accel} - \hat{\phi}_{gyro}\right) + \frac{\alpha_1^\phi}{s^2}\left(\phi_{accel} - \hat{\phi}_{gyro}\right) \tag{4}$$

$$\delta\hat{\phi}_{gyro} = \frac{s\delta\dot{\phi}_{gyro} + (\alpha_2^\phi s + \alpha_1^\phi)\delta\phi_{accel}}{s^2 + \alpha_2^\phi s + \alpha_1^\phi} \tag{5}$$

Applying the final value theorem to equation (5), we can get the convergence of the estimated attitude error to the error of the angle calculated from the accelerometer. The controller gains α_1^ϕ and α_2^ϕ in roll channel are chosen by relating them to the cut-off frequency ω_ϕ and damping ratio ς of the estimator as:

$$\alpha_1^\phi = (\omega_\phi)^2 \ , \ \alpha_2^\phi = 2\varsigma\omega_\phi \tag{6}$$

Therefore, the system is only characterized by the cut-off frequency when the damping ratio ς is fixed to a suitable value of 0.707, which is chosen appropriately to optimize the process. The cut-off frequency needs to be optimized to improve the performance of the estimation system over a wide range of dynamic conditions. To solve these problems, we proposed a fuzzy-GAs attitude estimation system algorithm to provide the attitude estimation system with a variable cut-off frequency using an adaptive function (i.e. fuzzy systems and GAs) under varying vehicle dynamics.

3.3 Fuzzy Attitude Estimation System

Fig. 5 shows a fuzzy logic based attitude estimation system. The approach taken here is to exploit fuzzy rules and reasoning to generate parameters of the filtering estimator. The parameters are determined only by the cut-off frequency.

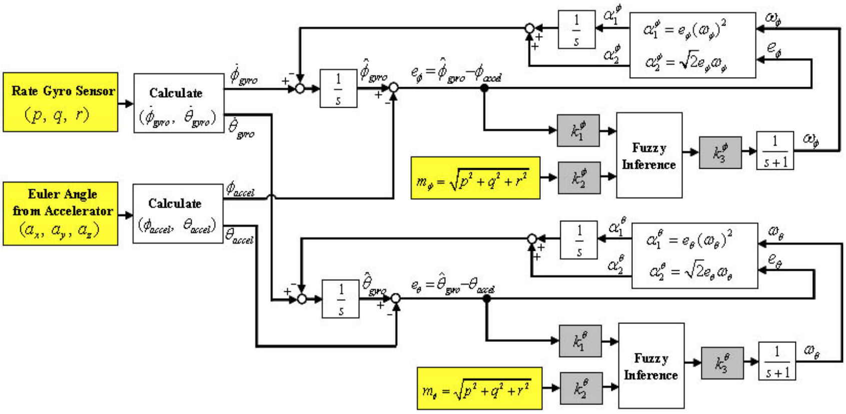

Rate Gyro Sensor
(p, q, r)
Calculate
$(\dot{\phi}_{gyro}, \dot{\theta}_{gyro})$
$\dot{\phi}_{gyro}$
$\dot{\theta}_{gyro}$
Euler Angle from Accelerator
(a_x, a_y, a_z)
Calculate
$(\phi_{accel}, \theta_{accel})$
ϕ_{accel}
θ_{accel}
$\frac{1}{s}$
$\hat{\phi}_{gyro}$
$e_\phi = \hat{\phi}_{gyro} - \phi_{accel}$
α_1^ϕ
α_2^ϕ
$\alpha_1^\phi = e_\phi(\omega_\phi)^2$
$\alpha_2^\phi = \sqrt{2}e_\phi\omega_\phi$
ω_ϕ
e_ϕ
k_1^ϕ
k_2^ϕ
k_3^ϕ
$m_\phi = \sqrt{p^2+q^2+r^2}$
Fuzzy Inference
$\frac{1}{s+1}$
$\hat{\theta}_{gyro}$
$e_\theta = \hat{\theta}_{gyro} - \theta_{accel}$
α_1^θ
α_2^θ
$\alpha_1^\theta = e_\theta(\omega_\theta)^2$
$\alpha_2^\theta = \sqrt{2}e_\theta\omega_\theta$
ω_θ
e_θ
k_1^θ
k_2^θ
k_3^θ
$m_\theta = \sqrt{p^2+q^2+r^2}$
+
-

Fig. 5. Fuzzy based closed loop attitude estimation system block diagram

In the proposed scheme, the cut-off frequency (ω_ϕ, ω_θ), is adaptively determined based on the magnitude of the current error (e_ϕ, e_θ) and the current dynamic motion (m_ϕ, m_θ). Here the magnitude of the dynamic motion m_ϕ is defined as:

$$m_\phi = \sqrt{p^2 + q^2 + r^2} \tag{7}$$

The parameters of the estimation filter is thus obtained by

$$\alpha_1^\phi = \left(e_\phi \omega_\phi\right)^2 \ \ and \ \ \alpha_2^\phi = \sqrt{2} e_\phi \omega_\phi \tag{8}$$

where α is determined by a set of fuzzy rules of the form:

$$\text{if } e_\phi \text{ is } A_i \text{ and } m_\phi \text{ is } B_i\text{, then } \alpha = \alpha_i \tag{9}$$

Here, A_i and B_i are fuzzy sets of the corresponding supporting sets; α_i is a constant. The membership functions of the fuzzy sets for e_ϕ and m_ϕ are shown in Fig. 6. A similar equation for calculating m_θ, α_1^θ, α_2^θ, and e_θ could also apply to the pitch channel.

Table 1. Fuzzy Rules for α

		e_ϕ or e_θ		
		PB	**PM**	**ZO**
m_ϕ or m_θ	**PB**	ZO	ZO	PS
	PM	ZO	PS	PM
	ZO	PS	PM	PB

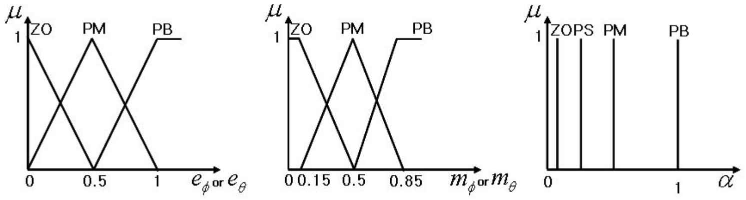


Fig. 6. Membership functions for e_ϕ, e_θ, m_ϕ, m_θ, and singleton membership functions for α

4 Simulations

The proposed fuzzy-GAs attitude estimation system has been tested and compared with a conventional system and a fuzzy inference only. Outputs from the low-cost solid-state inertial sensor during an experimental test of the vehicle were used to implement the estimation scheme. And we also try to optimize the four fuzzy input gains (k_1^ϕ , k_2^ϕ in roll channel and k_1^θ , k_2^θ in pitch channel) and the six position parameters of the output membership function ($ZO_\phi, PS_\phi, PM_\phi$ in roll channel and $ZO_\theta, PS_\theta, PM_\theta$ in pitch channel) in closed loop attitude estimation system using GAs. But we fixed the fuzzy out gain based on simulation study results is k_3^ϕ , k_3^θ = 0.02 (rad/s) and the position parameter PB_ϕ, PB_θ= 1 in the fuzzy system of both channels. That is, there are 10 parameters minimizing the IAE for roll angle and pitch angle.

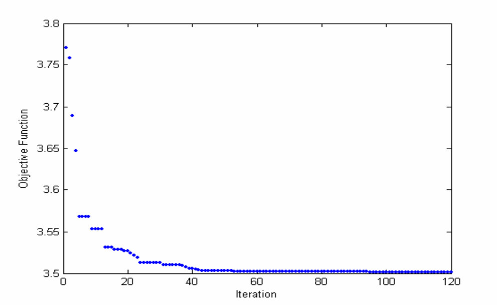


Fig. 7. Objective function values in GAs

Table 2. The values of the optimized parameter of fuzzy system

Roll Channel	Optimized Values	Pitch Channel	Optimized Values
k_1^ϕ	47.42	k_1^θ	9.04
k_2^ϕ	1.02	k_2^θ	0.25
ZO_ϕ	0.0000001	ZO_θ	0.0001
PS_ϕ	0.0001	PS_θ	0.00333
PM_ϕ	1	PM_θ	0.13112

The parameter optimization of the fuzzy based closed loop attitude estimation system developed after 110 generations in Fig. 7, the results are given as Table 2.

Table 3. Experimental results of Integral of the Absolute Error (IAE)

	Conventional in case of $\omega_\phi, \omega_\theta = 0.001$	Fuzzy Inference	Fuzzy Inference + GA Tuning
IAE for Roll Angle	9,578	1,935	1,583
IAE for Pitch Angle	7,964	1,799	1,588

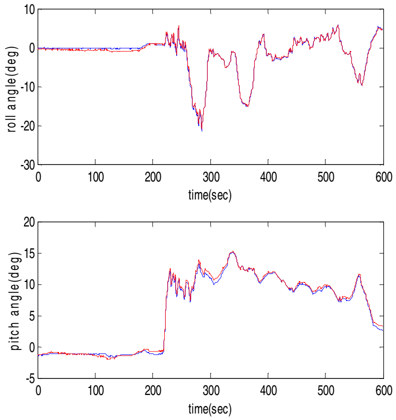


Fig. 8. Response of the fuzzy-GAs attitude-estimation system

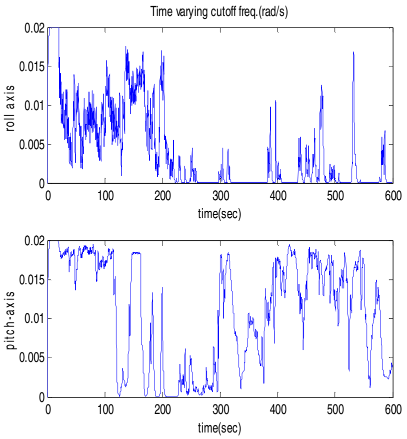


Fig. 9. Variation of the cutoff frequency in the fuzzy-GAs attitude estimation system

Table 3 shows the experimental results of Integral of the Absolute Error (IAE) for the fuzzy based attitude estimation system, and GA fine-tuning based attitude estimation system. In order to verify the effectiveness of the estimation, the levels of agreement between reference model and the scheme results are calculated. This performance is very encouraging for low-cost and low-performance inertial sensors. The time responses for roll and pitch attitudes are plotted in Fig. 8, and Fig. 9 shows how the cutoff frequencies are adapted under dynamic conditions. The results obtained from the on-board high-precision gyro are also presented for comparison.

5 Conclusions

As an alternative to expensive and heavy devices such as vertical gyros on the vehicle system, a solution scheme for a low-cost attitude estimation system has been studied to estimate closed-loop attitude based on the test experimental data. The attitude estimation system consists of 3 single-axis rate gyros in conjunction with 2 single-axis accelerometers. The proposed closed loop attitude estimation system scheme uses fuzzy inference and GAs to determine the filtering estimator parameters by adjusting cutoff frequency and to optimize the parameters of the fuzzy system.

The proposed system has shown a good performance for drift errors form gyro measurement as well as accelerometer measurement noise corruption in all dynamic conditions. The performance improved attitude estimation system by GAs fine-tuning compared to a fuzzy inference only. This performance is very encouraging and indicates that high accuracy attitude estimation system should be possible with low-cost, low-performance inertial.

References

1. Dukkipati, R. V.: Vehicle Dynamics, CRC Press (2000)
2. Garg, V. K. and Dukkipati, R. V.: Dynamics of Railway Vehicle Systems, Academic Press (1984)
3. Siouris, G. M.: Aerospace Avionics Systems: A Modern Synthesis, Academic Press (1993)
4. Wie, B.: Space Vehicle Dynamics and Control, AIAA Education Series, AIAA Inc., USA, (1998)
5. Gilmore, J. P.: Modular Strapdown Guidance Unit with Embedded Micro Processor, Journal of Guidance and Control, Vol. 3(1) (1980) 560-565
6. Koifman, M., Merhav, S. J.: Autonomously Aided Strapdown Attitude Reference System, Journal of Guidance, Control and Dynamics, Vol. 14(6) (1991) 1164-1172
7. Zadeh, L. A.: Fuzzy Sets Information and Control, Vol. 8(3) (1965) 338-353
8. Mamdani, E. H. and Assilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man Machine Studies, Vol. 7 (1) (1975) 1-13
9. Sugeno, M.: Industrial Applications of Fuzzy Control, Elsevier Science Pub. Co. (1985)
10. D. E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Publishing Company (1989)
11. Wright, A. H.: Genetic Algorithms for Real Parameter Optimization, In Foundations of Genetic Algorithms, J. E. Rawlins (eds.), Morgan Kaufmann (1991) 205-218
12. Catford, J. R.: Application of Strapdown Inertial System with Particular Reference to Underwater Vehicle, Strapdown Inertial System, AGARD-LS-95, (1995)
13. Humphrey, I. : Inertial Navigation System for a Micro Unmanned Aircraft, AIAA-97-3567 Paper, AIAA Guidance, Navigation, and Control Conference, New Orleans, (1997)
14. Hong, S. K.: Fuzzy Logic based Closed-Loop Stapdown Attitude System for Unmanned Aerial Vehicle (UAV), Sensors and Actuators A-Physical, Vol. 107, (2003) 109-118

Multi Objective Optimization in Machining Operations

Orlando Durán[1], Roberto Barrientos[1], and Luiz Airton Consalter[2]

[1] Pontificia Universidad Católica de Valparaíso, Av.Los Carrera, 01567,
Quilpué, Chile
orlando.duran@ucv.cl
[2] FEAR, Universidade de Passo Fundo, CP, Passo Fundo (RS), Brasil
lac@upf.br

Abstract. Process Planning activities are significantly based on experience and technical skill. In spite of the great efforts made for planning automation, this activity continues being made in manual form. Process Planning activities are significantly based on experience and technical skills. The advent of the CAM systems (Computer Aided Manufacturing) has partially close the gap left between the Automated Design and Manufacture. Meanwhile, a great dose of manual work still exists and investigation in this area is still necessary. This paper presents the application of a multi objective genetic algorithm for the definition of the optimal cutting parameters. The objective functions consider the production rate and production cost in turning operations. The obtained Pareto front is compared to high efficiency cutting range. This paper also describes one application of the developed mechanism using an example.

1 Introduction

Process Planning is a function that establishes a set of manufacturing operations and their sequence, and specifies the appropriate tools and process parameters in order to convert a part from its initial state to a final form. Computer Aided Process Planning (CAPP) can be considered as the solution that provides a great assistance in this aspect and could even replace the human planners in the planning procedure.

In addition, CAPP technology can assure the integration between CAD and CAM systems and a total consistency and correctness of the developed process plans.

Traditional CAPP systems can be roughly categorized as variant and generative systems. Variant CAPP is based on a Group Technology (GT) coding and classification approach to identify a number of part attributes. These attributes allow the system to select a standard process plan for the part family and accomplish an important part of the planning work. The planner needs to add the remaining of the effort of modifying or fine-tuning the process plan. The standard process plans stored in the computer are manually entered using a super planner concept, that is, developing standardized plans based on the accumulated experience and knowledge of multiple planners and manufacturing engineers. One drawback of the variant CAPP is that the generic family process may not be suitable for a particular new part. The generative CAPP system does not have this drawback because it generates process plans according to the experiences and knowledge of the planners. Instead of retrieving a generic family process and editing it, a process plan is created from scratch by using a set of intelligent tools.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 455–462, 2007.
springerlink.com

An standard CAPP system usually selects machine tools and operations, generates the various parameters required in each operation, routes the selected operations, and so on. While there has been some success in some problematic areas such as in the systematization of feature recognition and operation sequencing, the process of selecting cutting tool and conditions has not been as successful because it depends largely on human experience. As aforementioned, the selection of cutting conditions is an important step in process planning of machining operations. In past years, to determine the optimal cutting parameters, complex mathematical models have been formulated to associate the cutting parameters with the cutting performance. The complexity is augmented if one considers that the cutting parameters are subjected to several constraints such as permissible limit of power, cutting force or surface roughness among others. Main difficulties arise because of the coexistence of parameters that interact in non linear modes. Also, objective functions that are discontinuous pose high difficulty for traditional mathematical techniques. This is the field where heuristics methods like the genetic algorithms offer powerful opportunities. This paper deals with the use of a multi-objective genetic algorithm to effectively perform cutting conditions optimization for turning operations. Additionally, this work attempts to demonstrate that the Pareto front obtained by means of a multi objective optimization process constitutes a good approximation to the high efficiency cutting range.

2 Literature Review

Traditional optimization in machining operations involves the selection of feed and cutting speed according to a variety of economic criteria such as the minimum cost per component, maximum production rate or maximum profit rate [1,2,3]. In those optimization models constraints are considered, such as the machine tool feed and speed boundaries, machine tool maximum feed force, spindle torque and available power. Taylor [4] conducted the earliest research on cutting tool life; the main achievement of that research was the development of the well-known tool life equation. [5] proposed a mathematical model to find the optimal speed and feed rate that provides the highest production rate. In the other hand, cutting conditions are usually selected to minimize operation costs as a form to increase the long-term profits [6,5]. That two cutting conditions, maximum production rate speed and the minima cost cutting speed, together constitute the so-called high efficiency cutting range.

Several types of methods have been used for the optimization of cutting parameters. Direct search method emerges as one of the most popular mathematical optimization methods. Direct search methods compute the first derivative of one objective function and set it to zero. However, for applying these methods the objective function must be continuous and twice differentiable, requirement that it is not easily met in real world problems [7]. [8] pointed out that those single objective approaches have a limited value to fix the optimal cutting conditions, due to the complex nature of the machining processes, where several different and contradictory objectives must be simultaneously optimized. Multi-objective formulations are realistic models for many complex engineering optimization problems. In many real-life problems, objectives under consideration conflict with each other, and optimizing a particular solution with respect to a single objective can result in unacceptable results with respect to the other objectives [9]. A reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution.

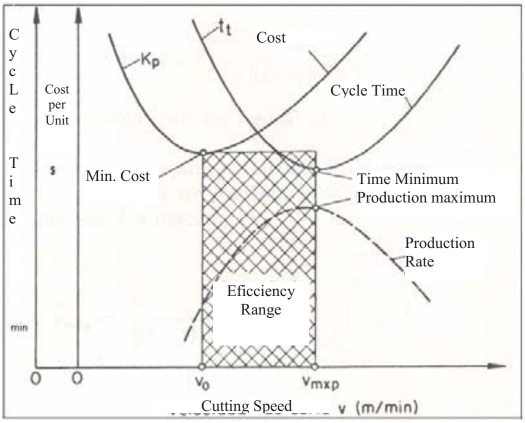


Fig. 1. High efficiency cutting range

An extensive review on meta-heuristics methods utilized within the paradigm of multi-objective programming is presented by [10]. The authors demonstrated the rising popularity of multi objective meta-heuristics since the late nineties. According the authors, several factors influenced the increase, the increase in computer power, the transferal effect of advances in meta heuristics methods and their application to single objective models that is subsequently extended into multi-objective models, and the growing awareness during the decade of the existence and importance of multiple objectives in various disciplines. [11] presents a Parameter Design (PD) approach that provides near-optimal settings to the process parameters of a single lathe machine with high-volume production. Optimized process parameters include both machining parameters (cutting speed, feed rate, and depth of cut) and production parameters (material order size and inventory safety stock and reorder point). This paper extends the conventional per-part machining cost model into a per order production cost model by consolidating the production economics of both machining parameters and production controls.

A series of versions of multi-objective optimizations algorithms have been developed. Comprehensive survey of several multi-objective optimization methods including useful and detailed discussion on their potential and limitations can be found in [12,13,14].

Among the versions we can enumerate the following approaches: Vector evaluated genetic algorithm (VEGA), Target vector approaches, Multi objective genetic algorithm (MOGA), Non-dominated sorting genetic algorithm (NSGA), Niched Pareto

Genetic Algorithm (NPGA) and Strength Pareto evolutionary algorithm (SPEA) among others. An improved version of the NSGA algorithm called NSGA-II, was proposed by DEB (2000). This improvement consists in the incorporation of elitism and parameter-free sharing approaches to the original algorithm.

3 Optimization Model

As it was mentioned above, the high efficiency cutting range is the interval of economic cutting speed (and/or tool feed rate). Within this range, there exists an optimal cutting speed that satisfies the economic restrictions and other speed that satisfies the production rate criteria. These two objectives functions are based on the well known extended Taylor´s tool life equation. Taylor Tool life equation is considered the first satisfactory correlation to model the wear process as a function of the cutting speed (eq.1).

$$K := T\cdot v^{x} \tag{1}$$

Lately, this first equation was extended to incorporate other cutting parameters, such as feed rate and depth of cut (eq.2).

$$K := T\cdot v^{x}\cdot f^{y}\cdot a^{z} \tag{2}$$

In the multi-objective optimization model these two main objectives are considered. The decision variables considered in this model are: cutting speed (v) and feed rate (f). Next, the mathematical equations are presented. Based on these equations, the production rate and production costs can be obtained. As it is expected, maximizing production rate corresponds to minimizing the cycle time of a cutting operation. Equation (3) express the total cycle time for one part.

$$t_t := \frac{\left(l_f\cdot\pi\cdot d\right)}{(1000\cdot f\cdot v)} + \left[t_s + t_a + \left(\frac{t_p}{N}\right)\right] + \left[\left[\frac{\left(l_f\cdot\pi\cdot d\cdot v^{x-1}\cdot f^{y-1}\cdot a\right)}{\left(1000\cdot K\right)}\right] - \left(\frac{1}{N}\right)\right]\cdot\left(t_{ft} - t_{fa}\right) \tag{3}$$

During a machining operation, the cost per workpiece can be expressed as it follows (eq.4):

$$K_p := C_1 + \left(\frac{\pi\cdot d\cdot l_f}{60\cdot 1000\cdot f\cdot v}\right)\cdot C_2 + \left[\frac{\left(\pi\cdot d\cdot l_f\cdot v^{x-1}\cdot f^{y-1}\cdot a^{z}\right)}{\left(1000\cdot K\right)}\right]\cdot C_3 \tag{4}$$

where:

$$C_2 := S_h + S_m$$

$$C_3 := K_{fT} + \left[\frac{\left(t_{ft} + t_{fa}\right)}{60}\right]\cdot\left(S_h + S_m\right)$$

and

d: Work piece diameter
lf: Length of cut
f: Feed rate
a: Depth of cut
Sh: Man-hour
Sm: Machine-hour
Kt: Cutting tool cost per life (T)
tft: Tool change time
ta: Time during which tool does not cut
ts: Tool installation time
tp: Machine tool set up time
N: Batch size
K: Taylor`s constant
x, y, z : Tool life equation Exponents

Figure 2 shows the relationship between decision variables and time and cost values respectively.

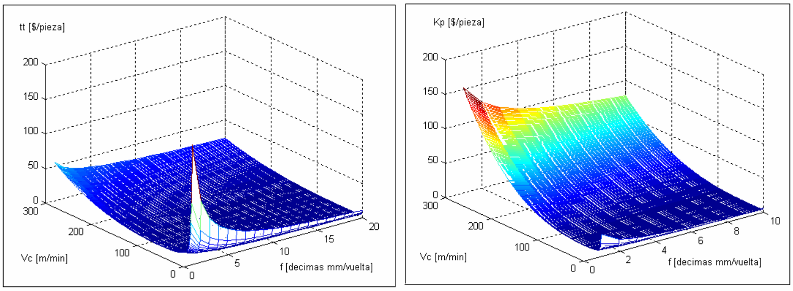


Fig. 2. Relation between feed rate, cutting speed and cycle time (a), between feed rate, cutting speed and cost per piece

4 Application Example and Results

Let us consider the turning operation of cast iron bar by means of a ISO K10 carbide tool. The expanded Taylor´s tool life equation exponents were obtained empirically by [15], and are shown as follows (Table 1). For comparison ends some simulation experiments were conducted. The data obtained by the simulation study were graphicated and processed for obtaining a good approximation of the minimum of each one of the two objective functions. Figure 2 represents solution space where the minimums are located. Figure 3 and figure 4 represent the time and costs curves as a function of cutting speed. As it was expected, the influence of the feed rate on the mentioned variables is lesser that the influence observed by the cutting speed. Those two minimums approximated through the use of the aforementioned simulation experiment, may be considered as the limits of the high efficiency range.

It can be observed that the Pareto front margins correspond to or are comparable to the limits of the high efficiency cutting range. In the graphic we can observe the minimum values of the cycle time per piece and the minimum cost per piece, which correspond to 1.8438 min/unit and $501/unit respectively. The used algorithm also presents the values of cutting speed and feed rate that correspond to the minimum values obtained. Table 2 presents these values.

Table 1. Summary of the data used by optimization process example

Work piece diameter	d = 90 [mm]
Length of cut	la = 80 [mm]
Feed rate	f = 0,3 [mm/rev]
Depth of cut	a = 1,5 [mm]
Man-hour	Sh = 6000 [$/h]
Machine-hour	Sm = 9000 [$/h]
Cutting tool cost per life (T)	Kft = 5000 [$]
Tool change time	tft =3 [min]
Time during which tool does not cut	ta = 0,1 [min/unit]
Tool installation time	ts = 1,5 [min/unit]
Machine tool set up time	tp = 15 [min]
Batch size	N = 200 [unit]
Taylor`s constant	K = 5,25*10^{12}
Tool life equation Exponents	x = 5,025
Exponente del avance	y = 0,91
Exponente de la profundidad de corte	z = 0,72

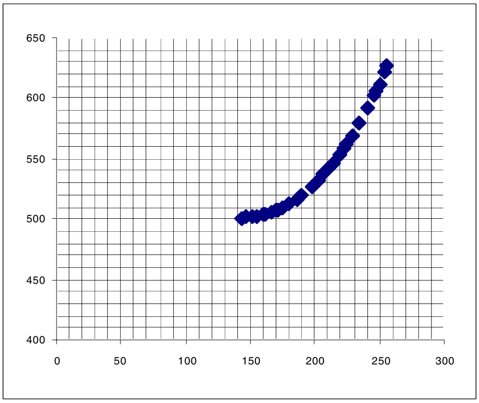


Fig. 3. Simulated values of work piece cost as a function of cutting speed

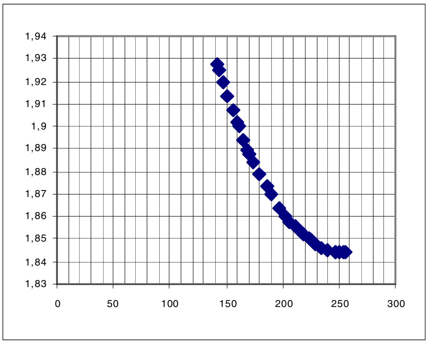


Fig. 4. Simulated values of cycle time as a function of cutting speed

Figure 5 shows the Pareto front obtained through the use of the NSGA-II algorithm.

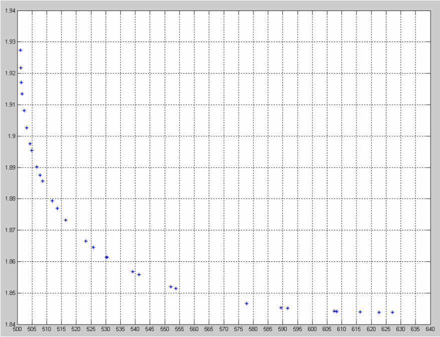


Fig. 5. Pareto front for the example

Table 2. Summary of the obtained as outcomes of the optimization process

Cutting Speed (m/min)	Feed rate (mm/rev)	Unit Cost ($/unit)	Cycle Time (min/unit)
256,06	0,6	627,08	1,8438
144,07	0,6	501,11	1,9273

5 Conclusions

For promoting adaptative capability of Automated Process Planning systems, an intelligent system for selecting the optimal cutting conditions was applied. This work points to the identification of economical cutting condition through the use of a multi-objective algorithm (NSGA-II). The use of the mentioned algorithm leads to the assumption that through the utilization of the multi-objective optimization approach one an obtain the limits of the cutting high efficiency range without the well known difficulties of the traditional approaches. Therefore, the selection of the appropriate cutting condition is possible in real world environments just using the metaheuristic approach. A variety of simulations were carried out to validate the performance of the approach and to show the usefulness of the applied algorithm. Further evolution of the system is possible. This evolution points to enhance the interconnectivity of the implemented algorithm with CAM systems. In addition, new tests are being performed for apply this approach in other types of cutting operations.

References

1. Tolouei-Rad M., Bidhendi I.M.: On the optimization of machining parameters for milling operations, Int. J. Mach. Tools Manuf. 37 (1) 1–16 (1997).
2. Wang J., Kuriyagawa T., Wei X.P., Guo D.M.: Optimization of cutting conditions for single pass turning operations using a deterministic approach.[International Journal of Machine Tools and Manufacture,42, ,1023-1033, (2002).
3. Armarego E.J.A., Smith A.J.R., Wang J.: Constrained optimization strategies and CAM software for single-pass peripheral milling, Int. J. Prod. Res. 31 (9) 2139–2160 (1993).
4. Taylor, F.W.: On the art of cutting metals. ASME Journal of Engineering for Industry 28, 310–350 (1906).
5. Hitomi, K.: Analyses of production models, Part 1: The optimal decision of production speeds. AIIE Transactions 8 (1), 96–105 (1976)
6. Taha, H.: A policy for determining the optimal cycle length for a cutting tool. Journal of Industrial Engineering 17 (3), 157–162.(1966).
7. Wang Z.G., Rahman M., Wong Y.S., Sun J.: Optimization of multi-pass milling using parallel genetic algorithm and parallel genetic simulated annealing. International Journal of Machine Tools & Manufacture 45 1726–1734 (2005).
8. Quiza Sardinas R., Rivas M. , Brindis E.A.: Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes, in Engineering Applications of Artificial Intelligence, 19(2):127-133, March (2006).
9. Konak, D., Coit W., Smith, A.E.: Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety, Vol. 91, No. 9, pp. 992--1007, September (2006)
10. Jones D.F., Mirrazavi S.K., Tamiz M.: Multi-objective metaheuristics: An overview of the current state-of-the-art. European Journal of Operational Research, Vol. 137, No. 1, pp. 1--9, February (2002).
11. Raid Al-Aomar, Ala'a Al-Okaily: A GA-based parameter design for single machine turning process with high-volume production. Computers & Industrial Engineering, Vol. 50, pp. 317–337 (2006).
12. Kicinger R., Arciszewski T., De Jong K.: Evolutionary computation and structural design: A survey of the state-of-the-art, Computers and Structures 83 1943–1978 (2005)
13. Coello Coello C.A., Van Veldhuizen DA, Lamont GB. Evolutionary algorithms for solving multi-objective problems. New York: Kluwer Academic (2002).
14. Deb K., Agrawal S., Pratap A., T Meyarivan: Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. Proceedings of the Parallel Problem Solving from Nature VI Conference (2000)
15. Consalter L., Arquivo de dados tecnológicos de usinagem para a determinação automática das condições automática das condições de corte em tornos com comando numérico. Msc Thesis, Universidade Federal de Santa Catarina, Florianópolis, Brasil (1985).

Evolutionary Computing for the Optimization of Mathematical Functions

Fevrier Valdez[1], Patricia Melin[2], and Oscar Castillo[2]

[1] PhD Student of Computer Science in the Universidad Autónoma de Baja California, Tijuana, B.C., México
fevrier@tectijuana.mx
[2] Computer Science in the Graduate Division, Tijuana Institute of Technology Tijuana, B.C.
pmelin@tectijuana.mx

Abstract. The Particle Swarm Optimization (PSO) and the Genetic Algorithms (GA) have been used successfully in solving problems of optimization with continuous and combinatorial search spaces. In this paper the results of the application of PSO and GAs for the optimization of mathematical functions is presented. These two methodologies have been implemented with the goal of making a comparison of their performance in solving complex optimization problems. This paper describes a comparison between a GA and PSO for the optimization of a complex mathematical function.

1 Introduction

We describe in this paper the application of a Genetic Algorithm (GA) [1] and Particle Swarm Optimization (PSO) [2] for the optimization of a mathematical function. In this case, we are using the Rastrigin's Function [4] to compare the optimization results between a Genetic Algorithm and Particle Swarm Optimization.

2 Genetic Algorithm for Optimization

John Holland, from the University of Michigan began his work on genetic algorithms at the beginning of the 1960s. His first achievement was the publication of *Adaptation in Natural and Artificial System* [7] in 1975.

Holland had two goals in mind: to improve the understanding of natural adaptation process, and to design artificial systems having properties similar to natural systems [8].

The basic idea is as follows: the genetic pool of a given population potentially contains the solution, or a better solution, to a given adaptive problem. This solution is not "active" because the genetic combination on which it relies is split between several subjects. Only the association of different genomes can lead to the solution.

Holland's method is especially effective because it not only considers the role of mutation, but it also uses genetic recombination, (crossover) [9]. The crossover of partial solutions greatly improves the capability of the algorithm to approach, and eventually find, the optimal solution.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 463–472, 2007.
springerlink.com

The essence of the GA in both theoretical and practical domains has been well demonstrated [1]. The concept of applying a GA to solve engineering problems is feasible and sound. However, despite the distinct advantages of a GA for solving complicated, constrained and multiobjetive functions where other techniques may have failed, the full power of the GA in application is yet to be exploited [12].

To bring out the best use of the GA, we should explore further the study of genetic characteristics so that we can fully understand that the GA is not merely a unique technique for solving engineering problems, but that it also fulfils its potential for tackling scientific deadlocks that, in the past, were considered impossible to solve. In figure 1 we show the reproduction cycle of the Genetic Algorithm.

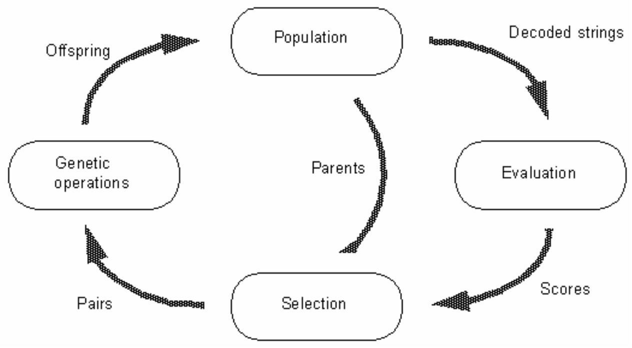


Fig. 1. The Reproduction cycle

The Simple Genetic Algorithm can expressed in pseudo code with the following cycle:

```
1. Generate the initial population of individuals aleatorily P(0).
2. While (number _ generations <= maximum _ numbers _ generations)
     Do:
       {
         Evaluation;
         Selection;
         Reproduction;
         Generation ++;
       }
3. Show results
   4. End of the generation
```

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling [3].

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA) [6]. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, the PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [10].

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far (The fitness value is also stored). This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p*best* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations.

In the past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods [11].

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

The pseudo code of the PSO is as follows

```
For each particle
   Initialize particle
END
Do
   For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   End
   Choose the particle with the best fitness value of all the particles as the gBest
   For each particle
      Calculate particle velocity
      Update particle position
   End
While maximum iterations or minimum error criteria is not attained.
```

4 Rastrigin's Function

For two independent variables, Rastrigin's function is defined in equation (1):

$$\mathrm{Ras}(x) = 20 + X_1^2 + X_2^2 - 10(\mathrm{Cos}2\pi X_1 + \mathrm{Cos}2\pi X_2) \tag{1}$$

Figure 2 shows a plot of Rastrigin's function [4] [5].

As this plot shows, Rastrigin's function has many local minimal [4] — the "valleys" in the plot. However, the function has just one global minimum, which occurs at the point [0 0] in the x-y plane, as indicated by the vertical line in the plot, where the value of the function is 0. At any local minimum other than [0 0], the value of Rastrigin's function is greater than 0. The further the local minimum is from the origin, the larger the value of the function is at that point. Rastrigin's function is often used to test the evolutionary computing methods, because its many local minimal make it difficult for standard, gradient-based methods to find the global minimum.

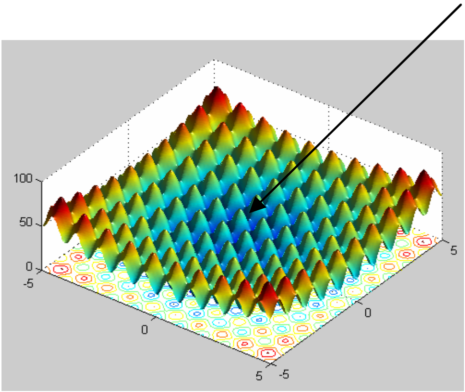


Fig. 2. Plot of the Rastrigin's function

5 Simulation Results

Several tests of the PSO and GA algorithms were made in the Matlab programming language. All the implementations were developed using a computer with processor AMD turion of 64 bits that works to a frequency of clock of 1800MHz, 512 MB of RAM Memory and Windows XP operating system.

5.1 Experimental Results with the Genetic Algorithm (GA)

The results obtained after applying the genetic algorithm to the Rastrigin's function are shown on table 1:

Parameters of Table 1:

No. = Number of experiment
TEST = Number of times that you executes the Genetic Algorithm with the same parameters
GEN = Generations number
POP= Population size
CROSS = Crossover type and % crossover
MUT = Mutation type and % mutation
BEST= Best Fitness Value
MEAN= Mean of 50 tests

Table 1. Results obtained after applying the genetic algorithm to the Rastrigin's function

GEN	POP	CROSS	% CROSS	MUT	% MUT	SEL	BEST	MEAN
50	20	ARITH	80	GAU	10	ROU	**1.92E-07**	0.333
80	50	TWO	60	UNI	5	UNI	1.64E-04	0.433
120	50	SCAT	75	GAU	15	TOU	0.004383493	0.59
120	60	SCAT	50	GAU	6	ROU	2.87E-05	**7.84E-03**
100	80	SCAT	90	GAU	9	ROU	0.015400193	0.0197

From Table 1 it can be appreciated that after running the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 1.92E-07, which is the shaded value in Table 1 (Experiment 1). The best average objective value was 7.84 E-03 obtained in Experiment 4.

In Figure 3 we show the convergence of the genetic algorithm, which is given by the best fitness and the average fitness of Experiment 1.

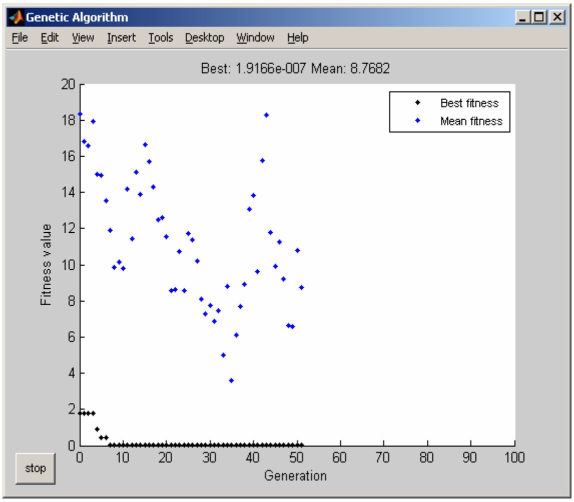
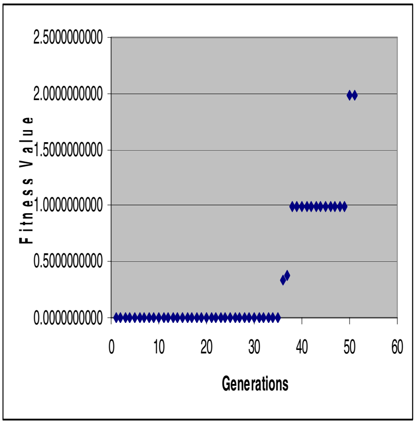


Fig. 3. Results of convergence of the genetic algorithm for the best test

Figure 4 and 5 show the results with the best fitness of the genetic algorithm for increasing number of generations in experiments 1 and 2.



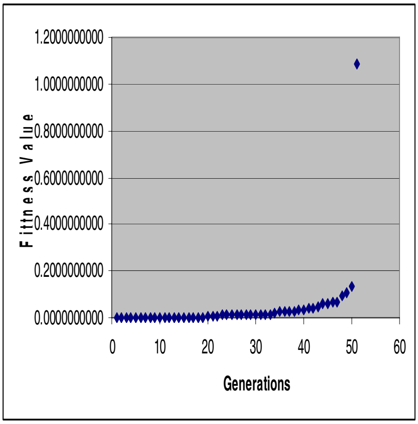
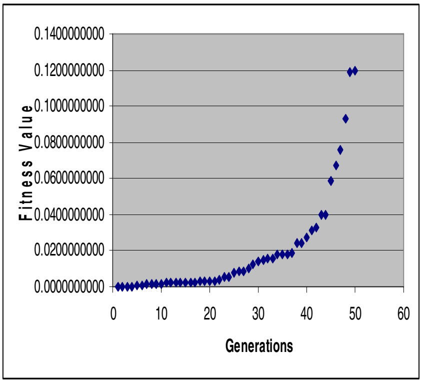


Fig. 4. Best fitness of the GA for experiment number 1 after 50 tests

Fig. 5. Best fitness of the GA for experiment number 2 after 50 tests

Figures 6 and 7 show the results of the best fitness for the genetic algorithm for increasing number of generations in experiments 3 and 4.



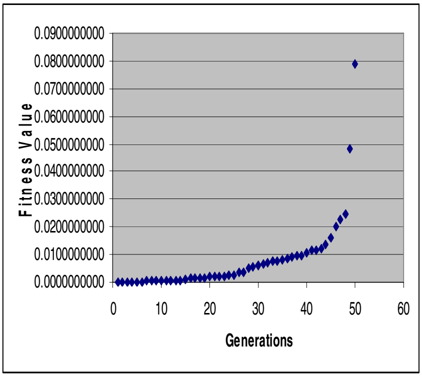


Fig. 6. Best fitness of the GA for experiment number 3 after 50 tests

Fig. 7. Best fitness of the GA for experiment number 4 after 50 tests

Figure 8 shows the comparison results between all the tests that were performed in this research with the genetic algorithm.

5.2 Results with Particle Swarm Optimization (PSO)

The results obtained after applying the particle swarm optimization to the Rastrigin's function are shown on table 2. The parameters shown in this table are described below.

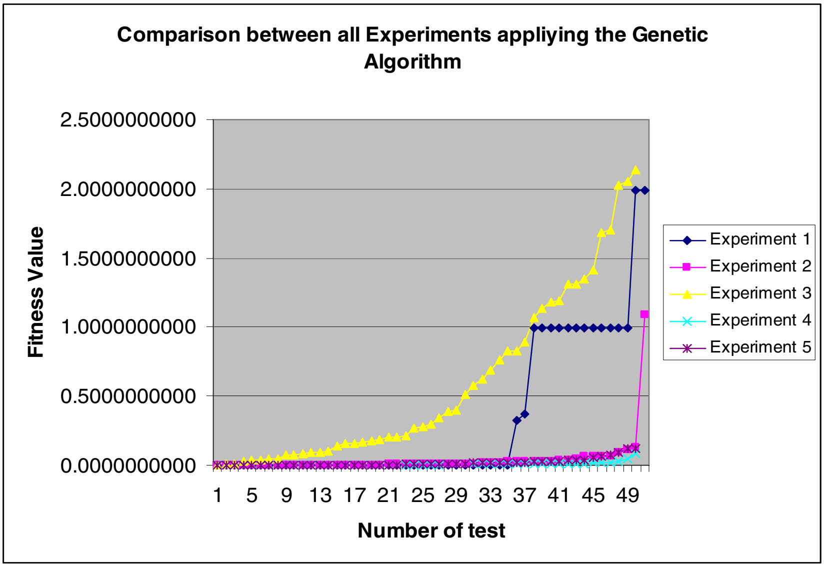


Fig. 8. Comparison results between all experiments with the GA

Parameters of Table 2:

No. = Number of experiment
TEST = Number of times that you executes the PSO with the same parameters
BEST = Best fitness value
DIM = Dimensions

Table 2. Results obtained after applying the PSO to the Rastrigin's function

No.	TEST	POP	DIM	BEST	MEAN
1	50	20	10	1.989918	6.872069
2	50	40	10	0.994961	4.057908
3	50	80	10	**0.000001**	**2.725607**
4	50	20	20	8.965743	25.979561
5	50	40	20	18.01619	19.265292

From Table 2 it can be appreciated that after running the PSO 50 times, only in 5 cases the global minimum was achieved with the best objective value at **0.000001**, which is the shaded value in Table 2 (Experiment 3). The best average objective value was **2.725607** obtained in Experiment 3.

Figures 9, 10, 11, 12 and 13 show the results of the best fitness of the particle swarm optimization in increasing form in the number of experiment 1, 2, 3, 4 and 5.

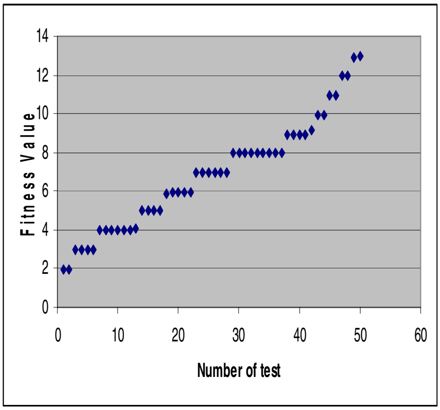


Fig. 9. Best fitness of the PSO for the number of experiment 1 for 50 tests.

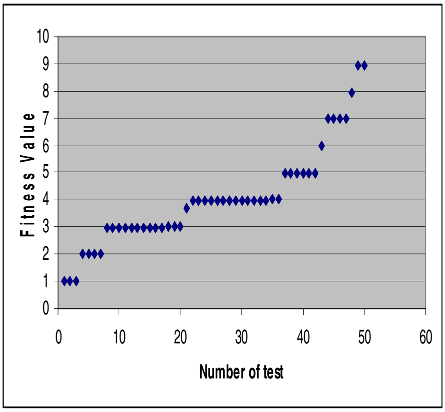


Fig. 10. Best fitness of the PSO for the number of experiment 2 for 50 tests

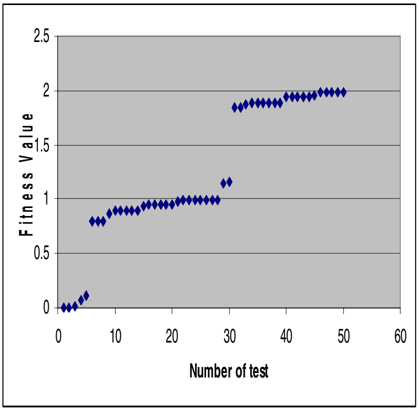


Fig. 11. Best fitness of the PSO for the number of experiment 3 for 50 tests.

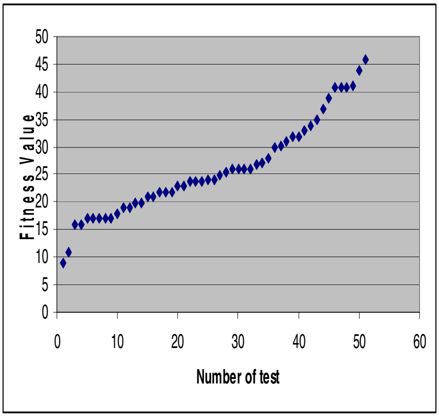


Fig. 12. Best fitness of the PSO for the number of experiment 4 for 50 tests.

Figure 14 shows the comparison results between all the tests that were developed in this investigation with the particle swarm optimization.

5.3 Comparison the Best Results Between the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)

The analysis of simulation results of the two evolutionary methods proposed in this paper, in this case the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO), lead us to the conclusion that for this problem of optimization of Rastrigin's function [4], the method with the better result was the genetic algorithm, although the PSO is also able to minimize the function, the best results were obtained with the GA. In all cases one can say that the two proposed methods work correctly and they can be applied for this type of problems.

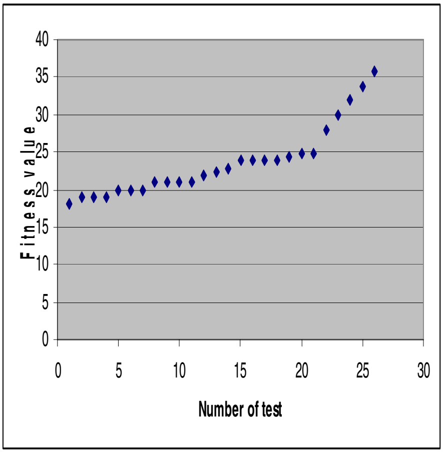


Fig. 13. Best fitness of the PSO for the number of experiment 5 for 50 tests

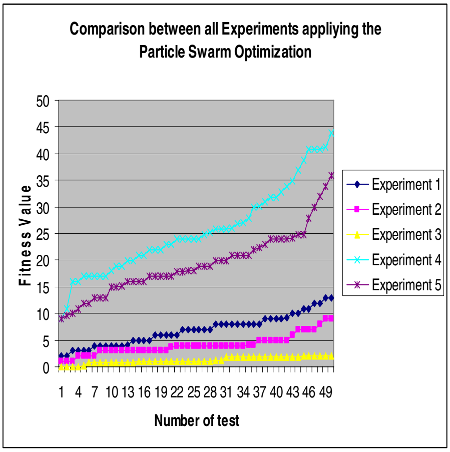


Fig. 14. Comparison results between all experiments with the PSO

6 Conclusions

After studying the two methods of evolutionary computing (GA and PSO), we reach the conclusion that for the optimization of Rastrigin's function the GA gives better results than the PSO, which can be appreciated in figure 14, in the comparison of the averages of these two methods, the GA obtains better results because stays very near the objective value in most of the tests, but if we observe Table 2 of results of the PSO, this method also achieves good. Figure 15 shows the results of the comparison between these two methods of evolutionary computing for the solution of the problem of finding the global minimum of the function above mentioned and you can observe that the best average of fitness value for the best solutions obtained it the genetic algorithm.

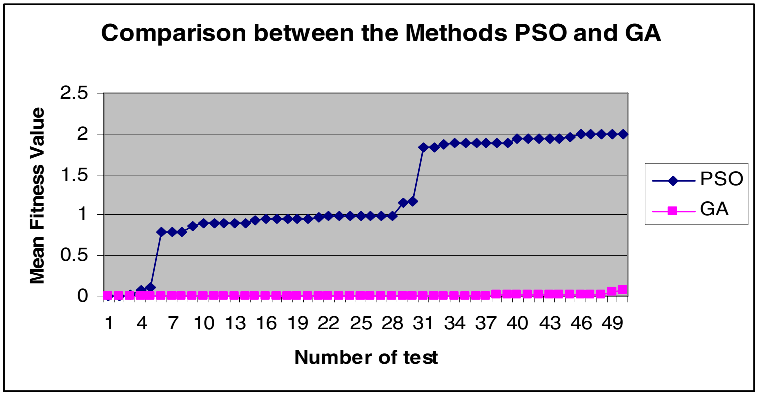


Fig. 15. Comparison results between PSO and GA

References

[1] K.F. Man, K.S. Tang, and S. Kwong (1999), "Genetic Algorithms: Concepts and Designs", Springer Verlag.

[2] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995

[3] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995

[4] Matlab Toolbox. www.mathworworks .com

[5] Germundsson, R. "Mathematica Version 4." Mathematica J. 7, 497-524, 2000.

[6] D.B. Fogel, "An introduction to simulated evolutionary optimization", IEEE transactions on neural networks, vol 5, n 1, jan 1994.

[7] Holland J.H., Adaptation in natural and artificial system, Ann Arbor, The University of Michigan Press, 1975.

[8] Goldberg D., Genetic Algorithms, Addison Wesley, 1988.

[9] Emmeche C., Garden in the Machine. The Emerging Science of Artificial Life, Princeton University Press, 1994, pp. 114.

[10] P. J. Angeline, Using Selection to Improve Particle Swarm Optimization , In Proceedings 1998 IEEE World Congress on Computational Intelligence, Anchorage, Alaska, IEEE, (1998): 84-89.

[11] P. J. Angeline, _Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences, Evolutionary Programming VII, Lecture Notes in Computer Science 1447, Springer, (1998): 601-610.

[12] T. Back, D. B. Fogel, and Z. Michalewicz, (Eds), Handbook of Evolutionary Computation, Oxford University Press, (1997).

Providing Intelligence to Evolutionary Computational Methods

Oscar Montiel[1], Oscar Castillo[2], Patricia Melin[2], and Roberto Sepúlveda[1]

[1] Centro de Investigación y Desarrollo de Tecnología Digital del Instituto
Politécnico Nacional (CITEDI-IPN) Av. del Parque No. 1310, Mesa de
Otay, Tijuana, B.C., México
o.montiel@ieee.org, r.sepulveda@ieee.org

[2] Department of Computer Science, Tijuana Institute of Technology P.O. Box 4207,
Chula Vista CA 91909, USA
ocastillo@galgo.tectijuana.mx, pmelin@tectijuana.mx

Abstract. It is presented an intelligent evolutionary method to solve single objective optimization problems, we called this method Single Objective Intelligent Evolutionary Algorithm (SO-IEA). This method uses several mechanisms that work synergistically to provide the optimal solution by handling in an intelligent way, the number of times that the objective function needs to be evaluated. The SO-IEA was subjected to several tests using complex benchmark functions and the results were statistically compared to other state of the art evolutionary algorithms (EA) obtaining that the SO-IEA outperformed in time and in precision the other methods. In general, the ideas presented here can be easily adapted to other EAs.

1 Introduction

There are several optimization methods that have been fully addressed in the literature [4,22] and they have several advantages and drawbacks. EAs are methods that have been successfully used for solving nonlinear optimization problems [14]; however, in any EA we have many parameters to adjust, and generally, they are adjusted by trial and error. Generally, they are adjusted one at time, since often it is unknown how they interact. Many different procedures have been researched to adapt the parameters. It is common that the parameters choice differ strongly from case to case, but the main idea is to no longer choose the parameters semi-arbitrarily but to let the parameters to auto adapt. Self-adaptation is a phenomenon, which makes EAs more flexible and closer to natural evolution. The Human Evolutionary Model is an intelligent global optimization method conceived to perform Single and Multiple Objective Optimization [21,23,24], this general method is still in development, especially the Multi Objective (MO) part is being improved. The Single Objective (SO) part has demonstrated that outperforms several algorithms that are in the state of the art, for example Differential Evolution (DE)[11,25], Particle Swarm Optimizer [1,6], and others. In this work we are presenting a novel intelligent method to perform SO optimization, we called this method SO Intelligent Evolutionary Algorithm (SO-IEA) since it uses an intelligent method based in human expertise to establish a fuzzy inference system with the purpose of making more efficient the exploration and exploitation of the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 473–481, 2007.
springerlink.com

landscape. The SO-IEA is part of the Human Evolutionary Model (HEM) [21], basically is the most basic part of HEM's Single Objective Optimization procedure. SO-IEA, uses Mediative Fuzzy Logic (MFL) for handling doubtful and contradictory information from experts to calculate the appropriated amount of individuals to create and/or to eliminate [19,20]. MFL, was proposed as an extension of traditional fuzzy logic [3,15,18] and includes Intuitionistic fuzzy Logic (IFL) in the Atanassov sense [15].

The paper is organized as follows: In section 2, we are giving a general definition for a Single Objective Intelligent Evolutionary Algorithm (SO- IEA). In section 3, we are describing the experiments that were achieved, and the corresponding results are discussed in section 4. Finally, in section 5 we have the conclusions.

2 SO-IEA Description

In general, we are defining a SO-IEA according to expression 1, where there are seven main components.

$$SO-IEA=\left(H,AIIS,P,O,S,L,E\right) \tag{1}$$

H represents the Human or group of Humans that should make the initial problem analysis of the specific problem to be solved, *H* should specify the corresponding fitness function and restrictions, also human(s) are in charge to decide weather to use the database knowledge that was previously established, or to update it.

The idea of the Adaptive Intelligent Intuitive System (*AIIS*) is to provide to the evolutionary process the most suitable parameters in order to advance in evolution. At present stage, *AIIS* adapts population size by creating and eliminating individuals of the actual population to exploit the landscape with very few individual to speed up the evolutionary process, but without getting trapped in local optima. This last part is due to the intelligent mechanism that *SO-IEA* has to become explorative when it is needed.

The population *P*, is given by *N* individuals, each individual p_i is defined as $p_i=(gr_i,ge_i,ov_i)$. In this definition, gr_i is the floating point genetic representation, i.e. where the decision variables of the problem to be solved are given. The variable ge_i is used to code some individual's genetic attributes, for example gender, actual age, maximal aged allowed, pheromone level, and others. The last part is the individual's objective value.

In *O* is specified the optimization goal, here we can program the termination criteria.

In general, *S* represents the evolutionary strategy as well as the operators that are available to use. The recombination and mutation operators are defined in this term.

The term *L* means landscape; hence it is included anything related with this topic. Here, it is given the function to be optimized, constrictions, as well as the fitness function.

The term *E* represents the Environment used to define operators that work synergistically with the operators defined in *S*. The aim of these operators is mainly focus to maintain the population in between desirable ranges of age, gender and values.

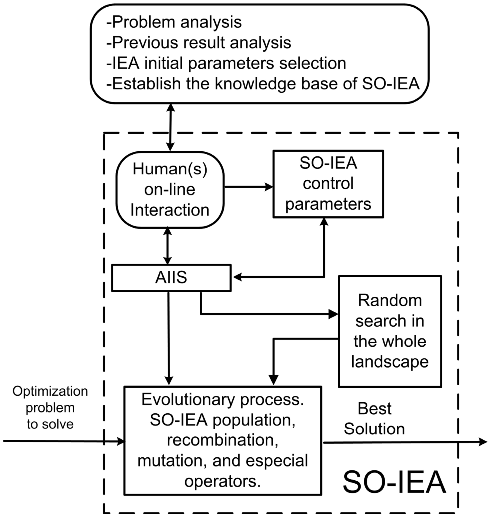

-Problem analysis
-Previous result analysis
-IEA initial parameters selection
-Establish the knowledge base of SO-IEA
Human(s) on-line Interaction
SO-IEA control parameters
AIIS
Random search in the whole landscape
Optimization problem to solve
Evolutionary process. SO-IEA population, recombination, mutation, and especial operators.
Best Solution
SO-IEA

Fig. 1. Block diagram of the SO-IEA

In Fig. 1, it is shown a general block diagram to describe in functional term the SO-IEA. The Human(s) is part of this model, always *H* can have on-line or off-line interaction with the evolutionary process either to improve the AIIS or to propose a different set of parameters to control the evolution via the "SO-IEA control parameters" block. The *AIIS* uses Mediative Fuzzy Logic (MFL) in order to establish a knowledge database to control population size and the parameters, the final idea is to make this unit to learn from the process. The evolutionary process is performed using floating point variables, and the Extended Intermediate Recombination (*EIR*) operator[7,8], and Discrete Mutation (*DM*) operator working over the population [7,8].

The *AIIS* will determine, based on the evolution performance, the amount of individuals to create and to eliminate. If *AIIS* determines that evolution is getting trapped in local optima, it will take a series of actions to avoid this situation. In Fig. 2 is shown in pseudo-code style the generic idea used to control population size. Basically, the MFL method in SO-IEA uses two variables. The variable "Variance" to control the variance value, and the variable "Cycling" to control the number of generations that the population remains in the same landscape region and/or without any significative change in the variance measurement. For the first one, we use the fitness value of the best individual of previous generations (we used 10 generations), to calculate their variance. For measuring the degree of cycling at each generation, we calculate the variable "Variance" and we make an increment in the variable "Cycling" each time the variable "Variance" is below a predetermined threshold value, we fixed this value to 0.05; otherwise reset the variable "Cycling", i.e. "Cycling=0".

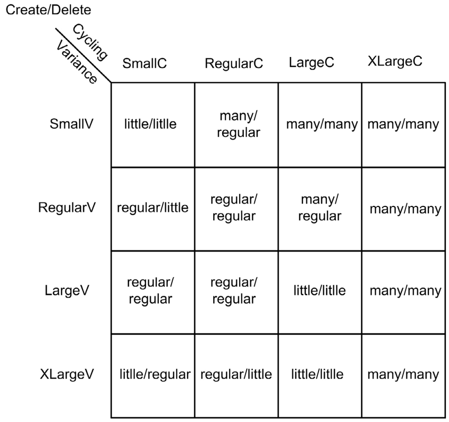
```
N=Initial population size;
ub=Upper bound;
lb=Lower bound;
maxGen=Maximum number of Generations;
actPs=N;
for gen=0 : maxGen

    Evolutionary process
    (Individuals selection, generate a new population of
    size actPs, fitness evaluation, operators aplication, etc.)

  if gen>11
      genBefore=gen-10;
      variance=var(Best_Fitness(genBefore:gen));
          If variance < 0.05
            Cycling=Cycling+1;
          else
            Cycling=0
          end
           . . .
           y_aiis=calculate_aiis_hem(Cycling,Variance);
          Create=round(y_aiis(1))*actPs;
          Delete=round(y_aiis(2))*actPs;

          while (actPS > lb)
              while (Delete)
                  Delete the individual with small fitness value;
                  Delete=Delete-1;
                  actPs=actPs-1;
              end
          end

          while (actPS < ub)
              while (Create )
                  Generete a new individual;
                   Create=Create-1;
                    actPs=actPs+1;
                end
          end
  end
```

Fig. 2. In the SO-IEA an intelligent system controls population size for optimizing the exploration and exploitation of landscape regions [20]

The threshold value was obtained experimentally, testing several complex test functions. The intelligent system will calculate the quantity of individuals that should be eliminated from the actual population, as well as the individuals that should be created in the actual population. Always, the operation of deletion is performed first, then we proceed to create the new individuals. We used two Sugeno inference systems of zero order, one for the agreement side, and the other one for the non-agreement side. In Fig. 3 we have the fuzzy associative memory for the agreement side, and in Fig. 4 the fuzzy assocative memory for the non-agreement side in an MFL system. we are summarizing the linguistic variables that were used in the fuzzy system. Each inference system has two output variables named “Delete” and “Create”. The values of these two variables are related to the amount of individuals that should be eliminated or created. Each output variable has three constant terms, which are: “little”, “regular”, and “many”. We assigned the values of “0”, “0.5”, and “1”, respectively. We used the same name and values at the output of each inference system (agreement and non-agreement).

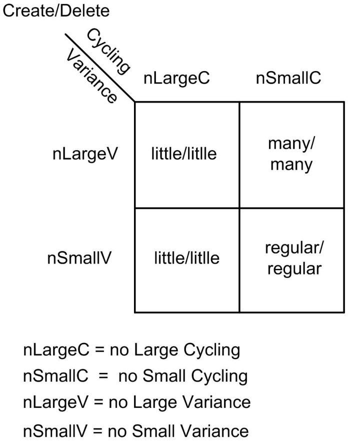
3 Experimental Results

The SO-IEA was used for all the tests. For achieving the recombination process, we selected a percentage of the best individuals (20%), and a percentage of individuals with highest pherhomone level (2%) (value of the genetic effect *ge*(*phLevel*) of each individual). The experiments were divided in two groups:



Create/Delete Cycling Variance	SmallC	RegularC	LargeC	XLargeC
SmallV	little/litlle	many/ regular	many/many	many/many
RegularV	regular/little	regular/ regular	many/ regular	many/many
LargeV	regular/ regular	regular/ regular	little/litlle	many/many
XLargeV	litlle/regular	regular/little	little/litlle	many/many

Fig. 3. Fuzzy associative memory for the variable “Agreement side”



Create/Delete Cycling Variance	nLargeC	nSmallC
nLargeV	little/litlle	many/ many
nSmallV	little/litlle	regular/ regular

nLargeC = no Large Cycling
nSmallC = no Small Cycling
nLargeV = no Large Variance
nSmallV = no Small Variance

Fig. 4. Fuzzy associative memory for the non-agreement side

Group 1. Experiments with Classical Test Functions

We performed experiments using the De Jong’s test function suite (F1-F5) [12], as well as the Rastrigin function (F6) [20]. We compared our results against the results obtained using a Genetic Algorithm (GA). We decided to use the Genetic Algorithm of Matlab’s Toolbox to perform the comparison because it provides an easy way to reproduce the results presented in this work. We made comparisons of processing time and average precision for each test function and for each method. Each test was conducted in the next form: For each test function (F1 to F6), we selected the most suitable parameters for the GA to achieve a good statistical result for 100 runs of the corresponding function. We executed the GA 100 times and we saved the best fitness value (maximal fitness), the minimum fitness value, and the statistical values of mean, median and standard deviation achieved in the 100 runs. We chose the parameters of SO-IEA in such a way to obtain a notable difference in precision in favor of SO-IEA. We used statistical t-Student test to ensure that statistical processes are different. The process was repeated for 2, 8, 16, and 32 variables. In all of the cases, we executed each algorithm 100 times for obtaining statistical meaningful results.

Group 2. Experiments with Composite Test Functions

We did experiments with composite test functions CF1 to CF6. These functions are highly complex to optimize. In [9] a method is given for constructing these Composite Functions. We considered as benchmark target values the results reported in [9] for the algorithms PSO, CPSO [5], CLPSO [10], CMA-ES [18], G3-PCX [13], and DE [26]. In this work the best objective value reported for the composite test function is 5.7348×10^{-8}, so we used that value as the stoping criteria for SO-IEA in this group of tests, a second criteria was the number of generations.

4 Discussion

In this section, a discussion of the performed experiments is presented.

For Group 1 (Classical test functions), the SO-IEA works better than the GA of the Matlab's Toolbox for 8, 16 and 32 variables. The best times and precision, finding the minimizer for this group of functions, were obtained using IEA. We noticed that the more difficult is the problem the better results were obtained using SO-IEA.

For Group 2 (Composite test functions), we compared the results obtained with the SO-IEA against the results reported in [9], acording to it, the CLPSO was the only optimization method that was able to find the minimizer of CF1, but fails with the other functions, CF2 to CF6. The methods PSO, CPSO, CMA-ES, G3-PCX, and DE failed finding the minimizers of all the functions CF1 to CF6. All the tests were made for 10 dimensions. It is known that these methods have demonstrated their excellent performance on standard benchmark functions; the problem was that the composite functions CF1 to CF6 are very complex.

On the other hand, with SO-IEA we had the following results:

1. For CF1 we obtained a success relation of 20/20, i.e. a success ratio of 100%, which means that the algorithm in 20 runs always found the minimizer. The average population size for these tests was 49 individuals.
2. In CF2 we had a success relation of 19/20, then we had a success ratio of 95%. The average population size was 475 individuals.
3. In CF3 we had a success relation of 5/20, then the success ratio is 25%.
4. CF4 was the most difficult function to optimize using HEM, we had a success relation of 0/20, but a remark is that comparatively we got better results than the other methods.
5. For CF5 we had a success relation of 4/20, so the success ratio is 20%. The average population size was 286 individuals.
6. For CF6 we had a success relation of 0/20. Only DE found a better mean value than HEM, the difference is minimal, and HEM has a smaller standard deviation. DE did not found the global minimizer.

For CF1 we obtained an average population of 49 individuals although the initial population size was 600 individuals with an upper and lower bounds of 600 and 100 individuals respectively. The SO-IEA and CLPSO founded the minimizer.

We executed the algorithms 300 generations for CF1 and CF3, and 500 generations for CF2, CF4, CF5, and CF6. The SO-IEA found solutions before the corresponding number of generations were achieved, for example for testing CF4, the best values were found around generation 50, so the time for finding this solution is more or less 200 seconds instead 1194 seconds. We have similar conditions in the other experiments.

5 Conclusions

The use of composite test functions was very important since these functions are hard to optimize, we had to reconsider and make a general review of each step of the SO-IEA of HEM as it was previously propossed in [20], for example, we were considering one operator for cloning the best individual in the population. This operator works great in our evolutionay model with classical test functions, but using the composite test function we realized that this operator was stucking the evolution.

The SO-IEA uses an intelligent strategy to "explore, exploit, ... ,exploit, explore,…", so we can say that the SO-IEA always searches for a promissory region for being exploited. AIIS will maintain or increase the population size; once a convenient region is found, generally the SO-IEA reduces its population size with the purpose of speeding up the algorithm, this is done taking care of no diminishing the performance of the evolution; and when AIIS determines that this region has been exploited enough, then AIIS decides to increase the population size by generating new individuals randomly, this is for being more explorative. In general, the SO-IEA has several mechanisms that work synergetically for performing search in an intelligent way. Hence, the SO-IEA is an intelligent evolutionary algorithm that uses knowledge from experts, and it is able to handle doubt and contradiction among human experts. An interesting thing that we observed is that almost always the dynamics of the population size, for the same type of problem behaves very similar.

In general, according to the comparisons that we made, we can conclude that SO-IEA outperformed the evolutionary algorithms PSO, CPSO, CLPSO, CMA-ES, G3-PCX and DE for the composite test functions CF1, CF2, CF3, CF5 and CF6.

We have demonstrated that having an intelligent evolutionary algorithm with the ability of learning from a group of experts is an interesting and promising idea. Future work may focus on improving the AIIS and increasing its knowledge data base to control more parameters of this evolutionary model.

References

1. P. Engelbrecht (2006) Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, U.S.A.
2. B.G.W. Craenen, A.E. Eiben (2006), Computational Intelligence, online document consulted in 2006 May 28, Encyclopedia of Life Support Sciences, EOLSS, EOLSS Co. Ltd., available at: http://www.xs4all.nl/~bcraenen/publications.html.
3. E. Coşkun (2000) Systems on intuitionistic fuzzy special sets and intuitionistic fuzzy special measures, Information Sciences 128 (2000) 105-118.
4. Edwin K. P. Chong and Stanislaw H. Żak (2001) An Introduction to Optimization, Second Edition. Ed. Wiley.

5. F. Van den Bergh, A. P. Engelbrecht (2004). A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation 8-3 (2004), 225-239.
6. F. Van den Bergh, A. P. Engelbrecht (2006) A study of particle swarm optimization particle trajectories, Information Sciences 176 (2006), 937-971.
7. H. Mühlenbein, D. Schilierkamp-Voosen (1993) Predictive Model for Breeder Genetic Algorithm. I. Continuous Parameter Optimization, Evolutionary Computation 1, pp. 25-49.
8. H. Mühlenbein, D. Schilierkamp-Voosen (1994) The science of breeding and its application to the breeder genetic algorithm BGA, Evolutionary Computation 1. pp. 335-360.
9. J. J. Liang, P. N. Suganthan and K. Deb (2005) Novel Composition Test Functions for Numerical Global Optimization, In: IEEE Swarm Intelligence Symposium, Pasadena California, U.S.A, pp. 68-75.
10. J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar (2004), Evaluation of Comprehensive Learning Particle Swarm Optimizer, Lecture Notes in Computer Science 3316, pp. 230-235.
11. J. Sun, Q. Zhang, and E. P.K. Tzang (2005), DE/EDA: A new evolutionary algorithm for global optimization, Information Sciences 169, pp. 249-262.
12. K. A. De Jong (1975) An analysis of the behaviour of a class of genetic adaptive systems, Ph.D. Thesis in Computer and Comunication Science, University of Michigan.
13. K. Deb, A. Anand, and D. Joshi (2002) A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization, KanGAL Report No. 20022003, Available at: http://www.iitk.ac.in/kangal/reports.shtml#2002.
14. Kalyanmoy Deb (2002) Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, LTD. Great Bretain.
15. K. T. Atanassov (1999) Intuitionistic Fuzzy Sets: Theory and Applications, Springer-Verlag, Heidelberg, Germany.
16. L. Belanche (1999) A Study in Function Optimization with the Breeder Genetic Algorithm, Research Report LSI-99-36-R. Universitad Politécnica de Catalunya, Available at: http://citeseer.nj.nec.com/belanche99study.html
17. N. Hansen, S. D. Muller and P. Koumoutaskos (2003) Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), Evolutionary Computation 11, pp. 1-18.
18. O. Castillo, P. Melin (2003) A New Method for Fuzzy Inference, In: Intuitionistic Fuzzy Systems, Proceedings of the International Conference NAFIPS 2003, IEEE Press, Chicago, Illinois, USA. pp. 20-25.
19. O. Montiel, O. Castillo, P. Melin, R. Sepúlveda (2005) Reducing the cycling problem in evolutionary algorithm, In: The 2005 International Conference on Artificial Intelligence (IC-AI'05), Las Vegas Nevada, USA. pp. 426-432.
20. O. Montiel, O. Castillo, P. Melin, R. Sepúlveda (2007). Mediative Fuzzy Logic: A Novel Approach For Handling Contradictory Knowledge, Studies in Fuzziness and Soft Computing, Hybrid Intellignt Systems, Springer-Verlag, USA.
21. O. Montiel et al., (2006). Human evolutionary model: A new approach to optimization, Informat. Sci. (2006), doi:10.1016/j.ins.2006.09.012.
22. Oliver Nelles (2001). Nonlinear System Identification. From Classical Approaches to Neural networks and Fuzzy Models. Springer-Verlag Berlin Heidelberg. Germany.
23. Oscar Montiel, Oscar Castillo, Patricia Melin, Roberto Sepúlveda (2007). Improving the Human Evolutionary Model: An intelligent optimization method. International Mathematical Forum. Journal for Theory and Applications. Vol. 2, no. 1-4.

24. Oscar Montiel, Oscar Castillo, Patricia Melin, Roberto Sepúlveda (2006). An Experimental Comparison between the Human Evolutionary Model and the Particle Swarm Optimizer Model. Proceedings of International Seminar on Computational Intelligence ISCI 2006. Tijuana, B. C., Mex.
25. R. Storm, and K. Price (1997) Differential evolution – A simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization 11, 341-359.

Part VII

Fuzzy Modeling

Representing Fuzzy Numbers for Fuzzy Calculus

Luciano Stefanini and Laerte Sorini

University of Urbino "Carlo Bo", Urbino, Italy
LucSte@Uniurb.it, Laerte@UniUrb.it

Abstract. In this paper we illustrate the LU representation of fuzzy numbers and present an LU-fuzzy calculator, in order to explain the use of the LU-fuzzy model and to show the advantage of the parametrization. The model can be applied either in the level-cut or in generalized LR frames. The hand-like fuzzy calculator has been developed for the MS-Windows platform and produces the basic fuzzy calculus: the arithmetic operations (scalar multiplication, addition, subtraction, multiplication, division) and the fuzzy extension of many univariate functions (*exponential, logarithm, power with numeric or fuzzy exponent, sin, arcsin, cos, arccos, tan, arctan, square root, Gaussian, hyperbolic sinh, cosh, tanh and inverses, erf and erfc error* functions, *cumulative standard normal* distribution).

1 Introduction

The arithmetic operations on fuzzy numbers are usually approached either by the use of the extension principle (in the domain of the membership function, [8]) or by the interval arithmetics (in the domain of the $\alpha - cuts$) as outlined by Dubois and Prade ([1]); the same authors have introduced the well known LR model and the corresponding formulas for the fuzzy operations ([2]); an extensive survey and bibliography is in [3]. In [4], the use of monotonic splines is suggested to approximate fuzzy numbers, using several interpolation forms and a procedure is described to control the error of the approximation. The parametric LU representation allows a large set of possible shapes (types of membership functions) that seems to be much wider than the well-known LR framework (see also [6] and [7]).

The paper is organized as follows: in sections 2 and 3 we describe the LU-fuzzy model and calculus and some example algorithms which implement the LU-fuzzy extension principle for unidimensional elementary functions. Section 4 contains a description of the LU-fuzzy calculator.

1.1 Basic Fuzzy Calculus

We adopt the so called $a - cut$ setting for the definition of a fuzzy number:

Definition. *A continuous fuzzy number (or interval) u is any pair* $\left(u^{-}, u^{+}\right)$ *of functions* $u^{\pm} : [0,1] \rightarrow \mathbb{R}$ *satisfying the following conditions:(i)* $u^{-} : \alpha \rightarrow u_{\alpha}^{-} \in \mathbb{R}$ *is a bounded monotonic increasing (non decreasing) continuous function* $\forall \alpha \in [0,1]$;

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 485–494, 2007.
springerlink.com

(ii) $u^+ : \alpha \to u_\alpha^+ \in \mathbb{R}$ *is a bounded monotonic decreasing (non increasing) continuous function* $\forall \alpha \in [0,1]$; *(iii)* $u_\alpha^- \le u_\alpha^+ \;\; \forall \alpha \in [0,1]$.

The notation $u_\alpha = \left[u_\alpha^-, u_\alpha^+\right]$ is used explicitly for the $\alpha - cuts$ of *u*. We will also refer to u^- and u^+ as the lower and the upper branches of *u*, respectively. If $u = \left(u^-, u^+\right)$ and $v = \left(v^-, v^+\right)$ are given fuzzy numbers, the interval-based arithmetic operations are defined in the usual way, for $\alpha \in [0,1]$:

(*Addition*) $\left(u+v\right)_\alpha = \left[u_\alpha^- + v_\alpha^-, u_\alpha^+ + v_\alpha^+\right]$.

(*Scalar Multiplication*) For $k \in \mathbb{R}$, $\left(ku\right)_\alpha = \left[\min\left\{ku_\alpha^-, ku_\alpha^+\right\}, \max\left\{ku_\alpha^-, ku_\alpha^+\right\}\right]$.

(*Subtraction*) $\left(u-v\right)_\alpha = \left[u_\alpha^- - v_\alpha^+, u_\alpha^+ - v_\alpha^-\right]$.

(*Multiplication*) $\begin{cases} \left(uv\right)_\alpha^- = \min\left\{u_\alpha^- v_\alpha^-, u_\alpha^- v_\alpha^+, u_\alpha^+ v_\alpha^-, u_\alpha^+ v_\alpha^+\right\} \\ \left(uv\right)_\alpha^+ = \max\left\{u_\alpha^- v_\alpha^-, u_\alpha^- v_\alpha^+, u_\alpha^+ v_\alpha^-, u_\alpha^+ v_\alpha^+\right\} \end{cases}$.

(*Division*) If $0 \notin \left[v_0^-, v_0^+\right]$, $\begin{cases} \left(\frac{u}{v}\right)_\alpha^- = \min\left\{\frac{u_\alpha^-}{v_\alpha^-}, \frac{u_\alpha^-}{v_\alpha^+}, \frac{u_\alpha^+}{v_\alpha^-}, \frac{u_\alpha^+}{v_\alpha^+}\right\} \\ \left(\frac{u}{v}\right)_\alpha^+ = \max\left\{\frac{u_\alpha^-}{v_\alpha^-}, \frac{u_\alpha^-}{v_\alpha^+}, \frac{u_\alpha^+}{v_\alpha^-}, \frac{u_\alpha^+}{v_\alpha^+}\right\} \end{cases}$.

2 LU-Fuzzy Representation and Calculus

The parametric LU representation of a fuzzy number is defined on a decomposition of the interval [0,1], $0 = \alpha_0 < \alpha_1 < ..... < \alpha_{i-1} < \alpha_i < .... < \alpha_N = 1$ for both the lower $u^-(\alpha)$ and the upper $u^+(\alpha)$ branches. In each of the *N* subintervals $I_i = [\alpha_{i-1}, \alpha_i]$, *i=1,…,N,* the values of the two functions $u^-(\alpha_{i-1}) = u_{0,i}^-$, $u^+(\alpha_{i-1}) = u_{0,i}^+$, $u^-(\alpha_i) = u_{1,i}^-$, $u^+(\alpha_i) = u_{1,i}^+$ and their first derivatives $u'^-(\alpha_{i-1}) = d_{0,i}^-$, $u'^+(\alpha_{i-1}) = d_{0,i}^+$, $u'^-(\alpha_i) = d_{1,i}^-$, $u'^+(\alpha_i) = d_{1,i}^+$ are assumed to be known; we are interested in families of monotonic functions that satisfy the above eight Hermite-type conditions for each subinterval I_i. In general, by transforming each subinterval I_i into the standard [0,1] interval, i.e. $t_\alpha = \frac{\alpha - \alpha_{i-1}}{\alpha_i - \alpha_{i-1}}$, $\alpha \in I_i$, we can determine each piece independently and obtain piecewise continuous LU-fuzzy numbers. Globally continuous or more regular $C^{(1)}$ fuzzy numbers can be obtained directly from the data if the following conditions are met for the values and possibly for the slopes:

$$u_{1,i}^- = u_{0,i+1}^-,\; u_{1,i}^+ = u_{0,i+1}^+,\; d_{1,i}^- = d_{0,i+1}^-,\; d_{1,i}^+ = d_{0,i+1}^+,\; \text{for } i = 1,2,...,N-1.$$

Let $p_i\left(t_\alpha\right)$ be a model function for u_α on a generic subinterval I_i; then, for $t_\alpha \in [0,1]$ we have

$$\begin{aligned} p_i\left(t_\alpha\right) &= u(\alpha_{i-1} + t_\alpha(\alpha_i - \alpha_{i-1})) \\ p_i'\left(t_\alpha\right) &= u'(\alpha_{i-1} + t_\alpha(\alpha_i - \alpha_{i-1}))(\alpha_i - \alpha_{i-1}). \end{aligned} \tag{1}$$

Proposed $p\left(t\right)$ functions are the *(2,2)-rational monotonic spline*

$$p\left(t\right) = \begin{cases} \frac{P(t)}{Q(t)} & \text{if } u_1 \neq u_0 \\ u_0 & \text{if } u_1 = u_0 \end{cases}, \text{ where}$$

$$P\left(t\right) = \left(u_1 - u_0\right)u_1 t^2 + \left(u_0 d_1 + u_1 d_0\right)t(1-t) + \left(u_1 - u_0\right)u_0\left(1-t\right)^2$$

$$Q(t) = \left(u_1 - u_0\right)t^2 + \left(d_1 + d_0\right)t(1-t) + \left(u_1 - u_0\right)\left(1-t\right)^2;$$

the *(3,2)-rational monotonic spline*

$$p\left(t\right) = \frac{P(t)}{Q(t)} \text{ with}$$

$$P(t) = u_0\left(1-t\right)^3 + \left(wu_0 + d_0\right)t\left(1-t\right)^2 + \left(wu_1 - d_1\right)t^2\left(1-t\right) + u_1 t^3$$

$$Q(t) = 1 + t\left(1-t\right)\left(w-3\right)$$

$$w = \frac{d_0 + d_1}{u_1 - u_0} \text{ to have monotonicity;}$$

and the *monotonic mixed cubic-exponential spline*

$$p(t) = u_0 + (u_1 - u_0 - \frac{d_0 + d_1}{a})t^2(3-2t) + \frac{d_0}{a} - \frac{d_0}{a}(1-t)^a + \frac{d_1}{a}t^a$$

$$a = 1 + \frac{d_0 + d_1}{u_1 - u_0} \text{ to have monotonicity.}$$

The models include linear (i.e. triangular fuzzy numbers), monotonic quadratic and monotonic cubic polynomials as special cases.

Using one of the previous forms to represent the lower and the upper branches of the fuzzy number $u = (u^-, u^+)$ we can write the general form of the representation (the symbol δ is used to denote the slopes or first derivatives)

$$\begin{array}{rcl} u & = & (u_{0,i}^-, \delta u_{0,i}^-, u_{1,i}^-, \delta u_{1,i}^-; u_{0,i}^+, \delta u_{0,i}^+, u_{1,i}^+, \delta u_{1,i}^+)_{i=1,\ldots,N} \\ & \Updownarrow & \\ u_\alpha & = & [p_i(t_\alpha; u_{0,i}^-, \widehat{\delta u}_{0,i}^-, u_{1,i}^-, \widehat{\delta u}_{1,i}^-), p_i(t_\alpha; u_{0,i}^+, \widehat{\delta u}_{0,i}^+, u_{1,i}^+, \widehat{\delta u}_{1,i}^+)]_{i=1,2,\ldots,N} \end{array} \tag{2}$$

with $\widehat{\delta u}_{k,i} = \delta u_{k,i}(\alpha_i - \alpha_{i-1})$, $k = 0,1$. For $N \geq 1$ we have a total of $8N$ parameters $u_{0,1}^- \leq u_{1,1}^- \ldots \leq u_{0,2}^- \leq u_{1,2}^- \leq \ldots \leq u_{0,N}^- \leq u_{1,N}^-$, $\delta u_{k,i}^- \geq 0$ defining the increasing lower branch u_α^- and $u_{0,1}^+ \geq u_{1,1}^+ \ldots \geq u_{0,2}^+ \geq u_{1,2}^+ \geq \ldots \geq u_{0,N}^+ \geq u_{1,N}^+$, $\delta u_{k,i}^+ \leq 0$ defining the decreasing upper branch u_α^+ (obviously, also $u_{1,N}^- \leq u_{1,N}^+$ is required).

A simplification of (2) can be obtained by requiring differentiable branches: $u_{1,i}^- = u_{0,i+1}^-$, $u_{1,i}^+ = u_{0,i+1}^+$ and $\delta u_{k,i}^- = \delta u_{k,i+1}^-$, $\delta u_{k,i}^+ = \delta u_{k,i+1}^+$. The number of parameters is reduced to $4N + 4$ and we can write

$$u=(u_i^-,\delta u_i^-,u_i^+,\delta u_i^+)_{i=0,1,\dots,N} \text{ with the data} \tag{3a}$$

$$\begin{aligned} &u_0^- \le u_1^- \le \dots \le u_N^- \le u_N^+ \le u_{N-1}^+ \le \dots \le u_0^+ \text{ and the slopes} \\ &\delta u_i^- \ge 0, \delta u_i^+ \le 0. \end{aligned} \tag{3b}$$

3 Arithmetic Operations

Given $u=(u_i^-,\delta u_i^-,u_i^+,\delta u_i^+)_{i=0,1,\dots,N}$ and $v=(v_i^-,\delta v_i^-,v_i^+,\delta v_i^+)_{i=0,1,\dots,N}$, the arithmetic operators associated to the LU representation can be obtained easily.

$$\{u+v=(u_i^-+v_i^-,\delta u_i^-+\delta v_i^-,u_i^++v_i^+,\delta u_i^++\delta v_i^+)_{i=0,1,\dots,N}$$

$$\begin{cases} ku=(ku_i^-,k\delta u_i^-,ku_i^+,k\delta u_i^+)_{i=0,1,\dots,N}, & \text{if } k\ge 0 \\ ku=(ku_i^+,k\delta u_i^+,ku_i^-,k\delta u_i^-)_{i=0,1,\dots,N}, & \text{if } k<0 \end{cases}$$

$$\{u-v=u+(-v)$$

$$\begin{cases} w=uv=\left(w_i^-,\delta w_i^-,w_i^+,\delta w_i^+\right)_{i=0,1,\dots,N} \text{ with} \\ w_i^-=\min\{u_i^-v_i^-,u_i^-v_i^+,u_i^+v_i^-,u_i^+v_i^+\} \\ w_i^+=\max\{u_i^-v_i^-,u_i^-v_i^+,u_i^+v_i^-,u_i^+v_i^+\} \\ w_i^-=u_i^{p_i^-}v_i^{q_i^-} \text{ and } w_i^+=u_i^{p_i^+}v_i^{q_i^+} \\ \delta w_i^-=\delta u_i^{p_i^-}v_i^{q_i^-}+u_i^{p_i^-}\delta v_i^{q_i^-}, \ \delta w_i^+=\delta u_i^{p_i^+}v_i^{q_i^+}+u_i^{p_i^+}\delta w_i^{q_i^+} \end{cases}$$

$$\begin{cases} z=u/v=\left(z_i^-,\delta z_i^-,z_i^+,\delta z_i^+\right)_{i=0,1,\dots,N} \text{ with} \\ (u/v)_i^-=\min\{u_i^-/v_i^-,u_i^-/v_i^+,u_i^+/v_i^-,u_i^+/v_i^+\} \\ (u/v)_i^+=\max\{u_i^-/v_i^-,u_i^-/v_i^+,u_i^+/v_i^-,u_i^+/v_i^+\} \\ z_i^-=u_i^{r_i^-}/v_i^{s_i^-} \text{ and } z_i^+=u_i^{r_i^+}/v_i^{s_i^+} \\ \delta z_i^-=(\delta u_i^{r_i^-}v_i^{s_i^-}-u_i^{r_i^-}\delta v_i^{s_i^-})/(v_i^{s_i^-})^2, \ \delta z_i^+=(\delta u_i^{r_i^+}v_i^{s_i^+}-u_i^{r_i^+}\delta v_i^{s_i^+})/(v_i^{s_i^+})^2 \end{cases}$$

where, for the *multiplication*, (p_i^-,q_i^-) is the pair of superscripts $+$ and $-$ giving the minimum $(uv)_i^-$ and similarly (p_i^+,q_i^+) is the pair of $+$ and $-$ giving the maximum $(uv)_i^+$; analogous symbols can be deduced for the *division,* (r_i^-,s_i^-) is the pair of $+$ and $-$ giving the minimum in $(u/v)_i^-$ and (r_i^+,s_i^+) is the pair of $+$ and $-$ giving the maximum in $(u/v)_i^+$.

As pointed out by the results of the experimentation reported in [4], the operations above are exact at the nodes α_i of the representation and have very small global errors on $[0,1]$. Further, it is easy to control the error by introducing additional nodes into the representation or by using a sufficiently high number of nodes with $\max\{\alpha_i-\alpha_{i-1}\}$ sufficiently small. To control the error of the approximation, we can proceed by increasing the number $N+1$ of points; a possible strategy is to double the number of points by using $N=2^K$ and by moving automatically to $N=2^{K+1}$ if a better precision is necessary.

The results in [4] of the parametric operators have shown that both the rational and the mixed models perform well with small $N \le 4$, with a percentage average error for a single multiplication and division of the order of 0.1%.

3.1 Fuzzy Extension of Univariate Functions

The fuzzy extension of a single (real) variable (differentiable) function $f:\mathbb{R}\to\mathbb{R}$ to a fuzzy argument $u_\alpha=\left[u_\alpha^-,u_\alpha^+\right]$ has $\alpha-cuts$

$f(u)_\alpha=\left[\min\{f(x)\,|\,x\in u_\alpha\},\max\{f(x)\,|\,x\in u_\alpha\}\right]$.

If f is monotonic increasing we obtain $f(u)_\alpha=\left[f\left(u_\alpha^-\right),f\left(u_\alpha^+\right)\right]$ while, if f is monotonic decreasing, $f(u)_\alpha=\left[f\left(u_\alpha^+\right),f\left(u_\alpha^-\right)\right]$.

Let X be the LU-fuzzy number $X=\left(x_i^-,\delta x_i^-,x_i^+,\delta x_i^+\right)_{i=0,1,\dots,N}$; then its image *Y*= *f(X)*= $\left(y_i^-,\delta y_i^-,y_i^+,\delta y_i^+\right)_{i=0,1,\dots N}$ is calculated as follows: let $\overset{\frown}{x}_i^- \in [x_i^-,x_i^+]$ and $\overset{\frown}{x}_i^+ \in [x_i^-,x_i^+]$ be where $\min\{f(x)\,|\,x\in[x_i^-,x_i^+]\}$ and $\max\{f(x)\,|\,x\in[x_i^-,x_i^+]\}$ are attained; possibly, $\overset{\frown}{x}_i^-$, $\overset{\frown}{x}_i^+$ are one of the extremes x_i^-,x_i^+ of the interval or may be internal points (where the derivative of f is zero). We then have

$$y_i^- = f(\overset{\frown}{x}_i^-)$$
$$y_i^+ = f(\overset{\frown}{x}_i^+)$$

$$\delta y_i^- = \begin{cases} f'(\overset{\frown}{x}_i^-)\delta x_i^- & \text{if } \overset{\frown}{x}_i^- = x_i^- \text{ is the left extreme point of the interval} \\ f'(\overset{\frown}{x}_i^-)\delta x_i^+ & \text{if } \overset{\frown}{x}_i^- = x_i^+ \text{ is the right extreme point of the interval} \\ 0 & \text{if } \overset{\frown}{x}_i^- \in]x_i^-,x_i^+[\text{ is an internal point} \end{cases}$$

$$\delta y_i^+ = \begin{cases} f'(\overset{\frown}{x}_i^+)\delta x_i^- & \text{if } \overset{\frown}{x}_i^+ = x_i^- \text{ is the left extreme point of the interval} \\ f'(\overset{\frown}{x}_i^+)\delta x_i^+ & \text{if } \overset{\frown}{x}_i^+ = x_i^+ \text{ is the right extreme point of the interval} \\ 0 & \text{if } \overset{\frown}{x}_i^+ \in]x_i^-,x_i^+[\text{ is an internal point} \end{cases}$$

Example 1: Fuzzy Extension of Hyperbolic Cosinusoidal Function
Let

$$Y=\cosh(X)=\frac{e^X+e^{-X}}{2}.$$

For each $i=0,1,\dots,N$:

if $X_i^+ \le 0$ then $\begin{cases} Y_i^- = \cosh\left(X_i^+\right) \\ Y_i^+ = \cosh\left(X_i^-\right) \\ \delta Y_i^- = \delta X_i^+ \sinh\left(X_i^+\right) \\ \delta Y_i^+ = \delta X_i^- \sinh\left(X_i^-\right) \end{cases}$	else if $X_i^+ \le 0$ then $\begin{cases} Y_i^- = \cosh\left(X_i^-\right) \\ Y_i^+ = \cosh\left(X_i^+\right) \\ \delta Y_i^- = \delta X_i^- \sinh\left(X_i^-\right) \\ \delta Y_i^+ = \delta X_i^+ \sinh\left(X_i^+\right) \end{cases}$
	else $\begin{cases} Y_i^- = 1\ ,\ \ \delta Y_i^- = 0 \\ \text{if } abs(X_i^-) \ge abs(X_i^+) \quad \text{then}\begin{cases} Y_i^+ = \cosh\left(X_i^-\right) \\ \delta Y_i^+ = \delta X_i^- \sinh\left(X_i^-\right) \end{cases} \\ \qquad\qquad\qquad\qquad\qquad\quad \text{else}\begin{cases} Y_i^+ = \cosh\left(X_i^+\right) \\ \delta Y_i^+ = \delta X_i^+ \sinh\left(X_i^+\right) \end{cases} \end{cases}$

Example 2: Fuzzy Extension of *erf* and *erfc* Error Functions

Let

$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-t^2\right) dt = 1 - \operatorname{erfc}(x)$ (increasing)
$\operatorname{erfc}\left(x\right) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} \exp\left(-t^2\right) dt$ (decreasing).

We use the following approximation, having a fractional error less than 1.2×10^{-7} :

$$z = abs\left(x\right),\ t = \frac{1}{1+\frac{1}{2}z}$$

$$\operatorname{erfc} = \begin{cases} t\exp\left(-z^2 + p\left(t\right)\right) & \textit{if } x \ge 0 \\ 2 - t\exp\left(-z^2 + p\left(t\right)\right) & \textit{if } x < 0 \end{cases}$$

with

$$p\left(t\right) = a_0 + t\Big(a_1 + t\Big(a_2 + t\Big(a_3 + t\Big(a_4 + t\Big(a_5 + t\Big(a_6 + t\left(a_7 + t\left(a_8 + ta_9\right)\right)\Big)\Big)\Big)\Big)\Big)\Big)$$

and

$a_0 = -1.26551223$ $a_1 = 1.00002368$ $a_2 = 0.37409196$
$a_3 = 0.09678418$ $a_4 = -0.18628806$ $a_5 = 0.27886807$
$a_6 = -1.13520398$ $a_7 = 1.48851587$ $a_8 = -0.82215223$
$a_9 = 0.17087277$

Let $Y = \operatorname{erf}\left(X\right)$. and $Z = \operatorname{erfc}\left(X\right)$ .For each $i = 0,1,...,N$

$\begin{cases} Y_i^- = \operatorname{erf}\left(X_i^-\right) \\ Y_i^+ = \operatorname{erf}\left(X_i^+\right) \\ \delta Y_i^- = \delta X_i^- \tfrac{2}{\sqrt{\pi}} \exp\left(-X_i^-\right)^2 \\ \delta Y_i^+ = \delta X_i^+ \tfrac{2}{\sqrt{\pi}} \exp\left(-X_i^+\right)^2 \end{cases}$	$\begin{cases} Z_i^- = \operatorname{erfc}\left(X_i^+\right) \\ Z_i^+ = \operatorname{erfc}\left(X_i^-\right) \\ \delta Z_i^- = -\delta X_i^+ \tfrac{2}{\sqrt{\pi}} \exp\left(-X_i^+\right)^2 \\ \delta Z_i^+ = -\delta X_i^- \tfrac{2}{\sqrt{\pi}} \exp\left(-X_i^-\right)^2 \end{cases}$

The cumulative normal function $\Phi(x)=\frac{1}{\sqrt{2\pi}}\int\limits_{-\infty}^{x}\exp\left(-\frac{t^2}{2}\right)dt,\quad x\in\mathbb{R}$, can be calculated by

$$\Phi(x)=\begin{cases}\frac{1}{2}\left(1+\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right) & \text{if } x\geq 0\\ \\ \frac{1}{2}\left(1-\operatorname{erf}\left(-\frac{x}{\sqrt{2}}\right)\right) & \text{if } x<0\end{cases}$$

Let $Y=\Phi(X)$. For each $i=0,1,...,N$

$\begin{cases}Y_i^-=\Phi\left(X_i^-\right)\\ Y_i^+=\Phi\left(X_i^+\right)\end{cases}$	$\begin{cases}\delta Y_i^-=\delta X_i^-\frac{1}{\sqrt{2\pi}}\exp(\frac{-X_i^{-2}}{2})\\ \delta Y_i^+=\delta x_i^+\frac{1}{\sqrt{2\pi}}\exp(\frac{-X_i^{+2}}{2})\end{cases}$

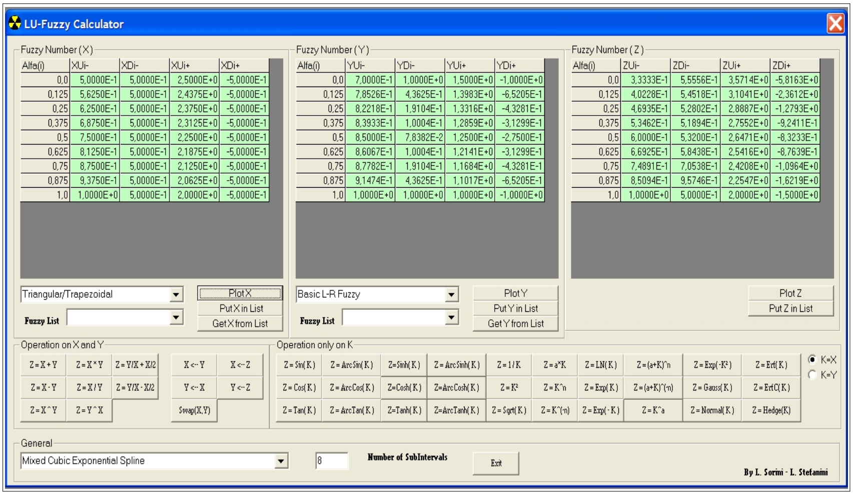
4 Implementation of the LU-Fuzzy Calculator

A hand-like fuzzy calculator has been implemented by a Windows-based frame.

It works by first defining input fuzzy numbers X and Y using the LU-fuzzy representation and produces Z as result of operations. Three boxes are designed to contain the LU-fuzzy representation (grid) of the fuzzy numbers X, Y and Z.

For each element $u\in\{X,Y,Z\}$, the grid box contains the LU-values α_i, u_i^-, δu_i^-, u_i^+ and δu_i^+ respectively.

To start the calculations, we have implemented a set of predefined types, including triangular, trapezoidal, general parametrized LU and LR fuzzy numbers.



LU-Fuzzy Calculator

Fuzzy Number (X)

Alfa(i)	XUi-	XDi-	XUi+	XDi+
0,0	5,0000E-1	5,0000E-1	2,5000E+0	-5,0000E-1
0,125	5,6250E-1	5,0000E-1	2,4375E+0	-5,0000E-1
0,25	6,2500E-1	5,0000E-1	2,3750E+0	-5,0000E-1
0,375	6,8750E-1	5,0000E-1	2,3125E+0	-5,0000E-1
0,5	7,5000E-1	5,0000E-1	2,2500E+0	-5,0000E-1
0,625	8,1250E-1	5,0000E-1	2,1875E+0	-5,0000E-1
0,75	8,7500E-1	5,0000E-1	2,1250E+0	-5,0000E-1
0,875	9,3750E-1	5,0000E-1	2,0625E+0	-5,0000E-1
1,0	1,0000E+0	5,0000E-1	2,0000E+0	-5,0000E-1

Triangular/Trapezoidal

Fuzzy List

Fuzzy Number (Y)

Alfa(i)	YUi-	YDi-	YUi+	YDi+
0,0	7,0000E-1	1,0000E+0	1,5000E+0	-1,0000E+0
0,125	7,8526E-1	4,3625E-1	1,3983E+0	-6,5205E-1
0,25	8,2218E-1	1,9104E-1	1,3316E+0	-4,3281E-1
0,375	8,3933E-1	1,0004E-1	1,2859E+0	-3,1299E-1
0,5	8,5000E-1	7,8382E-2	1,2500E+0	-2,7500E-1
0,625	8,6067E-1	1,0004E-1	1,2141E+0	-3,1299E-1
0,75	8,7782E-1	1,9104E-1	1,1684E+0	-4,3281E-1
0,875	9,1474E-1	4,3625E-1	1,1017E+0	-6,5205E-1
1,0	1,0000E+0	1,0000E+0	1,0000E+0	-1,0000E+0

Basic L-R Fuzzy

Fuzzy List

Fuzzy Number (Z)

Alfa(i)	ZUi-	ZDi-	ZUi+	ZDi+
0,0	3,3333E-1	5,5556E-1	3,5714E+0	-5,8163E+0
0,125	4,0228E-1	5,4518E-1	3,1041E+0	-2,3612E+0
0,25	4,6935E-1	5,2802E-1	2,8887E+0	-1,2793E+0
0,375	5,3462E-1	5,1894E-1	2,7552E+0	-9,2411E-1
0,5	6,0000E-1	5,3200E-1	2,6471E+0	-8,3233E-1
0,625	6,6925E-1	5,8438E-1	2,5416E+0	-8,7639E-1
0,75	7,4891E-1	7,0538E-1	2,4208E+0	-1,0964E+0
0,875	8,5094E-1	9,5746E-1	2,2547E+0	-1,6219E+0
1,0	1,0000E+0	5,0000E-1	2,0000E+0	-1,5000E+0

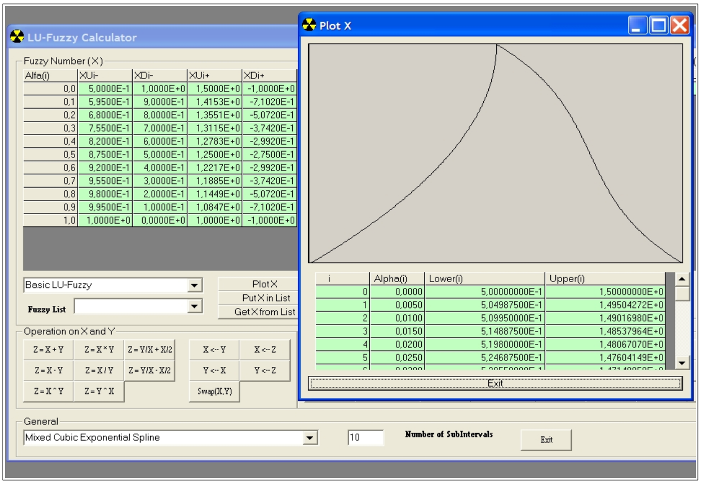
● K=X ○ K=Y

General

Mixed Cubic Exponential Spline

8 Number of SubIntervals

By L. Sorini - L. Stefanini



Fuzzy Number (X)

Alfa(i)	XUi-	XDi-	XUi+	XDi+
0,0	5,0000E-1	1,0000E+0	1,5000E+0	-1,0000E+0
0,1	5,9500E-1	9,0000E-1	1,4153E+0	-7,1020E-1
0,2	6,8000E-1	8,0000E-1	1,3551E+0	-5,0720E-1
0,3	7,5500E-1	7,0000E-1	1,3115E+0	-3,7420E-1
0,4	8,2000E-1	6,0000E-1	1,2783E+0	-2,9920E-1
0,5	8,7500E-1	5,0000E-1	1,2500E+0	-2,7500E-1
0,6	9,2000E-1	4,0000E-1	1,2217E+0	-2,9920E-1
0,7	9,5500E-1	3,0000E-1	1,1885E+0	-3,7420E-1
0,8	9,8000E-1	2,0000E-1	1,1449E+0	-5,0720E-1
0,9	9,9500E-1	1,0000E-1	1,0847E+0	-7,1020E-1
1,0	1,0000E+0	0,0000E+0	1,0000E+0	-1,0000E+0

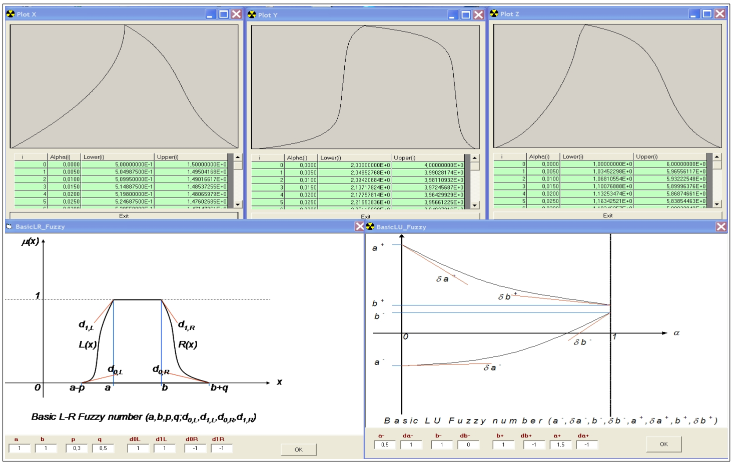
Basic LU-Fuzzy

Fuzzy List

i	Alpha(i)	Lower(i)	Upper(i)
0	0,0000	5,00000000E-1	1,50000000E+0
1	0,0050	5,04987500E-1	1,49504272E+0
2	0,0100	5,09950000E-1	1,49016980E+0
3	0,0150	5,14887500E-1	1,48537964E+0
4	0,0200	5,19800000E-1	1,48067070E+0
5	0,0250	5,24687500E-1	1,47604149E+0

Mixed Cubic Exponential Spline

10 Number of SubIntervals



Plot X

i	Alpha(i)	Lower(i)	Upper(i)
0	0,0000	5,00000000E-1	1,50000000E+0
1	0,0050	5,04987500E-1	1,49504168E+0
2	0,0100	5,09950000E-1	1,49016617E+0
3	0,0150	5,14887500E-1	1,48537255E+0
4	0,0200	5,19800000E-1	1,48065979E+0
5	0,0250	5,24687500E-1	1,47602685E+0

Plot Y

i	Alpha(i)	Lower(i)	Upper(i)
0	0,0000	2,00000000E+0	4,00000000E+0
1	0,0050	2,04852768E+0	3,99028174E+0
2	0,0100	2,09420684E+0	3,98110932E+0
3	0,0150	2,13717824E+0	3,97245687E+0
4	0,0200	2,17757814E+0	3,96429929E+0
5	0,0250	2,21553836E+0	3,95661225E+0

Plot Z

i	Alpha(i)	Lower(i)	Upper(i)
0	0,0000	1,00000000E+0	6,00000000E+0
1	0,0050	1,03452298E+0	5,96556117E+0
2	0,0100	1,06810554E+0	5,93222548E+0
3	0,0150	1,10076888E+0	5,89996376E+0
4	0,0200	1,13253474E+0	5,86874661E+0
5	0,0250	1,16342521E+0	5,83854463E+0

Basic L-R Fuzzy number $(a,b,p,q;d_{0,L},d_{1,L},d_{0,R},d_{1,R})$

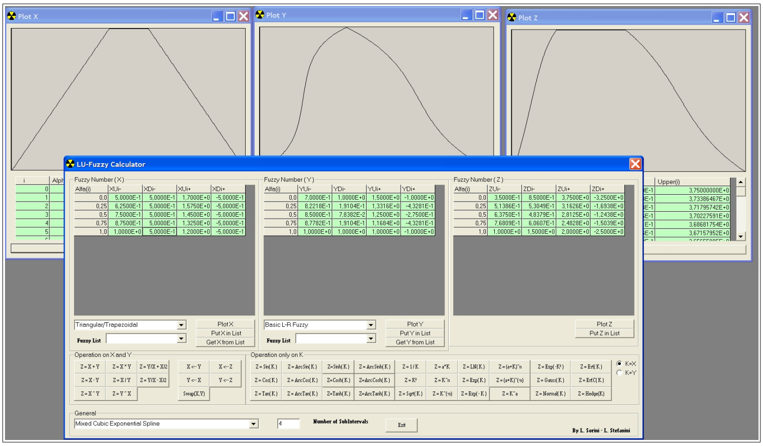
a	b	p	q	d0L	d1L	d0R	d1R
1	1	0,3	0,5	1	1	-1	-1

Basic LU Fuzzy number $(a^-,\delta a^-,b^-,\delta b^-,a^+,\delta a^+,b^+,\delta b^+)$

a-	da-	b-	db-	b+	db+	a+	da+
0,5	1	1	0	1	-1	1,5	-1

For any given type, it is possible to define the number N of subintervals (N+1 points) in the uniform $\alpha-$decomposition: all the calculations are performed exactly at the nodes of the decomposition and the monotonic splines are then used to interpolate at other values of $\alpha\in[0,1]$. It is possible to plot the membership functions of the inputs, the intermediate or final results. The Plot button opens a popup window with the graph of the membership function of the corresponding fuzzy number. To obtain the graphs or other representations, one of the models (rational or mixed monotonic splines) can be selected.

The standard arithmetic operations $Z=X+Y$, $Z=X-Y$, $Z=X*Y$, $Z=X/Y$ and the fuzzy extension of many elementary unidimensional functions are included. The



Fuzzy Number (X)

Alfa(i)	XUi-	XDi-	XUi+	XDi+
0,0	5,0000E-1	5,0000E-1	1,7000E+0	-5,0000E-1
0,25	6,2500E-1	5,0000E-1	1,5750E+0	-5,0000E-1
0,5	7,5000E-1	5,0000E-1	1,4500E+0	-5,0000E-1
0,75	8,7500E-1	5,0000E-1	1,3250E+0	-5,0000E-1
1,0	1,0000E+0	5,0000E-1	1,2000E+0	-5,0000E-1

Triangular/Trapezoidal

Fuzzy Number (Y)

Alfa(i)	YUi-	YDi-	YUi+	YDi+
0,0	7,0000E-1	1,0000E+0	1,5000E+0	-1,0000E+0
0,25	8,2218E-1	1,9104E-1	1,3316E+0	-4,3281E-1
0,5	8,5000E-1	7,8382E-2	1,2500E+0	-2,7500E-1
0,75	8,7782E-1	1,9104E-1	1,1684E+0	-4,3281E-1
1,0	1,0000E+0	1,0000E+0	1,0000E+0	-1,0000E+0

Basic L-R Fuzzy

Fuzzy Number (Z)

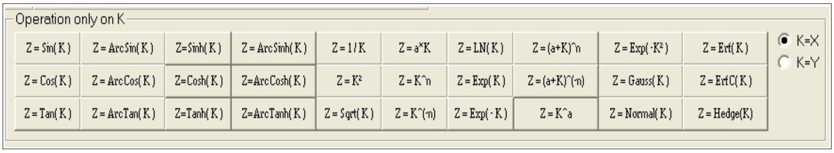
Alfa(i)	ZUi-	ZDi-	ZUi+	ZDi+
0,0	3,5000E-1	8,5000E-1	3,7500E+0	-3,2500E+0
0,25	5,1386E-1	5,3049E-1	3,1626E+0	-1,6938E+0
0,5	6,3750E-1	4,8379E-1	2,8125E+0	-1,2438E+0
0,75	7,6809E-1	6,0607E-1	2,4828E+0	-1,5039E+0
1,0	1,0000E+0	1,5000E+0	2,0000E+0	-2,5000E+0

Upper(i)
3,75000000E+0
3,73386467E+0
3,71795742E+0
3,70227591E+0
3,68681754E+0
3,67157952E+0

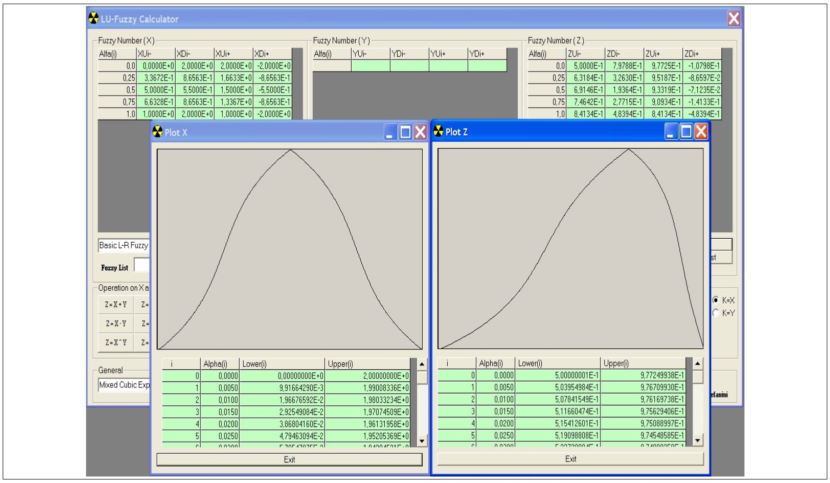
Mixed Cubic Exponential Spline

4 Number of SubIntervals

actual implemented functions are $Z = X^Y$, $Z = Y^X$ and, choosing $K = X$ or $K = Y$, $Z = \sin(K)$, $Z = \arcsin(K)$, $Z = \cos(K)$, $Z = \arccos(K)$, $Z = \tan(K)$, $Z = \arctan(K)$, $Z = \sinh(K)$, $Z = \sinh^{-1}(K)$, $Z = \cosh(K)$, $Z = \cosh^{-1}(K)$, $Z = \sqrt{K}$, $Z = 1/K$, $Z = \tanh(K)$, $Z = \tanh^{-1}(K)$, $Z = aK$ $(a \in \mathbb{R})$, $Z = K^2$, $Z = K^{\pm n}$ $(n \in \mathbb{N})$, $Z = \ln(K)$, $Z = \exp(K)$, $Z = \exp(-K)$, $Z = (a+K)^{\pm n}$, $Z = K^a$, $Z = \exp(-K^2)$, $Z = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}K^2)$, $Z = \operatorname{erf}(K)$, $Z = \operatorname{erfc}(K)$, $Z = Normal(K)$ $= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{K} \exp(-\frac{1}{2}t^2)dt$. Finally, some Hedge linguistic fuzzy operators are implemented (very, more or less, ...). The calculations are performed by clicking the button of the corresponding operation. The left group of buttons involves the binary operations. The second group of operators require the assigment of either X or Y to the temporary K and operate on K itself putting the result into Z.



It is possible to save a given (X, Y or Z) temporary result into a stored list (Put in List button), by assigning a name to it; a saved fuzzy number can be reloaded either in X or Y for further use (Get from List button). The data are saved into a formatted file having the same user-defined name. We illustrate an example $Z = Normal(X)$. First select a type of fuzzy number (trapezoidal, LU or LR) and set the number N of subintervals in the $\alpha-$decomposition (the higher N the higher the precision in the calculations); the maximal value of N is 100 and typical values are 2, 4, 8, 10. If the selection is loaded into the X-area, the corrisponding grid appears. To see the membership function of X, click the corresponding Plot button and a popup window appears. To apply the fuzzy extension to X, first select the assigment K=X nd then click the $Z = Normal(K)$ button.



Fuzzy Number (X)

Alfa(i)	XUi-	XDi-	XUi+	XDi+
0,0	0,0000E+0	2,0000E+0	2,0000E+0	-2,0000E+0
0,25	3,3672E-1	8,6563E-1	1,6633E+0	-8,6563E-1
0,5	5,0000E-1	5,5000E-1	1,5000E+0	-5,5000E-1
0,75	6,6328E-1	8,6563E-1	1,3367E+0	-8,6563E-1
1,0	1,0000E+0	2,0000E+0	1,0000E+0	-2,0000E+0

Fuzzy Number (Y)

Alfa(i)	YUi-	YDi-	YUi+	YDi+

Fuzzy Number (Z)

Alfa(i)	ZUi-	ZDi-	ZUi+	ZDi+
0,0	5,0000E-1	7,9788E-1	9,7725E-1	-1,0798E-1
0,25	6,3184E-1	3,2630E-1	9,5187E-1	-8,6597E-2
0,5	6,9146E-1	1,9364E-1	9,3319E-1	-7,1235E-2
0,75	7,4642E-1	2,7715E-1	9,0934E-1	-1,4133E-1
1,0	8,4134E-1	4,8394E-1	8,4134E-1	-4,8394E-1

Plot X

i	Alpha(i)	Lower(i)	Upper(i)
0	0,0000	0,00000000E+0	2,00000000E+0
1	0,0050	9,91664290E-3	1,99008336E+0
2	0,0100	1,96676592E-2	1,98033234E+0
3	0,0150	2,92549084E-2	1,97074509E+0
4	0,0200	3,86804160E-2	1,96131958E+0
5	0,0250	4,79463094E-2	1,95205369E+0

Plot Z

i	Alpha(i)	Lower(i)	Upper(i)
0	0,0000	5,00000001E-1	9,77249938E-1
1	0,0050	5,03954984E-1	9,76709930E-1
2	0,0100	5,07841549E-1	9,76169738E-1
3	0,0150	5,11660474E-1	9,75629406E-1
4	0,0200	5,15412601E-1	9,75088997E-1
5	0,0250	5,19098808E-1	9,74548585E-1

A detailed description of the calculator is in [5].

References

1. D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.
2. D. Dubois, H. Prade, *Possibility Theory. An approach to Computerized Processing of Uncertainty*, Plenum Press, New York, 1988.
3. D. Dubois, E. Kerre, R.Mesiar, H. Prade, Fuzzy Interval Analysis, in: *Fundamentals of Fuzzy Sets*, D. Dubois, H. Prade, Rds. Kluwer, Boston, The Handbooks of Fuzzy Sets Series, 2000, 483-581.
4. M.L. Guerra, L. Stefanini, Approximate Fuzzy Arithmetic Operations Using Monotonic Interpolations, *Fuzzy Sets and Systems*,150, 5-33, 2005.
5. L. Sorini, L. Stefanini, An LU-fuzzy calculator for the basic fuzzy calculus, Working Paper Series EMS, 101, University of Urbino, 2005 (extended version, 2006).
6. L. Stefanini, L.Sorini, M.L.Guerra, Parametric Representations of Fuzzy Numbers and Application to Fuzzy Calculus, *Fuzzy Sets and Systems*, 157, 2006, 2423-55.
7. L. Stefanini, L.Sorini, M.L.Guerra, Simulations of Fuzzy Dynamical Systems using the LU Representation of Fuzzy Numbers, *Chaos, Solitons and Fractals*, 29, 2006, 638-652.
8. L.A. Zadeh, Fuzzy Sets, *Information and Control,* 8, 1965, 338-353.

Fuzzy Parallel Processing of Hydro Power Plants – Why Not?

Dimitar Vasilev Lakov

Department of Intelligent Computer Technologies, Institute of Computer and Communication, Systems - Bulgarian Academy of Sciences, Acad. G. Bonchev street bl.2, 1113 Sofia, Bulgaria

Abstract. The paper deals with an improvement of implementation of Micro Hydro Power Plant. The main idea is to apply fuzzy control in such rather uncertain, ill defined and difficult problem as is in fact continuous power production in variable conditions: water supply, energy consumption, and different irregular consumers' loads. The aim is to demonstrate some advantages of fuzzy control able to resolve these problems borrowing human intuition, insight, and creative thinking inherent so far to human being only. Some results of the experiments are demonstrated on pilot MHPP, which now is in active stage of its exploitation. These experiments show a comparison between conventional microprocessor control and such with addition of fuzzy logic contours, which transform in fact the system in a hybrid control system.

1 Introduction

One of the most advanced tendencies of modern computing appears to be nowadays parallel processing that mimics human brain activity. A world problem upon decision is lack of parallel computing structures equal even in smallest extent to this extremely complex phenomenon – human brain. An attempt to overcome these difficulties is to apply fuzzy logic paradigm that virtually involves parallel principle of processing. This principle is imbedded taking into consideration at one and the same time multiple values of fuzzy representation instead of their single crisp value treatment.

Principally Micro Hydro Power Plant, which cover range between 500 and 1000KW power, work in two regimes:

1. Iceland, isolated power source, self controlling, and responsible for balance of the whole energy generation, consumption, and extra firing of excessive energy (as a rule they are synchronous type machines),
2. Main purpose power machines, which can act as energy generator for self consumption and outer of the system customers, as well as energy receiver from outer sources in recuperative regime. In the last case they act as energy accumulators. This strategy is more flexible and advanced. Therefore asynchronous type of machine is imperative.
3. Our consideration refers to the second smarter strategy, which has predominant role in nowadays private, private, and social distributed organization.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 495–503, 2007.
springerlink.com

4. Problems:
5. In case of connected power generation a necessity of very careful scheduling of power generation arises. This is very difficult due to uncertainty of power consumption, water source supply that define generation, and other unexpected outer perturbations. Therefore an intelligent scheme of control has to be provided, which take into consideration all uncertain factors. This scheme can be based on fuzzy control.
6. Parameters tuning of MHPP system also requires some intelligent actions since they rely in high extent on human intuition, experience, and proper action – another pros for fuzzy control application. (By the way this experiment is already described in the previous contribution [2]),
7. Program power generation, reflecting variable ambient conditions such as power generation, consumption, desired schedule of customer, etc. Another one pros for intelligent fuzzy control application.

All these problems naturally lead to just below accepted control strategy Apart from [2] some good examples of this sstrategy can be found in [3, 4, 5, and 6]. Authors in [7, 8, and 9] consider partial decisions of these problems.

2 Object Description

Figure 1 presents principal scheme of MHPP. It consists of two functionally different sub-systems: technological and informational. Technological streams are denoted by big bald arrows: One - for water stream from head water supply through water command effectors, Pelton turbine to tail water output; Other - for electrical power storm, which traces pressurized oil system, influencing on servos, from Pelton turbine through electrical generator and following grid connection with electricity as physical power output. Informational sub-system is formed by the following loops depicted by big hollow arrows and thin lines:

1. Information and emergency signals. The first are physical measurements from sensors to control ALU: temperature, pressure, level, nozzle position, etc.; other for signalization of correct generator functioning (U, f, cosφ). The second are signals and active responses by abnormal situations,
2. Control loop organization from/to control ALU, denoted as hollow arrows in vector form,
3. Auxiliary servo systems as amplifiers between controls unite and power effectors systems.

In Figure 2 the servo control system is shown. Emergency cut-off deflector is situated in the most front position of the water stream preventing the whole system from unwanted damages. All sub-systems are provided with two chamber reversible servo amplifiers and controlled coil devices.

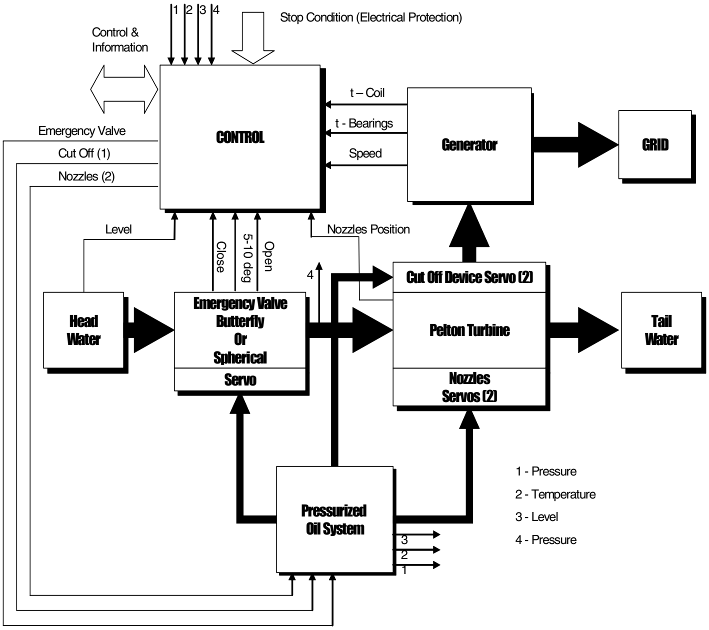


Fig. 1. Block scheme of MHPP

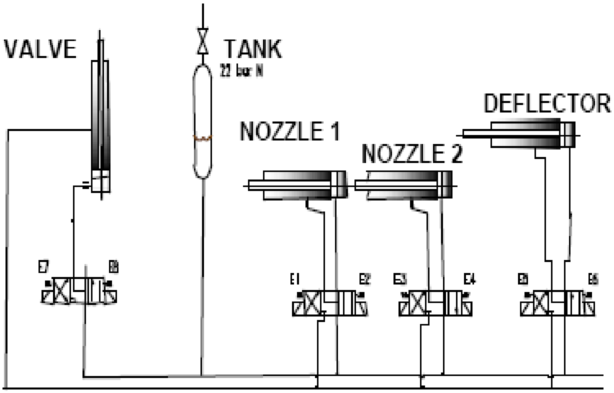


Fig. 2. Simplified local servo control scheme

3 Fuzzy Control Scheme

Optimal Power Generation (**OPG**) with two nozzles has feed forward organisation.

1. Variables:
2. Nozzle position opening **NOP** – antecedent 1 and 2 with five triangular terms: low **L**, sub-middle **S-M**, middle **M**, upper middle **U-M** and high **H**. Figure 3. shows granulation in normalized universe of discourse form.

3. Power generation **POW** - consequence of the fuzzy rule base with five triangular terms each: low **l**, sub-middle **s-m**, middle **m**, upper middle **u-m** and high **h**. **POW** has representation analogous to **NOP1/2**.
4. Fuzzy rule base representation

   They are presented in graphic as well table form in Figure 3 and Table 1.

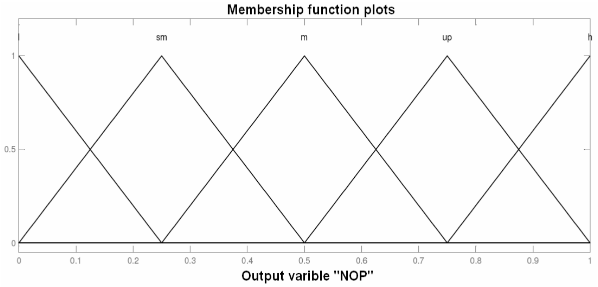


Fig. 3. NOP1/2 granulation

Table 1. Concise rule base representation for OPG

NOP1\NOP2	L	S-M	M	U-M	H
L	l	l	s-m	s-m	m
S-M	l	s-m	s-m	m	m
M	s-m	s-m	m	m	u-m
U-M	s-m	m	m	u-m	u-m
H	m	m	u-m	u-m	h

You can see symmetry in respect to upper left / lower right diagonal of Table 1., i.e. full mirror symmetry. So, we can take only low half plane and all diagonal consequences in consideration, excluding those above upper main diagonal, due to full symmetry response of both nozzles. By this way the number of rules drastically decreases from 25 to 15.
Rule base becomes:

If **NOP1**is **L** and **NOP2** is **L** then **POW** is **l**
If **NOP1**is **S-M** and **NOP2** is **L** then **POW** is **l**
If **NOP1**is **M** and **NOP2** is **L** then **POW** is **s-m**
If **NOP1**is **U-M** and **NOP2** is **L** then **POW** is **s-m**
If **NOP1**is **H** and **NOP2** is **L** then **POW** is **m**
If **NOP1**is **S-M** and **NOP2** is **S-M** then **POW** is **s-m**
If **NOP1**is **M** and **NOP2** is **S-M** then **POW** is **s-m**
If **NOP1**is **U-M** and **NOP2** is **S-M** then **POW** is **m**
If **NOP1**is **H** and **NOP2** is **S-M** then **POW** is **m**
If **NOP1**is **M** and **NOP2** is **M** then **POW** is **m**
If **NOP1**is **U-M** and **NOP2** is **M** then **POW** is **m**
If **NOP1**is **H** and **NOP2** is **M** then **POW** is **u-m**
If **NOP1**is **U-M** and **NOP2** is **U-M** then **POW** is **u-m**
If **NOP1**is **H** and **NOP2** is **U-M** then **POW** is **u-m**
If **NOP1**is **H** and **NOP2** is **H** then **POW** is **h**

i. Some additional consideration concerning control algorithms

Although there are many different variants of realisation, for example Mamdani, Takagi-Sugeno, Tzukamoto, etc. type of inference machines, here for simplicity, are accepted the following options: Mamdani type inference mechanism, center of gravity defuzzification method, and MAX/MIN fuzzy operators, representing OR/AND conjunctions.

An example of MATLAB representation is shown in Figure 4. The reason of this is a better understanding, simple algorithm visualisation, and graphical support of human operator in learning process of intelligent tuning. Figure 4 shows a typical decision of **POW** in MATLAB simulation. As you can see after definition, fuzzy inference, and defuzzification output is a crisp value, which is another proof for computer realisation for 'clever' artificial intelligence that unfortunately works up to now incomparable poorly in comparison with the genuine parallel processing of human brain. Despite of that, and as a necessary first step in this complex field, one can benefit of this strategy in respect to conventional one. In what follows in experimental part of the work, there are some comparisons in respect to this suggestion.

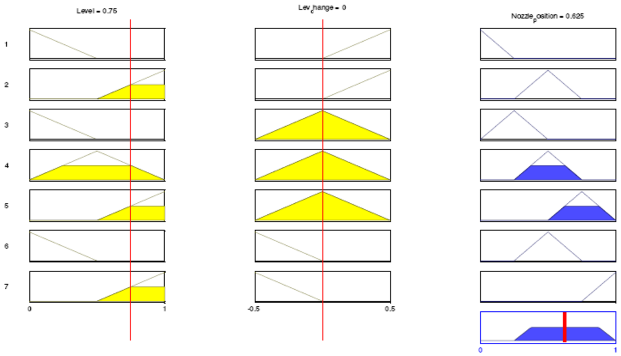


Fig. 4. POW in MATLAB simulation

Decision surface for this case is presented in Figure 5.

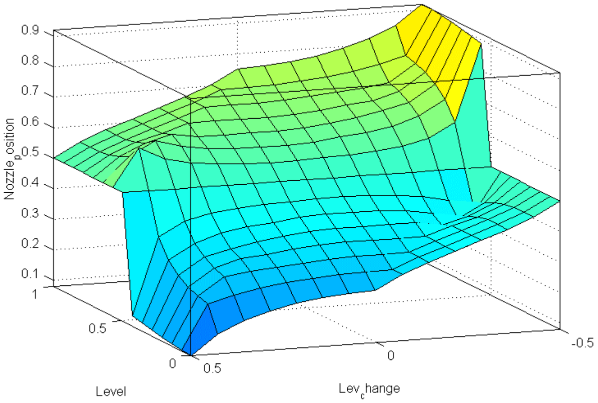


Fig. 5. MATLAB representation of ddecision surface for OPG

4 Experiments

This virtually parallel fuzzy based processing technique has been successfully experimented in control of real Micro HPP. As an example of intelligent **OPG** a pilot of ‘Vodemil’ MHPP is presented, an object that is situated on water bed in the area of ‘Vlahinska’ river near to ‘Cresna’ valley. It has 1000 KW full power at maximal loading and asynchronous type of active hydro power turbine, Pelton type. In accordance with accepted strategy the process of control and signaling comprises above described three stages: data acquisition, information processing, and control actions to servo system. The whole control system is incorporated in industrial computer Beckhoff CX1000 provided with appropriate peripherals: combined digital and analog input modules, digital output modules and serial interfaces.

Apart from data acquisition system, and inn addition to conventional of MHPP information sources, several signal and safety tools are considered:

1. Level sensor of input water stock,
2. Pressure sensor of input turbine water stock,
3. Incremental sensors of nozzle positions,
4. End cut-off actuators of valves and deflectors,
5. Specialized sensors, built into oil pressure device,
6. Temperature sensors of oil pressure and generator,
7. Universal measuring tool of generator and grid,
8. Auxiliary contacts of automatic fuse positions,
9. Control buttons and switches,
10. Sensor of local panel temperature.

All these facilities are serviced by means of specialised input modules sizing 48/4 digital/ analog inputs.

Information processing system consists of scaling, normalization, compression and other preliminary data processing, followed by performance of control algorithms and on-line visualization. These tasks are implemented by industrial computer supporting the following configuration:

1. Processor equivalent to MMX/266 Pentium,
2. 16MB flash memory, 32Mb RAM,
3. Serial interfaces: RS232, Ethernet LAN,
4. Standard Windows CE operation system.

Control actions comprise bbasic functions of control strategy, namely: performance of operator’s commands, automatic start/stop of generator, nozzle positioning control, optimal power generation, detecting and processing of emergency situations, and registering and recording of energy generation. Their specific characteristics are:

1. Two levels’ program realization: low for control program and high for visualization,
2. Compatible to IEC 61131-3 requirements of used program languages: IL, FBD, LD, SFC, ST, CFC,
3. VisualBasic Ver.6.0 applied as program realization for on-line archive information support,

4. Ethernet connection of industrial computer with standard operator's PC for timely reaction,
5. **OPG** algorithms are realized in MATLAB executive code, then after testing, are cross-complied on ST language and directly applied in control loops of Beckhoff CX1000 computer,
6. Highest priority of fuzzy control programs in operation system that assure steady state conditions for initial tuning and on-line operation.

Figure 6. shows a comparison of static characteristics of fuzzy and conventional control. Here full power **FP** (0 to 1MW) is a function of Nozzle Opening **NO** (0 to 100 per cent), respectively debit Q (0 to 6 m^3/min). *Series 1* presents power generation in emergency, cut-off deflector situation, *Series 2* is fuzzy, *Series 3* - conventional.

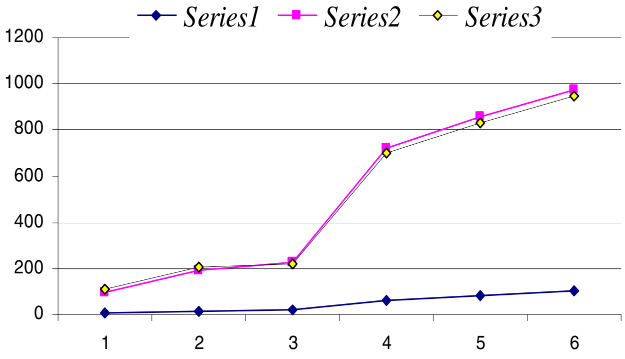


Fig. 6. Comparison of fuzzy and conventional control

Figure 7 presents daily schedule of power generation mapped in twelve hours for fuzzy and conventional control (*Series 1* – conventional, *Series 2* – fuzzy).

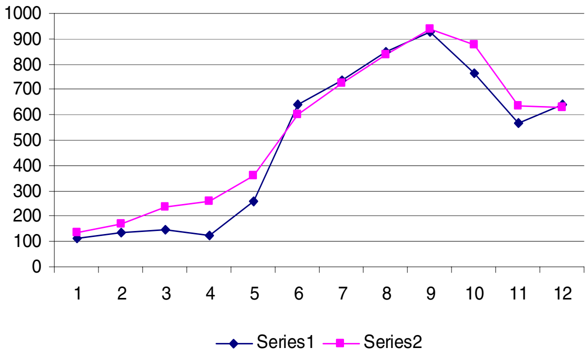


Fig. 7. Daily comparison of both controls

Figure 8 shows distribution of power generation within half a month for fuzzy and conventional control (*Series 1* – conventional, *Series 2* – fuzzy).

Although there are some differences of monthly schedule for both controls, principally, our intention was they to be equal. So, the differences could be explained as the different flexibilities of two principles of control.

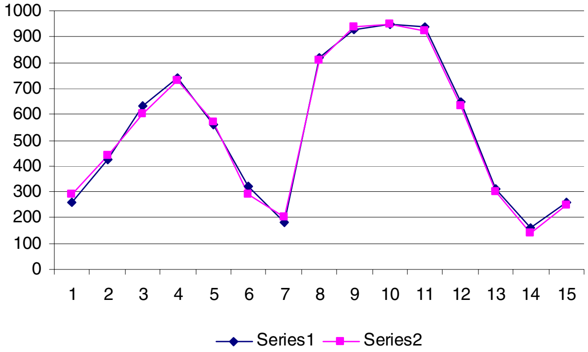


Fig. 8. Monthly comparison of both controls

5 Discussion

There are at least three themes to debate in this contribution:

1. How to estimate results of comparison between two approaches (fuzzy and conventional) in our case?
2. Obviously, static characteristics show slightly improvement in case of fuzzy control since it performs better for power generation above middle levels. It can be explained by smoother fuzzy control, which takes into consideration adjacent values with smaller membership degrees in parallel processing. Similar conclusion can be drawn in the time series analysis (daily and monthly curves of dynamic control). You can see slightly better performance of fuzzy control again in heavy loads. In lower loads opposite conclusion can be made.
3. What are the future perspectives of such approach?
4. Both hypotheses for better performance in some conditions have to be proved in more profound investigations, in different MHPPs as well as synchronous and asynchronous power generators. In any case such investigations will be very useful. Similar strategy can be applied if we involve into consideration other soft computing methods such as neural, genetic, etc. as well their combination. A cross comparison of such results hide promising perspectives for proving of soft computing approach in this field of investigation.
5. Is there any reason in development of this virtual parallelism into information processing?

Affirmatively, yes! The main supporting argument is what we can gain is simple: the simplicity of already proved soft computing methods, instead of other application of complex and rather sophisticated parallel computing structures. Apart from application in smart control of MHPP one can be thought about applications in distributed control and information systems. In such cases we can apply intelligent agent technology for realization of truly distributed structures instead of parallel processing computers. For example, it will be useful to apply new, fast developing paradigm, namely Soft Computing Agents [10].

References

[1] http://www.fuzzytech.com/.

[2] D.V.Lakov and S. Vasilev, Intelligent Tuning and Control of Micro Hydro Power Plant, Proc. of The 7th International Conference on Thechnical Informatics, 8-9 June, Timişoara, Romania, pp. 115-120.

[3] R. Babuška H.B. Verbruggen, An overview of fuzzy modelling for control, Proc. of the Third Conference of Energy Practice, 1996; 4: 1593-1606.

[4] IEEE Working Group. Hydraulic turbine and turbine control models for system dynamic studies. *IEEE Trans of Power Systeme Apparatus,* ; *7.* 1992, 167-179.

[5] S. Hagihara, H. Yokota, K. Gode, K. Isobe, Stability of a hydraulic turbine-generating unit controlled by PID governor, *IEEE Trans Power System Apparatus, 98,* 1979, 2294-2298.

[6] L. N. Hannett, B. Fardanesh, Field test to validate hydro turbine governor model structure and parameters, *IEEE Transactions Power System Apparatus, 9,* 1994, 1744-1751.

[7] L. N. Hannet, J. W Feltes, B. Fardanesh, W. Crean, Modeling and control tuning of a hydro station with units sharing a common penstock section, *IEEE Transactions Power Systems,* *14,* 1999; 1407-1414.

[8] J-SR. Jang J-SR, ANFIS: Adaptive-network based fuzzy inference system, *IEEE Transactions Systems, Man and Cybernetics, 23,* 1993; 665-685.

[9] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modelling and control, *IEEE Transactions Systems Man and Cybernetics, 15*, 1985,116-132.

[10] H. G. Berenji, The Unique Strength of Fuzzy Logic Control, *IEEE Expert,* 1994, 9 -14.

[11] C. C. Lee, Fuzzy Logic in Control Systems, *IEEE Trans on Systems Man and Cybernetics,* 20, 1990, 404 – 435.

[12] M. G. Cooper, Evolving a Rule-base Fuzzy Controller, *Simulation*, 1995, 65, 1.

[13] Robert A. Jacob, Michael I. Jordan, Steven J. Nolan, and Geoffrey E. Hinton, Adaptive Mixture of Local Experts, *Neural Computation*, 3, 1991, 39 – 87,

[14] S. Abe, M. S. Lang, A Classifier Using Fuzzy Rules Extracted Directly from Numerical Data, Proc. 2nd Int. Conf on Fuzzy Systems, March 1993, 1191 – 1198.

[15] G. Castellano, G. Attolico, E. Stella, and A. Distante, Automatic Generation of Rules for a Fuzzy Robot Controller, Proc. IEEE IROS'96, 1996, 1179 – 1186.

[16] R. R. Yager, On the Hierarchical Structure for Fuzzy Modelling and Control, *IEEE Trans on Systems Man and Cybernetics,* 22, 4, July/August 1993, 1189 – 1197.

[17] F. Michaud, G. Lachiver, and C. T. Le Dinh, A New Control Architecture Combining Reactivity, Combination and Motivation for Situated Autonomous Agents, in Proc. Conf. Simulation of Adaptive Behaviour, September 1996.

[18] D. Lakov, Routing via Dedicated Intelligent Agents, ESIT2000 – the European Symposium on Intelligent Techniques, 2000, September 14-15, Aachen, Germany, 270-275.

A Dynamic Method of Experiment Design of Computer Aided Sensory Evaluation

Bin Luo

College of Chemistry & Environment Protection Engineering, Southwest University for Nationalities 610041 Chengdu, P.R. China
luomaster@126.com

Abstract. It plays an important role in sensory evaluation to optimize experiment design, i.e. using less number of tests to obtain available data as possible by intelligent technologies. This paper presents one method for optimizing experiment design based on learning automaton, and this method is applied in computer aided sensory evaluation (abbr. CASE). The validity of this method is showed by the result of experiment from CASE.

1 Introduction

In sensory and consumer research, a panel of assessors is often used to study properties of certain products [1]. There are a lot of methods of how to obtain the data from panel [2], such as GPA [3], GCA [4][5], Test of pairs and balanced-block [6] are basic methods to obtain the data from panel in sensory evaluation. The method of test of pairs can distinguish correspondingly the difference between two samples, however, this method is faced with two defaults: firstly, if the number of samples is large, the number of test is very large also, for n samples, the number of test is $N = C_n^2$, for example, n=35, N=595; secondly, the quality of evaluation is influenced by the number of samples, along with increasing sample's quantity, the quality of evaluation drops. Herbert Stone [6] described impact of contrast and convergence on evaluation of foods. The contrast error is characterized by two products scored as being very different from each other and the magnitude of the difference being much greater than expected. Convergence is the opposite effect, usually brought about by contrast between two (or more) products masking or overshadowing smaller differences between one of these and other products in the test. The balanced-block method is faced with the 2nd problem that exists in the method of test of pairs. We should pay attention to the fact of different of memory and learning among evaluators. In researches of psychology [7], memory curve and learning curve express the degree of recollection of grasp knowledge and acquirement of new knowledge using experience respectively. Learning automaton (LA)[8][9] has been widely used in the field of control and researchers have established optimization for no-linear systems, and many models of LA have been built.

In this paper, we presents a dynamic method for designing experiment in sensory evaluation; this method based on the LA and aims at optimizing experiment design, i.e. reducing the number of tests. In sensory evaluation, firstly, we use

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 504–510, 2007.
springerlink.com

dichotomy to judge the direction of a new sample, i.e. the new sample belongs to left set or right set; secondly, we use the intelligent technologies include fuzzy sets and LA to optimize the experiment designs, i.e. using the less number of test to get the result for evaluating the new sample as possible. In fact, we can use dichotomy to fulfill a taxis for all samples in less number of test than that of the traditional experiment methods, and apply the method of LA in the process of test optimizing the method of dichotomy, therefore the new number of test will be less than that of dichotomy. We design a special archetypal program to aid human being doing test in sensory evaluation, and the result of between CASE and expert test is high relative.

2 Applying Learning Automaton in Experiment Design

2.1 Learning Automaton

A learning automaton (LA) is a stochastic automaton in feedback connection with a random environment [8] (See Fig.1). Its output (actions) and inputs are the input to the environment and the output of the environment (responses) respectively.

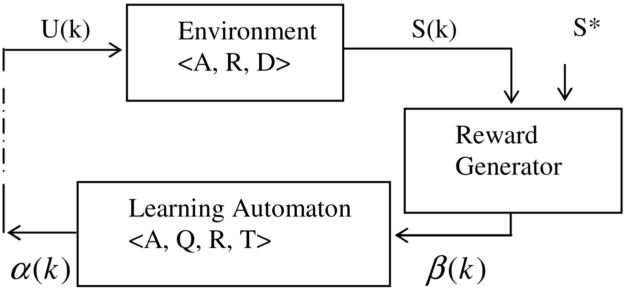


Fig. 1. General scheme of learning automaton

In general, a learning automaton is defined by <*A, Q, R, T*> and the environment by <*A, R, D*>, where

A and $\alpha(k)$ are the set of all actions of the automaton and the action of the automaton at instant *k*, and $\alpha(k) \in A$ for *k=0, 1, 2, ...* *A* is the set of outputs of the automaton and it is also the set of inputs to the environment.

R and $\beta(k)$ are the domain of responses from the environment and the response received by the automaton at instant *k* , and $\beta(k) \in R$, $\forall k$. $\beta(k)$ is the output of the environment at instant *k* and it is also the input to the automaton.

D, *Q* and *T* are the set of reward probabilities, the state of the automaton and the learning algorithm or the reinforcement scheme

S* express the character of memory and learning for evaluators, shorted form a set of result of evaluated memorized values.

2.2 Applying Dichotomy in Sensory Evaluation

The basic principle of the method of dichotomy is as following. Under given condition, it is prior to evaluate the new sample using the intermediate result of having evaluated samples. This method can cancel the repeat experiments that are no useful in distinguishing the new sample, and fulfill taxis for evaluating samples in lesser experiment times than that of traditional method in sensory evaluation.

The biggest number of tests using dichotomy is calculated by formula(1) [10]:

$$N_d = \sum_{i=1}^{n} p_i = 1 + 2 + \sum_{i=2}^{r} (i+1) \times 2^i - (2^{r+1} - n - 1) \times (r+1) \tag{1}$$

For example, if $n = 35$, the number of test using test of pairs is $N = 595$, and the result of the biggest number of test using dichotomy is $N_d = 151$, and $N_d < n \log_2 n = 35 \log_2 35 = 180$.

2.3 Strategies for Strengthening the Newest Test Using the Method of LA

There are three steps of strategies for strengthening the newest test using the method of LA.

Step 1, $k = 0$, it denotes starting to estimate the new sample. According to the relation of $\succ$ to make a taxis for r evaluated samples, and $\succ$ denotes the result of evaluation being from big to small, we suppose the equal similarity to all evaluated samples. Set $[0,1]$ to denote the result of similarity among samples, $u_r(k)$ to denote the reward probability of the k$^{\text{th}}$ with the r$^{\text{th}}$ action of test, and set

$$u_1(0) = u_2(0) = \dots = u_r(0) = 0.5 \tag{2}$$

$$a_1(0) = a_2(0) = \dots = a_r(0) = \frac{1}{r} \tag{3}$$

(2) and (3) denote the new sample is the same similar with all evaluated samples. The parameters of the model of LA are built as followings. Set the action sets, action probability, reward probability set and learning algorithm or the reinforcement scheme are A, $p_i(0)$, D and T respectively.

$$A = \{a_1, a_2, \dots, a_r\}, r \geq 3 \tag{4}$$

$$p_i(0) = a_i(0) = \frac{1}{r} \tag{5}$$

$$D = \{u_1(k), u_2(k), \dots, u_r(k)\} \tag{6}$$

$$T\text{: } p(k+1) = (1-\lambda \times d_m(k))p(n) + \lambda \times d_m(k) \times e_m \quad (7)$$

Step 2, $k \geq 1$, we apply the dichotomy in doing tests, i.e. the middle evaluated sample is prior chosen to do test, and $a_m(1) = 0.5 > \frac{1}{r}$, $p(1) = a_m(1) = 0.5$, and then calculate the result of the set $\{u_1(1), u_2(1), \ldots u_r(1)\}$. $p(k)$ is determined by the result of strength operator (*T*) , and do the next test with the evaluated sample according to the maximum of $p_i(k)$, $i = 1,2,\ldots,r$.

Step 3, the condition of stop testing is as followings. For the left set, if $d_{mj}(k+1) = j-1$, j is the evaluated sample that is chosen to do testing according to the action probability, then stop doing test. For right set, if $d_{mj}(k+1) = j+1$, then stop doing test. Otherwise to repeat the step 2.

3 Design the Parameters of Memory and Learning for Evaluators in Computer Aided Sensory Evaluation

In the process of designing computer aided sensory evaluation, we should pay attention to evaluators' ability of memory and learning for samples. We use error coefficient of contrast and convergence to denote the parameters of memory and learning for evaluators.

Set $R(1)$ and $R(2)$ denote the first two tests, and $R(1) = R(1/0)$, $R(2) = R(2/1)$. There exit four cases for the first two tests, i.e. $R(1) < 0.5$ and $R(2) < 0.5$, $R(1) > 0.5$ and $R(2) > 0.5$, $R(1) > 0.5$ and $R(2) < 0.5$, and $R(1) < 0.5$ and $R(2) > 0.5$.

Case 1, $R(1) < 0.5$ and $R(2) < 0.5$, we can't calculate the error coefficient of contrast or convergence, and the more tests are needed for case 1.

Case 2, $R(1) > 0.5$ and $R(2) > 0.5$, we can't calculate the error coefficient of contrast or convergence yet, in the same way; the more tests are needed for case 2 also.

Case3, $R(1) > 0.5$, $R(2) < 0.5$, and we set $R(1) - 0.5 = a_1$, $0.5 - R(2) = a_2$, $R(3) = R(2/0)$. There include two cases for case 3, i.e. $a_1 \geq a_2$ and $a_1 < a_2$.

① $a_1 \geq a_2$, $R(3) > 0.5$, and we set $R(3) - 0.5 = a_3$. This case can be described as the following model.

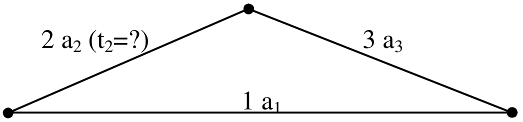


Fig. 2. A model for calculating the error coefficient of contrast or convergence

In Fig.2. $a_1 \geq a_2$, $a_1 \geq a_3$, and $a_1 - a_2 \leq a_3$, set t_2 denote the error coefficient of contrast for the 2nd sample, and it's calculated by the following formula.

$$t_2 = 1 - \frac{a_1 - a_3}{a_2} \leq 1 \tag{8}$$

② $a_1 < a_2$, $R(3) < 0.5$, and we set $0.5 - R(3) = a_3$,in the same way, we can calculate the error coefficient of convergence for the 1st sample, and denoted as t_1 that can be determined by the formula (9)

$$t_1 = 1 - \frac{a_2 - a_3}{a_1} \leq 1 \tag{9}$$

Case 4, $R(1) < 0.5$, $R(2) > 0.5$, and we set $0.5 - R(1) = a_1$, $R(2) - 0.5 = a_2$, $R(3) = R(2/0)$. There include two cases also, i.e. $a_1 \geq a_2$ and $a_1 < a_2$.

① $a_1 \geq a_2$, $R(3) < 0.5$, and we set $0.5 - R(3) = a_3$, we can get the error coefficient of contrast for the 2nd sample using the formula (8).

② $a_1 < a_2$, $R(3) > 0.5$,and set $R(3) - 0.5 = a_3$, we can get the error coefficient of convergence for the 1st sample using the formula (9).

As similar in the first two tests, we can get the error coefficient of contrast and convergence about more other samples.

4 Result of Test Using CASE

We invited one expert to do test using CASE, and she took less than 16 minutes for evaluating nine samples (the number of sample is from zero to eight) for *Roughness* using CASE on 19, October, 2005.

The process of evaluation and the variation of memory and learning of this expert are shown as Table 1 and Fig. 3 respectively.

In Table1, the result of test is given by the expert according the similarity among samples using one number belong to [0,1]. The total number of tests using our method is 17. The total number of tests using test of pairs is 36, and its will take about 33 minutes, estimated from the averaged speed of evaluation.

Table 1. Process of evaluation

Number of tests	New sample	Evaluated sample	Result of test	Times of test	Record of time
1	1	0	0.42	1	19:19:51
2	2	0	0.91	1	19:20:52
3	3	0	0.58	1	19:21:50
4	3	2	0.17	2	19:23:14
5	4	3	0.58	1	19:24:08
6	4	2	0.2	2	19:25:17
7	5	3	0.67	1	19:26:11
8	5	2	0.58	2	19:27:15
9	6	4	0.42	1	19:28:22
10	6	0	0.58	2	19:29:17
11	6	3	0.58	3	19:30:20
12	7	6	0.33	1	19:31:22
13	7	1	0.25	2	19:31:52
14	8	6	0.58	1	19:33:03
15	8	5	0.33	2	19:33:47
16	8	2	0.09	3	19:34:26
17	8	4	0.42	4	19:35:02

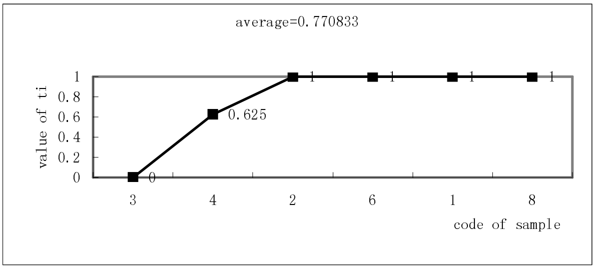


Fig. 3. Variation of memory and learning

According to the result of Fig.3, the memory and learning of this expert in the process of evaluation is different for samples.

5 Conclusions

This paper putted forward a dynamic method of experiment design in sensory evaluation. This method has two aspects role: one is to decrease the total number of tests, and it can reach $N_{LA} \le N_d \le n\log_2 n$. Secondly, due to considering the evaluator's intelligent (include memory ability and learning ability, they are denoted as an initial of t_0 and using intelligent technologies (fuzzy sets, LA), this method can improve the relative quality in sensory evaluation. However, the factors that affect experiment design in sensory evaluation are various, including product, evaluation goal, environment, evaluation cost, testing method and so on, therefore it's necessary for further studying evaluation experiment design in many ways.

Acknowledgments

We gratefully acknowledge the support of National Natural Science Foundation of China (Grant No. 60474022).

References

1. Dijksterhuis,G.B., Multivariate data analysis in sensory and consumer science, Food & nutrition press, Inc. Trumbull, Connecticut 06611 USA（1997）.
2. Zeng, X.Y., Ding,Y.S., An introduction to intelligent evaluation, Journal of Donghua university no.3(2004):1-4
3. Goer.J.C., Generalized Procrustes Analysis, Psychometrika, (1975) 40,33-51
4. Van der Burg, E. Nonlinear canonical correlation and some related techniques. Leiden: DSWO press (1988)
5. Van der Burg, E., Dijksterhuis, D.B., Nonlinear canonical Correlation Analysis of Multiway Data, In: Coppi,R., Bolasco, S. (Eds.) Multiway Data Analysis, 245-255, North-Holland, Elsevier Science Publisher B.V. (1989)
6. Stone H, Sidel J.L, Sensory Evaluation Practices (Third Edition). Academic Press, Inc., San Diego, CA (2004)
7. Lewandowsky, S., and Murdock, B.B., Memory for serial order. Psychological Review, (1989) 96,25-57
8. Lakshmivarahan S., Learning algorithms theory and applications, Springer, New York (1981)
9. Zeng，X.Y., Liu, Z.Y., A learning automata based algorithm for optimization of Continuous Complex Functions，information science, Vol.174, Issue3-4 (2005): 165-175
10. Liu, X.H., Zeng,X.Y, Xu,Y Ludovic K. A method for optimizing dichotomy in sensory evaluation, 2006 International Conference on Intelligent Systems & Knowledge Engineering (ISKE2006), Shanghai, China, April 6-7, (2006)

Measure of Uncertainty in Regional Grade Variability

Bulent Tutmez[1] and Uzay Kaymak[2]

[1] Inonu University, School of Engineering, 44280 Malatya, Turkey,
Tel: +90.422.3410010, Fax: +90.422.3410046
btutmez@inonu.edu.tr
[2] Erasmus University Rotterdam, Econometric Institute, P.O. Box 1738,
3000DR, Rotterdam, The Netherlands

Abstract. Because the geological events are neither homogeneous nor isotropic, the geological investigations are characterized by particularly high uncertainties. This paper presents a hybrid methodology for measuring of uncertainty in regional grade variability. In order to evaluate the fuzziness in grade values at ore deposit, point cumulative semimadogram (PCSM) measure and a metric distance have been employed. By using the experimental PCSMs and their linear models, measures of fuzziness have been carried out for each location. Finally, an uncertainty map, which defines the regional variation of the uncertainty in different categories, has been composed.

Keywords: Uncertainty, fuzziness, grade, regional variability, semimadogram.

1 Introduction

Ore grade is one of the most important characteristics for evaluating a mineral deposit. The grades are used in all phases of a mining project, including feasibility, mine planning and production scheduling [1]. The degree of grade variability (heterogeneity) may be quite varied depending on the related geological processes and environment. In general, the higher variability of the grade, the larger will be the uncertainty of its evaluation. The assessment of uncertainty in grade variability has paramount importance for mining investments, their financial investors and/or shareholders.

Grade variability deal with random (stochastic) event and this is the reason why the probability theory is at present the basic tool to handle uncertainties. In recent years, a number of studies have been carried out based on probabilistic uncertainty method such as geostatistical simulation for handling the uncertainty of ore grades [2-3]. However, no single uncertainty theory can claim to be able to model all types of uncertainty [4]. In addition, a measure of uncertainty can also be used for measuring two types of natural categories: 'vagueness' and 'ambiguity', respectively.

Regional variability is a qualitative characteristic denoting that the grades observed at different locations do not have the same value. Therefore, both uncertainties due to natural variability and qualitative property of the regional structure enable us to use fuzzy set theory for measures of the uncertainty in regional grade variability. This study gives a hybrid methodology which uses the probabilistic regional dependence measure based on point cumulative semimadogram (PCSM) and then measures of

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 511–518, 2007.
springerlink.com

fuzziness in linear madogram models via metric distance function. Use of fuzzy approach for identification of the uncertainty in regional grade variability is a novel study. This methodology provides useful information for spatial uncertainty assessments in a sample ore deposit.

This paper is organized as follows. Section 2 presents a description for measures of regional variability by PCSM. Section 3 deals with measures of fuzziness by distance metric. Section 4 gives an application of the methodology. Section 5 concludes the paper.

2 Regional Variability

Quantifying the regional grade variability within an ore site is carried out using the grade measurements at distinct locations. According to the theory of regionalized variables [5], two grade values *g(x)* and *g(x+h)* at two points *x* and *x+h* separated by the vector *h* are spatially correlated. As the distance between these grades increases, one would expect that the correlation decreases and vice versa. The spatial correlation is expressed the variance of the differences between two attribute grade values

$$2\gamma(h) = Var\left[g(x) - g(x+h)\right] \tag{1}$$

where *g(x)* and *g(x+h)* are random grades defined at locations *x* and *x+h, Var* is the variance operator and *γ(h)* is the semivariogram at distance *h*. Large variability denotes that the degree of dependence among grades might be rather small even for locations close to each other. Although different variance and correlation techniques in the literature used to quantify the degree of regional variability, these techniques cannot account correctly for regional dependence caused by either irregularity or nonnormal distribution functions of sampling designs. Because of this, some new measure approaches such as cumulative semivariogram [6] and cumulative semimadogram [7] have been proposed.

The PCSM function [8] is employed in this study for identifying the regional behaviours of grades around a location concerned. By this function, the regional effect of all of the other locations within the ore deposit on the location concerned is quantified. This function uses the absolute difference between *g(x)* and *g(x+h)*. The PCSM is more convenient, if the deposit has outlier values due to the absolute difference measure [8],

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} \left| g(x) - g(x+h_i) \right| . \tag{2}$$

The experimental PCSM graph can be obtained by plotting the distances versus corresponding successive cumulative sums of half-absolute differences. This procedure gives a nondecreasing function which is the sample PCSM at the concerned location.

3 Measures of Uncertainty

Although a PCSM function provides useful information for a location concerned, the model of the function includes uncertain-fuzzy characteristics, that is, the madogram can not be modeled exactly and which depends on the spatial grade variability. Therefore, measure of fuzziness in PCSM models can be treated by fuzzy set theory. In general, a measure of fuzziness is a function

$$f : P(X) \to \Re^{+},$$

where $P(X)$ denotes the set of all fuzzy subsets of *X*. That is, the function f assigns a value of *f(A)* to each fuzzy subset *A* of *X* and characterizes the degree of fuzziness of *A* [9].

Measure of fuzziness is defined in terms of a metric distance (*e.g*. Hamming or Euclidean) of *A* from any of the nearest crisp sets, say crisp set *C*, for which

$$\mu_C(x) = 0 \qquad if\ \mu_A(x) \le \frac{1}{2}$$

and

$$\mu_C(x) = 1 \qquad if\ \mu_A(x) > \frac{1}{2}.$$

If the Hamming and Euclidean distances are used, the measures of fuzziness are expressed by the following functions, respectively [10]:

$$f(A) = \sum_{x \in X} \left| \mu_A(x) - \mu_C(x) \right|, \tag{3}$$

$$f(A) = \Big(\sum_{x \in X} \left| \mu_A(x) - \mu_C(x) \right|^2 \Big)^{1/2} \tag{4}$$

In addition to metric distances mentioned above, other metric distances may be used as well. For example, the Minkowski class of distances yields a class of fuzzy measures

$$f_w(A) = \Big(\sum_{x \in X} \left| \mu_A(x) - \mu_C(x) \right|^w \Big)^{1/w} \tag{5}$$

where $w \in [1, \infty]$. Obviously, (2) and (3) are special cases of (5) for *w=1* and *w=2*, respectively. A related measure of fuzziness is obtained in a similar way in terms of the distance between the fuzzy set *A* and its complement A^c. In general, the less *A* differs from A^c, the fuzzier it is.

In our methodology, first the PCSM values are standardized between 0 and 1 and then the PCSM is evaluated like a membership function. Therefore, the degree of fuzziness of a PCSM function can be expressed in terms of the distinction between the PCSM line (fuzzy set) and its complement (crisp set). The less a PCSM differs from its complement, the fuzzier it is. If a PCSM *A* is less fuzzy than PCSM *B,* then [11]

$$A < B \text{ if and only if } |\mu_A(x) - c(\mu_A(x))| \geq |\mu_B(x) - c(\mu_B(x)| \, . \tag{6}$$

The measure of fuzziness for a distance function $D_{c,w}$ can also be expressed for infinite, but bounded PCSM functions. For a distance interval $X = [a,b]$, $D_{c,w}$ then becomes

$$D_{c,w}(A, A^c) = \left(\int_a^b \delta_{c,A}^w(x)\, dx \right)^{1/w} \tag{7}$$

where δ denotes the metric distance and A^c is complement of PCSM function. For a general metric distance *D,* an arbitrary crisp subset *Z* of *X* can be defined. By using this subset, the largest possible distance in *P(X)* can be given for a given *c*,

$$D_{c,w}(Z, Z^c) = \left(\int_a^b dx \right)^{1/w} = (b-a)^{1/w} \, . \tag{8}$$

Thus, the measure of fuzziness $f_{c,w}$ and its normalized version for the distance $D_{c,w}$ can be expressed as follows

$$f_{c,w}(A) = (b-a)^{1/w} - \left(\int_a^b \delta_{c,A}^w(x)\, dx \right)^{1/w} \, , \tag{9}$$

$$\hat{f}_{c,w}(A) = \frac{f_{c,w}(A)}{(b-a)^{1/w}} \, , \tag{10}$$

where $\hat{f}_{c,w}(A)$ is the normalized measure of fuzziness.

4 Implementation

In order to assess the regional uncertainty, the data set taken from chromium deposit has been employed. Adana-Karsantı Chromium deposit considered in this case study is one of the most productive areas in Anatolia, Turkey. The locations of the 25 records are shown in Fig.1.

The chromium data have been scaled by using a linear transformation between 1.8 and 6.8. The semimadogram analysis is used for defining the regional dependence around each reference (pivot) location concerned. An established PCSM plot at one of the locations, the experimental PCSM of location no 19 is shown in Fig. 2.

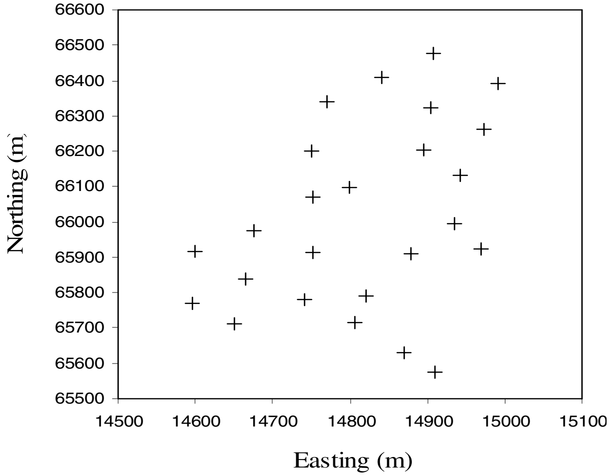


Fig. 1. The data locations employed in the case study

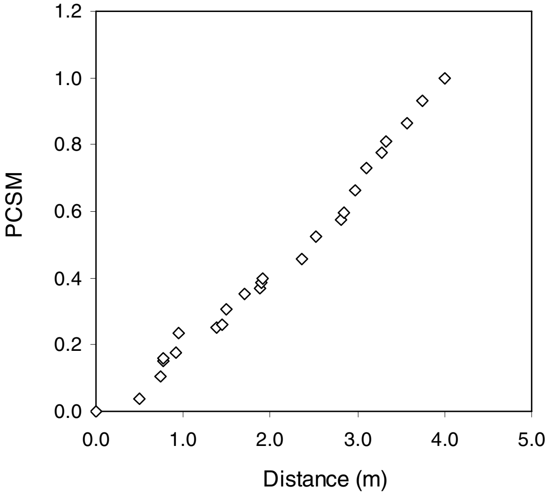


Fig. 2. Sample experimental PCSM plot for location no 19

In order to measure the fuzziness of the sample PCSMs, first the PCSM functions have been modeled and then the complements of the model equations have been determined. Although there are typical models used in spatial modelling, the linear modeling may be more suitable due to the cumulative trends of the PCSM functions. Figs. 3-4 indicate two sample trend models and their complements. In Figs. 3-4, the models A_{19} and A_{20} have been stated on the set of standardized distances $H = [0,4]$ by the linear memberships. To calculate the actual degrees of fuzziness of the linear models considered, we use the Hamming distance $w = 1$. As can be seen in the Figures 3 and 4, the models and their complements cross at a critical distance where the membership equal to 0.5. The critical distance of model A_{19} can be defined as a following calculation

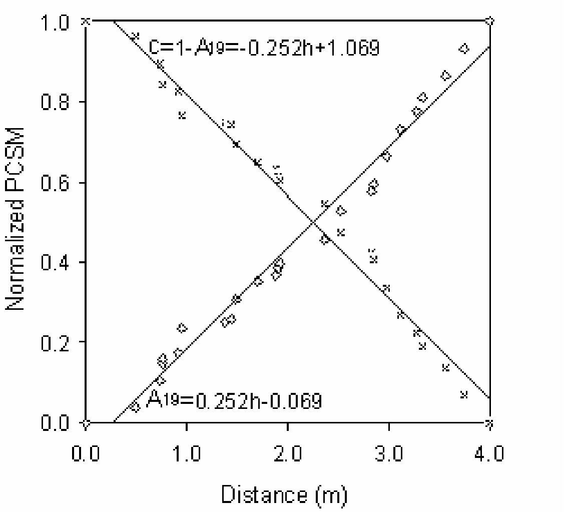


Fig. 3. PCSM model and complement for location no 19

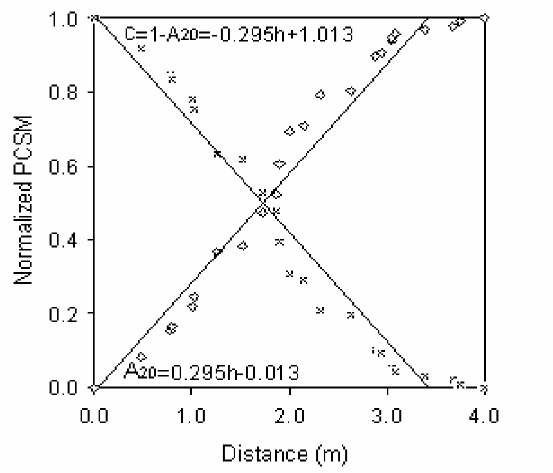


Fig. 4. PCSM model and complement for location no 20

$$0.5 = 1.069 - 0.252\,h \Rightarrow h = 2.26\ .$$

The metric distance as a function of h is given by

$$\delta_{c,19}(h) = |\mu_c(h) - (1 - \mu_c(h)| = |2\mu_c(h) - 1| = |-0.504h + 1.138|$$

and

$$D_{c,1}(A_{19}^c, A_{19}) = \int_0^{2.26} [(-0.504h + 1.138) - 1]\, dh + \int_{2.26}^{4} [1 - (-0.504h + 1.138)]\, dh = 1.529.$$

Now applying Eq. (9),

$$f_{c,1}(A_{19}) = 4 - 1.529 = 2.471\,;$$

after normalization, presented by Eq. (10), the degree of fuzziness can be obtained

$$\hat{f}_{c,1}(A_{19}) = \frac{2.471}{4} = 0.618.$$

The same calculation procedure is used for A_{20}:

$$0.5 = 1.013 - 0.295h \Rightarrow h = 1.71.$$

$$\delta_{c,20}(h) = |\mu_c(h) - (1 - \mu_c(h)| = |2\mu_c(h) - 1| = |-0.59h + 1.026|$$

$$D_{c,1}(A_{20}^c, A_{20}) = \int_0^{1.71} [(-0.59h + 1.026) - 1]\, dh + \int_{1.71}^{4} [1 - (-0.59h + 1.026)]\, dh = 3.030\,,$$

$$f_{c,1}(A_{20}) = 4 - 3.030 = 0.970\,,$$

$$\hat{f}_{c,1}(A_{20}) = \frac{0.970}{4} = 0.243.$$

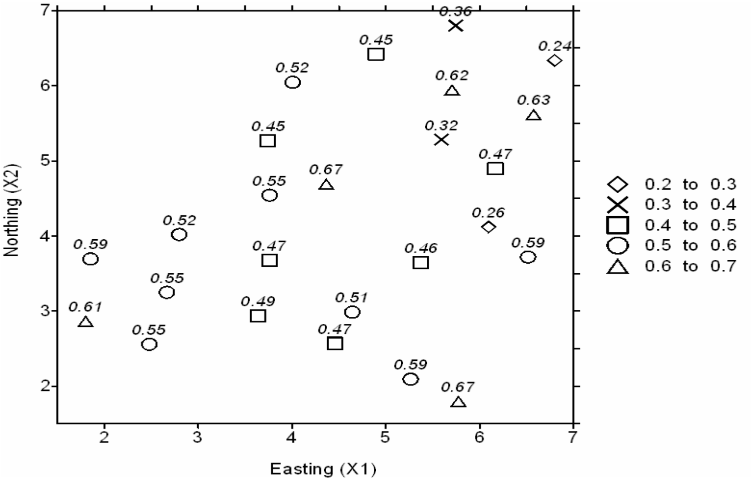


Fig. 5. Uncertainty map of the deposit

The results show that, the degree of fuzziness of model A_{20} is smaller than that of model A_{19}. The calculation process outlined above has been applied to the entire deposit and the actual degrees of fuzziness have been calculated for each grade location.

Fig. 5 indicates the degrees of uncertainties in the considered deposit by an uncertainty map with different levels of fuzziness. For a mining project, the described uncertainty levels above may be used in a production planning for financial based decision making.

5 Conclusion

By this paper, the degrees of uncertainty of regional grades have been appraised using the PCSM function, fuzzy set theory and metric distance. The methodology presented in the paper provides useful information for regional uncertainty assessments in an ore deposit. By this approach, uncertainty distribution in an ore deposit can be evaluated using an uncertainty map which provides a basis for future ore production planning. In particular, the information obtained may be used in a cut-off grade based financial investment as a decision criterion.

References

1. Bardossy, Gy., Fodor, J.: Evaluation of Uncertainties and Risks in Geology. Springer-Verlag, Heidelberg (2004).
2. Tercan, AE., Akcan E.: Assessment of uncertainty associated with grade tonnage curves using geostatistical simulation. IMM Section A-Mining Technology 113(2) (2004) 129-136.
3. Srivastava, RM.: Probabilistic modeling of ore lens geometry: An alternative to deterministic wireframes. Mathematical Geology 37(5) (2005) 513-544.
4. Zimmermann, H.-J.: An application–oriented view of modelling uncertainty. European Journal of Operational Research 122 (2000) 190-198.
5. Matheron, G.: Estimating and Choosing. Springer-Verlag, Berlin (1989).
6. Şen, Z.: Cumulative semivariogram model of regionalized variables, Mathematical Geology 21 (1989) 891-903.
7. Tutmez, B.: Reserve Estimation Using Fuzzy Set Approach, PhD Thesis, Hacettepe University, Ankara (2005).
8. Tutmez, B., Tercan, AE., Kaymak, U.: Fuzzy modelling for reserve estimation based on spatial variability, Mathematical Geology 39(1) (2007) (in press).
9. Klir, GJ., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice Hall, New Jersey (1995).
10. Ayyub, BM., Klir, GJ.: Uncertainty Modelling and Analysis in Engineering and the Sciences. Chapman Hall/CRC, Boca Raton (2006).
11. Klir, GJ., Folger, T.: Fuzzy Sets, Uncertainty, and Information. Prentice Hall, New Jersey (1988).

PC-TOPSIS Method for the Selection of a Cleaning System for Engine Maintenance

Mª Socorro Garcia[1], Mª Teresa Lamata[2], and J.L.Verdegay[2]

[1] Dpto de Electrónica, Tecnología de Computadoras y Proyectos.
Universidad de Politécnica de Cartagena. Murcia, España
socorro.garcia@upct.es

[2] Dpto. Ciencias de la Computación e Inteligencia Artificial.
Universidad de Granada.18071- Granada. España
mtl@decsai.ugr.es,
verdegay@decsai.ugr.es

Abstract. The purpose of this paper is to present an effective framework PC-TOPSIS when using both Analytic Hierarchy Process (AHP) to address the weights of the criteria, and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) extended to a fuzzy environmental for performing alternatives evaluation. Thus we could handle the decision problems in a more realist way. An example is provided to illustrate the application of the proposed PC-TOPSIS in a decision-making (DM) of selection of a cleaning system in a maintenance problem.

1 Introduction

Considerable attention has been focused recently upon effective extraction and encoding of the knowledge of experts. This is a well-known basic problem of Artificial Intelligence. By combining the advantages of the Pairwise Comparison (PC) as in the AHP to obtain the weight of the criteria together with the TOPSIS analysis, a new method (PC-TOPSIS) has been devised and is presented in this paper.

It is well known that a solid management program often involves conflicting economical, environmental and socio-ecological impacts. This is our case, in which we study a DM maintenance problem, the selection of a cleaning system for four stroke diesel engines maintenance for naval applications, electricity generating plants, etc. The selection problems have often been tackled using Multicriteria Decision-Making (MCDM), and in the last times several authors have proposed a version of MCDM that incorporates fuzzy set theory.

Natural language to express perception or judgement is always subjective, uncertain or vague. Since words are less precise than numbers, the concept of a linguistic variable approximately characterizes phenomena that are poorly defined to be described with conventional quantitative terms Herrera & Herrera-Viedma [12]. To resolve the vagueness, ambiguity and subjectivity of human judgment, fuzzy sets theory Zadeh [23], was introduced to express the linguistic terms in DM process. Bellman and Zadeh [3] were the first researchers to consider the DM problem using fuzzy sets.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 519–529, 2007.
springerlink.com

The AHP proposed by Saaty [19, 20] is a very popular approach to MCDM that involves qualitative data. It has been applied during the last twenty years in many situations of DM. The AHP has been used on a wide range of applications in a lot of different fields. The method uses a reciprocal decision matrix obtained by PC such that the information is given in a linguistic form.

In this context, the objective of this paper is to introduce an MCDM application in a maintenance DM problem where the company staff is only able or willing to provide information of a linguistic nature that we perform here by means of the fuzzy PC-TOPSIS approach.

This paper is organized as follows. Section 2 presents the fuzzy set theory. In section 3, the Analytic Hierarchy Process is proposed. Section 4 examines the TOPSIS approach. Section 5 presents the new method PC-TOPSIS. Section 6 illustrates an example with the new method. Finally, section 7 summarizes the work of this paper.

2 Fuzzy Set Theory

The fuzzy set theory, introduced by Zadeh [23] to deal with vague, imprecise and uncertain problems has been used as a modelling tool for complex systems that can be controlled by humans but are hard to define precisely. A collection of objects (universe of discourse) X has a fuzzy set A described by a membership function f_A with values in the interval $[0,1]$.

$$f_A : X \to [0,1]$$

Thus A can be represented as $A = \{f_A(x) \mid x \in X\}$. The degree that u belongs to A is the membership function $f_A(x)$.

The basic theory of the triangular fuzzy number is described as follows:

Definition 1. A real fuzzy number A is described as any fuzzy subset of the real line $\Re$ with membership function f_A which processes the following properties:

(a) $f_A(x)$ is a continuous mapping from $\Re$ to the closed interval [0, 1];
(b) $f_A(x) = 0$, for all $x \in$ $(-\infty, a]$;
(c) $f_A(x)$ is strictly increasing on $[a, b]$;
(d) $f_A(x) = w$, for all $x \in$ $[b,c]$;
(e) $f_A(x)$ is strictly decreasing on $[c, d]$;
(f) $f_A(x) = 0$, for all $x \in$ $(d, \infty]$,

where a, b, c, d are real numbers.

Definition 2. The fuzzy number A will be triangular if its membership function is given by:

$$f_A(x)=\begin{cases} f_A^L(x)=\dfrac{x-a}{b-a}, & a\le x\le b,\\ f_A^R(x)=\dfrac{x-c}{b-c}, & b\le x\le c,\\ 0, & otherwise, \end{cases} \tag{1}$$

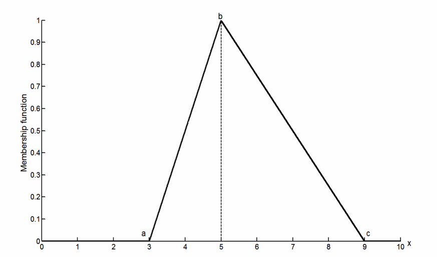


Fig. 1. Triangular fuzzy number

where *a, b* and *c* are real numbers. Then we designate *A* as the triangular fuzzy number. In this formula, *c* and *a* are the upper and lower values of the support of A , respectively, and *b* is the median value of A . The triangular fuzzy number is denoted as (a,b,c).

Definition 3. If T_1 and T_2 are two triangular fuzzy numbers defined by the triplets (a_1,b_1,c_1) and (a_2,b_2,c_2), respectively. Then, for this case, the necessary arithmetic operations with the fuzzy numbers are:

a) Addition: $T_1 \oplus T_2 = [a_1+a_2, b_1+b_2, c_1+c_2]$ (2)

c) Scalar multiplication $k \odot T = (k\cdot a, k\cdot b, k\cdot c)$ *if* $k>0$ (3)

d) Division $T_1 \oslash T_2 = \left[[a_1,b_1,c_1]\cdot[1/c_2, 1/b_2, 1/a_2]\right],\ 0 \ne [a_2,b_2,c_2]$ (4)

3 Analytic Hierarchy Process

The *AHP* is based on a firm theoretical foundation. The basic theory of *AHP* may be simplified as follows: Assume that we have *n* different and independent alternatives $(A_1, A_2, \ldots, A_n)$ and they have the weights $(w_1, w_2, \ldots, w_n)$, respectively. The decision-maker does not know in advance the values of w_i, $i=1,2,\ldots,n$, but he is capable of making pair-wise comparison between the different alternatives. Also, we assume that the quantified judgments provided by the decision-maker on pairs of alternatives (A_i, A_j) are represented in a $n\times n$ matrix $A = [a_{ij}]$. Let us $a_{ij} \approx (w_i/w_j)$ be an expert elicitation of intensity of preference of element A_i over A_j . For quantifying these values a_{ij} Saaty uses the scale that we transcribe in Table 1.

Having recorded the numerical judgments a_{ij} in the matrix A , the problem now is to recover the numerical weights $(w_1, w_2, ..., w_n)$ of the alternatives from this matrix. Saaty's method computes w as the principal right eigenvector of the matrix A , that is,

$$Aw = \lambda_{\max} w \tag{5}$$

where $\lambda_{\max}$ is the principal eigenvalue of the matrix A . If matrix A is a positive reciprocal one then $\lambda_{\max} \geq n$.

Table 1. Values for Saaty´s Analytic Hierarchy Process

Verbal judgements of preferences between alternative *i* and alternative *j*	**Numerical rating**
A_i is equally important than A_j	1
A_i is slightly more important than A_j	3
A_i is strongly more important than A_j	5
A_i is very strongly more important than A_j	7
A_i is extremely more important than A_j	9
Intermediate values	2,4,6,8

The judgments of the decision-maker are perfectly consistent as long as

$$a_{ij} a_{jk} = a_{ik},\ i, j, k = 1, 2, \ldots, n \tag{6}$$

The eigenvector method yields a natural measure of consistency. Saaty defined the consistency ratio as

$$CR = \frac{CI}{RI} \tag{7}$$

where RI is the average value of CI for random matrices using the Saaty scale given in table 1.

Table 2. Table of the random index for several matrices dimensions

n	1-2	3	4	5	6	7	8	9
RI	0,00	0.52	0.89	1.11	1.25	1.35	1.45	1.49

One matrix is accepted as consistent if $CR \leq 0.1$, in other case the matrix is rejected as an inconsistent matrix.

4 TOPSIS Approach

The TOPSIS approach is a MCDM method for the arrangement of preferences to an ideal solution by similarity, it was developed by Hwang and Yoon [13] 1981 and improved by the own authors in 1987 and 1992, also Zeleny [28], Lai et al [17] and many others have worked in this theme. TOPSIS is a MCDM method that identifies the solutions for a finite set of alternatives. The basic principle is that the chosen alternative must have the shortest distance to the positive ideal solution (PIS) and the longest distance to the negative ideal solution (NIS).

Under many conditions, crisp data are inadequate to model real-life situations. Since human judgements including preferences are often vague and cannot estimate his preference with an exact numerical value. In this paper, we further extend the concept of TOPSIS to develop a methodology for cleaning system selection problems in fuzzy environment Chen [7]. Considering the fuzziness in the decision data, linguistic variables are used to assess the ratings of each alternative with respect to each criterion. We can convert the decision matrix into a fuzzy decision matrix.

According to the concept of TOPSIS, we define the Fuzzy-PIS and Fuzzy-NIS. Finally, a closeness coefficient of each alternative is defined to determine the ranking order of all alternatives. The higher value of closeness coefficient indicates that an alternative is closer to Fuzzy-PIS and farther from Fuzzy-NIS simultaneously. TOPSIS procedure can be expressed in a series of steps that can turn in Chen and Hwang [9] and that we summarize in:

Step 1. Identify the evaluation criteria and the appropriate linguistic variables for the importance weight of the criteria and determine the set of feasible alternatives with the linguistic score for alternatives in terms of each criterion.

Step 2. Construction of the (fuzzy) decision matrix and transfer this matrix to the normalized one. Construct the weighted normalized decision matrix and define the positive ideal $\overline{A}_j^+$ and negative ideal $\overline{A}_j^-$ solutions, j=1,2,…,n.

Step 3. Determine the distances, d_i^+ and d_i^-, i=1,2,...,m and the ranking of the alternatives by means of $\overline{R}_j = \dfrac{\overline{d}_j^-}{\overline{d}_j^+ + \overline{d}_j^-}$, $j = 1,\ldots,m$.

5 PC_TOPSIS

This new method consists in the evaluation of a decision problem where the weights of the criteria have been obtained from the $n(n-1)/2$ pair comparisons in the AHP and the eigenvector method as estimator.

From here we obtain two solutions; taking into account the normalized matrix in the TOPSIS approach. In this point, it is possible to introduce a step to obtain the mathematical expectation as a first solution, the other one is the own TOPSIS solution.

The scheme of the PC-TOPSIS is resumed in the Fig 2.

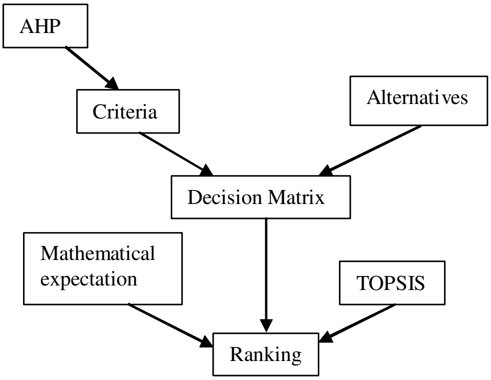


Fig. 2. PC-TOPSIS

6 Illustrative Example: Decision Aid in the Selection of a Cleaning System

The theoretical approach outlined above has been applied to the development of a decision aid in the selection of a cleaning system. The hierarchy in the sense of Saaty is presented in fig 2

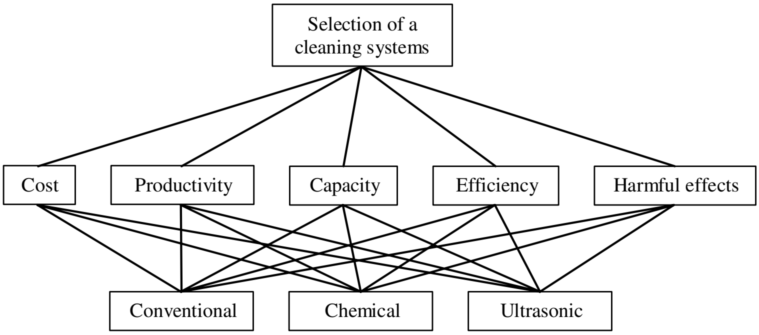


Fig. 3. Hierarchy

Step 1. *Assignment of the scores against the measures*

In this problem, we have one decision-maker, five criteria C_1 = Total annual operation cost, C_2 = System productivity, C_3 = System load capacity, C_4 = Cleaning efficiency and C_5 = Harmful effects and three alternatives, A_1 = Conventional cleaning, A_2 = Chemical cleaning and A_3 = Ultrasonic cleaning. Let assume us that decision-maker uses the linguistic score set S = {VP, F, MG, VG}, where *VP*= Very poor = [0,1,1.9], *F*= Fair = [4.2,4.8,5], *MG* = Medium good = [5,5.8.6.8], *VG*=Very good = [7.9,9,10] to evaluate the suitability of each alternative under each of the criteria.

It is important to consider that the criteria can have different importance. In the case that occupies to us and since we dealt with linguistic labels one via is the use of the model of PC. We assume that the decision-maker employes linguistic values as in the AHP approach to obtain the relative criteria weight; in this case all this information is expressed in table 3 taking into account the values expressed in table1.

Table 3. Criteria weight

	C1	**C2**	**C3**	**C4**	**C5**	**Weight**
C1	1	1	4	3	3	**0.3461**
C2	1	1	3	2	3	**0.2975**
C3	1/4	1/3	1	1/4	1/2	**0.0686**
C4	1/3	1/2	4	1	2	**0.1812**
C5	1/3	1/3	2	1/2	1	**0.1066**

With $\lambda_{\max}$ =5.1475, CI= 0.03675 and CR= 0.033<0.10. Then this matrix is accepted.

Step 2. *Determination of the ratings using the mathematical expectation and the TOPSIS approaches*

Determining the Performance Matrix. The system manager can define the ratings of the three alternatives under all criteria (table 4). Then, also the system manager can define his range for the linguistic variables employed in this problem that are based on his subjective judgments within a graphical scale to determine the performance value to each alternative respect to each criterion.

Table 4. Decision matrix

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & \mathrm{C}_5 \\ \left(\begin{array}{c} \text{Medium Good} \\ \text{Medium Good} \\ \text{Fair} \end{array}\right. & \begin{array}{c} \text{Medium Good} \\ \text{Fair} \\ \text{Very Good} \end{array} & \begin{array}{c} \text{Fair} \\ \text{Very Good} \\ \text{Medium Good} \end{array} & \begin{array}{c} \text{Fair} \\ \text{Medium Good} \\ \text{Medium Good} \end{array} & \left.\begin{array}{c} \text{Medium Good} \\ \text{Very Poor} \\ \text{Medium Good} \end{array}\right) \end{array}$$

The raw data are normalized, by means of (8) and taking into account the operations in (2,4)

$$\bar{n}_{ij} = \underset{\sim}{x}_{ij} \Big/ \sqrt{\sum_{j=1}^{m} \left(\underset{\sim}{x}_{ij}\right)^2}\,,\ j = 1,\ldots,m,\ i = 1,\ldots,n \qquad (8)$$

with the object to eliminate anomalies with different measurement units and scales and it is show in table 5.

Then the weight vector is incorporated into the normalized decision matrix using equation (9) to obtain the weighted normalized decision matrix, and taking into account (3).

Table 5. Normalized decision matrix

	Weights	A_1	A_2	A_3
C_1	0.3461	[0.4613,0.6103,0.8268]	[0.4613,0.6103,0.8268]	[0.3875,0.5051,0.6080]
C_2	0.2975	[0.3821,0.4943,0.6635]	[0.3210,0.4091,0.4878]	[0.6037,0.7670,0.9757]
C_3	0.0686	[0.3210,0.4091,0.4878]	[0.6037,0.7670,0.9757]	[0.3821,0.4943,0.6635]
C_4	0.1812	[0.3875,0.5051,0.6080]	[0.4613,0.6103,0.8268]	[0.4613,0.6103,0.8268]
C_5	0.1066	[0.5101,0.7019,0.9617]	[0,0.1210,0.2687]	[0.5101,0.7019,0.9617]

$$\bar{v}_{ij} = \underset{\sim}{w}_j \odot \bar{n}_{ij},\ j=1,\ldots,n,\ i=1,\ldots,m, \tag{9}$$

Table 6. Weighted normalized decision matrix

	A_1	A_2	A_3
C_1	[0.1597,0.2112,0.2862]	[0.1597,0.2112,0.2862]	[0.1341,0.1748,0.2104]
C_2	[0.1137,0.1471,0.1974]	[0.0955,0.1217,0.1451]	[0.1796,0.2282,0.2903]
C_3	[0.0220,0.0281,0.0335]	[0.0414,0.0526,0.0669]	[0.0262,0.0339,0.0455]
C_4	[0.0702,0.0915,0.1102]	[0.0836,0.1106,0.1498]	[0.0836,0.1106,0.1498]
C_5	[0.0544,0.0748,0.1025]	[0,0.0129,0.0286]	[0.0544,0.0748,0.1025]
$E[A_i]$	**[0.4199,0.5527,0.7297]**	**[0.3801,0.5090,0.6767]**	**[0.4779,0.6223,0.7985]**

Many methods to order alternatives in a DM problem with uncertainty can be found, but one of the most used in the literature is the Mathematical Expectation.

$$E[A_i] = \sum_{j=1}^{n} \bar{v}_{ij} = \underset{\sim}{w}_j \odot \bar{n}_{ij},\ j=1,\ldots,n,\ i=1,\ldots,m, \tag{10}$$

where $E[A_i]$ is the final score for alternative *i*. We represent in table 7 and fig 4, being the final ranking associated with the alternatives the following

$$A_3 \succ A_2 \succ A_1$$

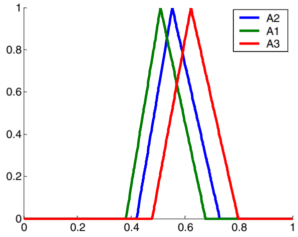


Fig. 4. Mathematical expectation

Determination of the Fuzzy-PIS and Fuzzy-NIS. In this problem we take the $\max_i \bar{v}_{ij}\ \forall i$=1,2,3 for v_j^+ and the $\min_i \bar{v}_{ij}\ \forall$i=1,2,3 for v_j^-, being the corresponding values in table 8.

Table 7. Mathematical expectation

	E[A$_i$]	**Ranking**
A$_1$	[0.4199,0.5527,0.7297]	2
A$_2$	[0.3801,0.5090,0.6767]	3
A$_3$	[0.4779,0.6223,0.7985]	1

Table 8. Fuzzy-PIS and Fuzzy-NIS for each criterion

v_j^+	v_j^-
[0.1597,0.2112,0.2862]	[0.1341,0.1748,0.2104]
[0.1796,0.2282,0.2903]	[0.0955,0.1217,0.1451]
[0.0414,0.0526,0.0669]	[0.0220,0.0281,0.0335]
[0.0836,0.1106,0.1498]	[0.0702,0.0915,0.1102]
[0.0544,0.0748,0.1025]	[0,0.0129,0.0286]

Table 9. Distance of each alternative to Fuzzy-PIS and Fuzzy-NIS

	A$_1$	*A$_2$*	*A$_3$*
Positive distance	[0.0700,0.0869,0.1064]	[0.1002,0.1232,0.1628]	[0.0297,0.0409,0.0787]
Negative distance	[0.0628,0.0762,0.1180]	[0.0348,0.0479,0.0918]	[0.1011,0.1248,0.1680]
TOPSIS	**[0.2797,0.4672,0.8887]**	**[0.1365,0.2799,0.6806]**	**[0.4099,0.7530,1.2841**

Step 3. *Ranking*

According to the closeness coefficients of the three alternatives (cleaning machines) we know that the ranking observed in fig 4 and calculated in table 9 is

$$A_3 \succ A_2 \succ A_1$$

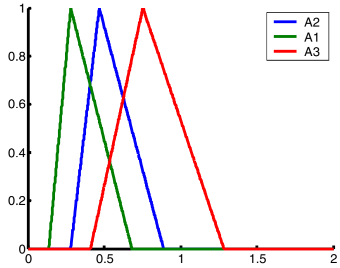


Fig. 5. TOPSIS Results

Table 10. TOPSIS Results

	TOPSIS	**Ranking**
A$_1$	[0.2797,0.4672,0.8887]	2
A$_2$	[0.1365,0.2799,0.6806]	3

A_3	[0.4099,0.7530,1.2841]	1

7 Conclusions

This study, presents a scientific framework to assess the selection of a cleaning system for four stroke diesel engines maintenance, uses triangular fuzzy numbers to express linguistic variables that consider the subjective judgments of decision-maker and adopts fuzzy MCDM approach to synthesize the decision. PC-TOPSIS extended to a fuzzy environment is utilized to determine overall performance value and to rank the alternatives cleaning systems.

In this paper, we have studied a linguistic MCDM problem because only linguistic information (as the type of data) was available from the company. It was attempted to approach the problem trying to simulate the behaviour of the decision-maker, so that by means of surveys the expert's knowledge was obtained as well as his disposition towards the problem. Several alternatives have been considered and evaluated in terms of many different conflicting criteria in an optimal selection cleaning system problem, leading to a large set of subjective or ambiguous data. It has been also shown as the ranking obtained by the two approaches are the same.

This way of approaching the problem meant that we could handle the decision problems in a more real way, in that the methodologies implemented in a computer turn out to be very interesting from the point of view of Artificial Intelligence when we try to simulate human behaviour.

Acknowledgements

This work is partially supported by the Spanish DGICYT under projects TIN2005-02418, TIN2004-21700-E, TIN2005-24790-E.

References

1. Alonso J.A., Lamata M.T.: Consistency in the analytic hierarchy process. a new approach. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 14, (2006) 445-459.
2. Al Harbi K. M. A.: Application of the AHP in project management. *International Journal of Project Management* 19, (2001) 19-27.
3. Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. *Management Sci.*, 17, (1970) 141--164.
4. Braglia, M., Frosolini, M., Montanari, R.: Fuzzy TOPSIS approach for failure mode, effects and criticality analysis. Quality and Reliability Engineering International, 19, (2003), 425–443
5. Chan F.T.S, Chan M.H, Tang N.K.H.: Evaluation methodologies for technology selection. *Journal of Materials Processing Technology* 107, (2000a) 330–337.
6. Chan F.T.S., Jiang B., Tang N.K.H.: The development of intelligent decision support tools to aid the design flexible manufacturing systems. *International Journal of Production Economics* 65, (2000b) 73–84.

7. Chen C.T.: Extensions of the TOPSIS for group decision-making under fuzzy environment, *Fuzzy Sets and Systems*, volume 114, (2000) 1-9.
8. Chen, M. F., Tzeng, G. H.: Combing grey relation and TOPSIS concepts for selecting an expatriate host country. *Mathematical and Computer Modelling*, 40, (2004) 1473–1490.
9. Cheng C.-H., Yang K.-L., Hwang C.-L.: Evaluating attack helicopters by AHP based on linguistic variable weight. *European Journal of Operation al Research* 116, (1999) 423–435.
10. Chu, T. C.: Selecting plant location via a fuzzy TOPSIS approach. *International Journal of Advanced Manufacturing Technology*, 20, (2002) 859–864.
11. Chu, T. C., Lin, Y. C.: A fuzzy TOPSIS method for robot selection. *International Journal of Advanced Manufacturing Technology,* 21,(2003) 284–290.
12. Herrera, F., Herrera-Viedma, E.: Linguistic decision analysis: steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, 115, (2002) 67-82.
13. Hwang C.L., Yoon K.: Multiple Attibute Decision Methods and Applications. Springer, Berlin Heidelberg (1981).
14. Kablan M. M.: Decision support for energy conservation promotion: an analytic hierarchy process approach. *Energy Policy*, 32, (2004) 1151-1158.
15. Kacprzyk, J., Yager, R.R..: Linguistic summaries of data using fuzzy logic. *International Journal of General Systems,* 30, (2001) 133-154.
16. Kerre E.E.: The use of fuzzy set theory in electrocardiological diagnostics, in : M.M. Gupta and E. Sanchez, Eds., Approximate Reasoning in Decision Analysis (North-Holland, Amsterdam, (1982) 277-282
17. Lai Y.J., Liu T.Y., Hwang C.L.: TOPSIS for MODM, *European Journal of Operational* Research, volume 76, (1994) 486-500.
18. Luce R.D., Raiffa H.: *Games and Decisions: Introduction and Critical Survey*. New York: John Wiley and Sons (1967).
19. Saaty T.L.: The Analytic Hierarchy Process. Macgraw Hill (1980).
20. Saaty.T.L.: Group Decision Making and the AHP. Springer Verlag. New York (1989).
21. Shim, J.P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R., Carlsson, C.: Past, present, and future of decision support technology. Decision Support Systems 33, (2002) 111–126.
22. Triantaphyllou E.: *Multi-Criteria decision making methods: A comparative study*. The Netherlands: Kluwer Academic (2000).
23. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning: Part 1. *Information Sciences* 8, (1975) 199-249.
24. Zadeh, L.A., Kacprzyt, J.: *Computing with Words in Information / Intelligent Systems*. Part 1. Foundations, Physica-Verlag (Springer-Verlag), Heidelberg and New York (1999*).*
25. Zadeh, L.A., Kacprzyt, J.: *Computing with Words in Information/ Intelligent Systems*. Part 2. Foundations, Physica-Verlag (Springer-Verlag), Heidelberg and New York (1999).
26. Zeleny, M.: Multiple Criteria Decision Making, McGraw-Hill, New York (1982).

Coordination Uncertainty of Belief Measures in Information Fusion

Zhenyuan Wang[1] and George J. Klir[2]

[1] Department of Mathematics, University of Nebraska at Omaha, Omaha, NE 68182, USA
zhenyuanwang@mail.unomaha.edu
[2] Department of Systems Science and Industrial Engineering, Thomas J. Watson School of Engineering, Binghamton University-SUNY, Binghamton, NY 13902, USA

Abstract. Fuzzy measures are used to describe interactions among given attributes towards a certain target. Given an observation as a function defined on the set of attributes, an integral of the function with respect to a given fuzzy measure is usually taken as an aggregation tool in information fusion. Various types of integrals represent different coordination modes of attributes. Thus, in case the coordination mode is unknown, due to the nonadditivity of the fuzzy measure, the integral value is uncertain and varies according to the type of the integral. A commonly used type of fuzzy measures are believe measures. The coordination uncertainty associated with belief measures in information fusion is discussed. Furthermore, the behavior of coordination uncertainty under the Dempster rule of combination is investigated in this paper.

1 Introduction

Fuzzy measures and relevant nonlinear integrals have been widely applied in information fusion and data mining. In information fusion, each information source is regarded as an attribute and a fuzzy measure [4, 5, 6] can be used to describe the interaction among the contribution rates of attributes towards the fusion target [6, 7]. Then, an integral, which is usually nonlinear, is used as an aggregation tool to fuse the received information from attributes to a single real number. Various types of integrals, such as the Choquet integral, the lower integral, the upper integral, and even the classical Lebesgue integral (ignoring the nonadditivity of fuzzy measures), can be chosen for this purpose. These types of integrals represent different coordination modes of attributes when they contribute towards the target. Hence, if the coordination mode is not known, the fusion value has some coordination uncertainty. Such an uncertainty results from the nonadditivity of the fuzzy measure employed. Belief measures are commonly-used special types of fuzzy measures. This paper discusses the coordination uncertainty associated with belief measures in information fusion. Furthermore, the behavior of coordination uncertainty under the Dempster-Shafer combination rule [3] is investigated.

The paper is organized as follows. Section 2 recalls some preliminaries regarding fuzzy measures and belief measures. The various types of nonlinear integrals with respect to fuzzy measures are summarized in Section 3. Based on nonlinear integrals, the coordination uncertainty associated with fuzzy measures and belief measures in information fusion is analyzed and estimated in Section 4. In Section 5, the behavior of coordination uncertainty under the Dempster combination rule is investigated.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 530–538, 2007.
springerlink.com

2 Preliminary Knowledge on Belief Measures

Let $X = \{x_1, x_2, \cdots, x_n\}$ be the set of all considered information sources. Then (X, $\mathrm{P}(X)$) is a measurable space, where $\mathrm{P}(X)$ is the power set of X. Each information source x_i, $i = 1, 2, \cdots, n$, is also called an attribute. We assume that all sets involved in this paper are subsets of X.

Definition 1. Set function μ: $\mathrm{P}(X) \to [0, \infty)$ is called a fuzzy measure iff:

(FM1) $\mu(\varnothing) = 0$;

(FM2) $\mu(A) \le \mu(B)$ if $A \subseteq B$.

A fuzzy measure μ is normalized if $\mu(X) = 1$.

Generally, a fuzzy measure may not be additive. Its nonadditivity describes the interaction among the contribution rates from attributes towards a certain fusion target [7].

Definition 2. Normalized fuzzy measure $\mu : \mathrm{P}(X) \to [0, 1]$ is called a belief measure, denoted usually by *Bel*, iff

$$\text{(B1)}\quad \mu(\bigcup_{i=1}^{k} E_i) \ge \sum_{I \subseteq \{1, \cdots, k\},\, I \ne \varnothing} (-1)^{|I|-1} \mu(\bigcap_{i \in I} E_i).$$

for any nonempty set class $\{E_1, E_2, \cdots, E_k\}$.

Condition (B1) means that any belief measure is strongly superadditive [3, 5, 12].

Definition 3. Set function m: $\mathrm{P}(X) \to [0, 1]$ is called a basic probability assignment iff $m(\varnothing) = 0$ and $\sum_{E \subseteq X} m(E) = 1$.

Set function m is not, in general, a probability measure on $\mathrm{P}(X)$. However, if we define p on the class of all singleton consisting of only one set in $\mathrm{P}(X)$ by $p(\{E\}) = m(E)$ for any $E \subseteq X$, then p is a probability density on $\mathrm{P}(X)$ and, therefore, can be uniquely extended to a probability measure on $\mathrm{P}(\mathrm{P}(X))$.

It is well known [3, 5] that there is a one-to-one correspondence between belief measures and basic probability assignments described as follows.

1. If $Bel : \mathrm{P}(X) \to [0, 1]$ is a belief measure, then $m : \mathrm{P}(X) \to [0, 1]$ defined by

$$m(E) = \sum_{F \subseteq E} (-1)^{|E-F|} Bel(F) \quad \forall E \in \mathrm{P}(X)$$

is a basic probability assignment.

2. If $m : \mathrm{P}(X) \to [0, 1]$ is a basic probability assignment, then $Bel : \mathrm{P}(X) \to [0, 1]$ defined by

$$Bel(E) = \sum_{F \subseteq E} m(E) \quad \forall E \in \mathrm{P}(X)$$

is a belief measure.

Example 1. Let $X = \{x_1, x_2, x_3\}$. Fuzzy measure $\mu : \mathrm{P}(X) \to [0, 1]$ is given as $\mu(\varnothing) = 0$, $\mu(\{x_1\}) = 0.1$, $\mu(\{x_2\}) = 0.15$, $\mu(\{x_3\}) = 0.35$, $\mu(\{x_1, x_2\}) = 0.25$, $\mu(\{x_1, x_3\}) = 0.5$, $\mu(\{x_2, x_3\}) = 0.75$, and $\mu(X) = 1$. Set function μ is a belief measure. Its corresponding basic probability assignment is $m : \mathcal{P}(X) \to [0, 1]$ with $m(\emptyset) = 0$, $m(\{x_1\}) = 0.1$, $m(\{x_2\}) = 0.15$, $m(\{x_3\}) = 0.35$, $m(\{x_1, x_2\}) = 0$, $m(\{x_1, x_3\}) = 0.05$, $m(\{x_2, x_3\}) = 0.25$, and $m(X) = 0.1$.

3 Integrals Used in Information Fusion

A traditional aggregation tool in information fusion is the weighted average, i.e., $y = \sum_{i=1}^{n} w_i f(x_i)$, where y is the fusion value, $f(x_i)$ is the information amount received from attribute x_i, w_i is the weight of x_i, $i = 1, 2, \cdots, n$, satisfying $\sum_{i-1}^{n} w_i = 1$.

The weighted average can be regarded as a Lebesgue's integral [2]: $y = \int f \, d\mu$, where the integrand $f : X \to (-\infty, \infty)$ is a function whose value at attribute x_i is $f(x_i)$ and μ: $\mathcal{P}(X) \to [0, 1]$ is a classical additive measure determined from the weights $(w_1, w_2, \cdots, w_n)$ by

$$\mu(E) = \sum_{i : x_i \in E} w_i \quad \forall E \in \mathcal{P}(X).$$

In such a fusion model, we need a basic assumption that there is no interaction among the contribution rates of attributes towards y so that the global contribution from a group of attributes is just the sum of the contributions from each individual attribute in the group. However, when the interaction cannot be ignored, a fuzzy measure should be used to replace the weights and describe the above-mentioned interaction. In this case, the Lebesgue integral as an aggregation tool fails due to the nonadditivity of the fuzzy measure, and we have to use other types of integrals, nonlinear integrals, that are generalizations of the Lebesgue integral. One of the commonly-used nonlinear integrals is the Choquet integral [1].

Definition 4. The Choquet integral of function $f : X \to (-\infty, \infty)$ with respect to fuzzy measure μ is defined by

$$(\mathrm{C})\int f \, d\mu = \int_{-\infty}^{0} [\mu(F_\alpha) - \mu(X)] d\alpha + \int_{0}^{\infty} \mu(F_\alpha) d\alpha$$

if the two Riemann integrals on the right-hand side are not both infinite, where $F_\alpha = \{x \mid f(x) \geq \alpha\}$ is the α -level set of *f* for $\alpha \in (-\infty, \infty)$.

When function *f* is nonnegative, we have other choices for the type of integral to be used as an aggregation tool in information fusion. In the literature, the upper integral and the lower integral have been proposed for this purpose [7, 8].

Definition 5. The upper integral of $f : X \to [0, \infty)$ with respect to fuzzy measure μ, denoted by $(\mathrm{U})\int f \, d\mu$, is defined by

$$(\mathrm{U})\int f \, d\mu = \sup\{\sum_{j=1}^{2^n - 1} a_j \cdot \mu(A_j) \mid \sum_{j=1}^{2^n - 1} a_j \chi_{A_j} = f\},$$

while the lower integral of *f* with respect to μ, denoted by $(\mathrm{L})\int f\ d\mu$, is defined by

$$(\mathrm{L})\int f\ d\mu = \inf\{\sum_{j=1}^{2^n-1} a_j \cdot \mu(A_j) \mid \sum_{j=1}^{2^n-1} a_j \chi_{A_j} = f\},$$

where $a_j \geq 0$ and $A_j = \bigcup_{i:j_i=1}\{x_i\}$ if *j* is expressed in binary digits as $j_n j_{n-1} \cdots j_1$ for every $j = 1, 2, \cdots, 2^n - 1$.

The value of $(\mathrm{U})\int f\ d\mu$ is the solution of the following linear programming problem [9], where $a_1, a_2, \cdots, a_{2^n-1}$ are unknown parameters:

$$\begin{aligned} &\text{maximize } z = \sum_{j=1}^{2^n-1} a_j \cdot \mu_j \\ &\text{subject to } \sum_{j=1}^{2^n-1} a_j \chi_{A_j}(x_i) = f(x_i),\ i = 1, 2, \cdots, n \\ &\qquad a_j \geq 0,\ j = 1, 2, \cdots, 2^n - 1, \end{aligned}$$

in which $\mu_j = \mu(A_j)$ for $j = 1, 2, \cdots, 2^n - 1$. Similarly, the value of $(\mathrm{L})\int f\ d\mu$ is the solution of the linear programming problem:

$$\begin{aligned} &\text{minimize } z = \sum_{j=1}^{2^n-1} a_j \cdot \mu_j \\ &\text{subject to } \sum_{j=1}^{2^n-1} a_j \chi_{A_j}(x_i) = f(x_i),\ i = 1, 2, \cdots, n \\ &\qquad a_j \geq 0,\ j = 1, 2, \cdots, 2^n - 1. \end{aligned}$$

For a given fuzzy measure $\mu : \mathrm{P}(X) \to [0, \infty)$, we may introduce an additive measure $\mu' : \mathrm{P}(X) \to [0, \infty)$ by

$$\mu'(E) = \sum_{i:x_i \in E} \mu(\{x_i\}) \quad \forall E \in \mathrm{P}(X).$$

Measure μ' coincides with fuzzy measure μ on the class of all singletons in $\mathrm{P}(X)$. Thus, the Lebesgue integral $\int f\ d\mu'$ can be regarded as a special nonlinear integral of *f* with respect to fuzzy measure μ, still called the Lebesgue integral of *f* and denoted as $\int f\ d\mu$ in this paper if there is no confusion.

Generally, given a fuzzy measure $\mu : \mathrm{P}(X) \to [0, \infty)$ and a function $f : X \to [0, \infty)$, any type of integral of *f* with respect to μ is just a rule, denoted by *r*, by which *f* can be decomposed to become a set function $\pi : \mathrm{P}(X) \to [0, \infty)$ with $\pi(\varnothing) = 0$ satisfying

$$f(x) = \sum_{A: x \in A \subseteq X} \pi(A) \quad \forall x \in X.$$

π is called a partition of *f*. Then, regarding both μ and π as 2^n-dimensional vectors, the value of the *r*-integral of *f* with respect to μ is

$$(\mathrm{r})\int f\ d\mu = \pi \cdot \mu\,.$$

Here $\pi \cdot \mu$ is the inner product of vectors π and μ .

Different decomposing rules represent different coordination modes of attributes in the fusion and, therefore, yield different types of integrals. These integrals are generally nonlinear with respect to the integrand *f*. The Lebesgue integral and the Choquet integral are two special types of *r*-integrals and form a pair of extreme *r*-integrals with respect to the degree of coordination among the attributes, while the upper integral and the lower integral are another pair of extreme *r*-integrals with respect to the fusion value. Any *r*-integral is a generalization of the Lebesgue integral, i.e., when the fuzzy measure is additive, all *r*-integrals coincide with the Lebesgue integral [7]. It is not difficult to show that

$$(\mathrm{L})\int f\ d\mu \le (\mathrm{r})\int f\ d\mu \le (\mathrm{U})\int f\ d\mu$$

for any fuzzy measure $\mu : \mathrm{P}(X) \to [0, \infty)$ and any function $f : X \to [0, \infty)$ [7].

Example 2. Let $X = \{x_1, x_2, x_3\}$. Fuzzy measure $\mu : \mathrm{P}(X) \to [0, \infty)$ is given as $\mu(\varnothing) = 0$, $\mu(\{x_1\}) = 5$, $\mu(\{x_2\}) = 6$, $\mu(\{x_3\}) = 7$, $\mu(\{x_1, x_2\}) = 14$, $\mu(\{x_1, x_3\}) = 8$, $\mu(\{x_2, x_3\}) = 10$, and $\mu(X) = 16$. For function

$$f(x) = \begin{cases} 4 & \text{if } x = x_1 \\ 8 & \text{if } x = x_2 \\ 2 & \text{if } x = x_3 \end{cases},$$

we have $\int f\ d\mu = 82$, $(\mathrm{C})\int f\ d\mu = 84$, $(\mathrm{L})\int f\ d\mu = 74$, and $(\mathrm{U})\int f\ d\mu = 94$. Their corresponding partitions of *f* are listed in Table 1.

Table 1. Partitions of *f* in Example 2

	Lebesque integral	Choquet integral	Lower integral	Upper integral
$\pi(\varnothing)$	0	0	0	0
$\pi(\{x_1\})$	4	0	2	0
$\pi(\{x_2\})$	8	4	8	4
$\pi(\{x_3\})$	2	0	0	2
$\pi(\{x_1,x_2\})$	0	2	0	4
$\pi(\{x_1,x_3\})$	0	0	2	0
$\pi(\{x_2,x_3\})$	0	0	0	0
$\pi(X)$	0	2	0	0

4 Coordination Uncertainty Associated with Belief Measures

Let a fuzzy measure $\mu : \mathrm{P}(X) \to [0, \infty)$ and function $f : X \to [0, \infty)$ be given. Since different coordination modes correspond to different types of integrals, the value of the integral is uncertain if the coordination mode in the fusion procedure is unknown. Such a coordination uncertainty results from the nonadditivity of the given fuzzy measure. This uncertainty can be measured by its degree defined as follows [6].

Definition 6. The degree of coordination uncertainty associated with fuzzy measure μ is defined by

$$\gamma_\mu = \frac{(\mathrm{U})\int 1\, d\mu - (\mathrm{L})\int 1\, d\mu}{\mu(X)} .$$

When μ is an additive measure, the upper integral coincides with the lower integral and, therefore, $\gamma_\mu = 0$. In fact, from the additivity of μ, we know that all r-integrals coincide with the Lebesgue integral and, therefore, no uncertainty exists in the fusion procedure.

Since any belief measure is strongly superadditive, we have the following theorem [12].

Theorem 1. Let $Bel : \mathrm{P}(X) \to [0, 1]$ be a belief measure. Then $(\mathrm{L})\int f\ dBel = \int f\ dBel$ and $(\mathrm{U})\int f\ dBel = (\mathrm{C})\int f\ dBel$ for any function $f : X \to [0, \infty)$.

Thus, the degree of coordination uncertainty associated with a belief measure *Bel* can be easily calculated as follows.

$$\begin{aligned}\gamma_{Bel} &= \frac{(\mathrm{U})\int 1\, dBel - (\mathrm{L})\int 1\, dBel}{Bel(X)} \\ &= \frac{(\mathrm{C})\int 1\, dBel - \int 1\, dBel}{1} \\ &= 1 - \sum_{i=1}^{n} Bel(\{x_i\}) \\ &= 1 - \sum_{i=1}^{n} m(\{x_i\}) .\end{aligned}$$

Now, it follows directly from Definition 3 that $\gamma_{Bel} \in [0, 1]$.

Example 3. We use the belief measure μ given in Example 1. The degree of uncertainty carried by μ is

$$\gamma_\mu = 1 - \sum_{i=1}^{3} \mu(\{x_i\}) = 1 - 0.6 = 0.4 .$$

The degree of coordination uncertainty associated with a belief measure depends on how many nonzero values of the basic probability assignment focus on the singletons in P(X). For any belief measure, the more the basic probability assignment focuses on singletons, the lower the degree of the coordination uncertainty. As an extreme case, when the basic probability assignment focuses only on the singletons, there is no coordination uncertainty. In fact, in this case, the belief measure collapses to a probability measure.

5 Coordination Uncertainty Under Dempster Rule of Combination

As one of the major topics discussed in [3], given two basic probability assignments, m_1 and m_2, on the power set of finite universal set *X*, Dempster's combination rule produces a new basic probability assignment *m* as follows.

$$m(C) = \frac{\sum\limits_{A \cap B = C} m_1(A) \cdot m_2(B)}{1 - \sum\limits_{A \cap B = \varnothing} m_1(A) \cdot m_2(B)}$$

for any nonempty set *C*. Thus, given two belief measures, Bel_1 and Bel_2, on P(*X*), we can "combine" them and obtain a new belief measure *Bel* with basic probability assignment *m* that comes from m_1 and m_2 via Dempster's combination rule. In this rule, to guarantee that the combined set function *m* is still a basic probability assignment, as mentioned in [3], the only remedy is to use a factor

$$(1 - \sum_{A \cap B = \varnothing} m_1(A) \cdot m_2(B))^{-1}$$

for normalizing $\sum\limits_{A \cap B = C} m_1(A) \cdot m_2(B)$. However, this normalization is not an ideal action for information processing and reasoning since it modifies the evidence expressed by m_1 and m_2. It leads to an over-belief on some sets. From the view point of the coordination uncertainty discussed in this paper, it can easily be shown that, after combining two belief measures, the coordination uncertainty may be either enlarged or reduced depending on what sets are over-believed. The following is an example showing that a set containing more than one attributes is over-believed when Dempster's combination rule is used to combine two basic probability assignments.

Example 4. Let $X = \{a, b, c, d\}$,

$$m_1(A) = \begin{cases} 0.1 & \text{if } A = \{a, b\} \\ 0.9 & \text{if } A = \{c\} \\ 0 & \text{otherwise} \end{cases},$$

and

$$m_2(A) = \begin{cases} 0.1 & \text{if } A = \{a, b\} \\ 0.9 & \text{if } A = \{d\} \\ 0 & \text{otherwise} \end{cases}.$$

From the Dempster's combination rule, we have

$$m(A) = \begin{cases} 1 & \text{if } A = \{a, b\} \\ 0 & \text{otherwise} \end{cases}.$$

Here, set $\{a, b\}$ containing two attributes is over-believed and, therefore, the coordination uncertainty of the belief measure determined by *m* is enlarged in comparison with the uncertainties γ_1 and γ_2 of the belief measures determined by m_1 and m_2 respectively. In fact, $\gamma_1 = \gamma_2 = 1 - 0.9 = 0.1$, while $\gamma = 1 - 0 = 1$.

The next example shows an opposite case, i.e., some singletons of attributes are over-believed such that the coordination uncertainty of the belief measure determined by the combined basic probability assignment is excessively reduced.

Example 5. Let $X = \{a, b, c, d, e\}$,

$$m_1(A) = \begin{cases} 0.1 & \text{if } A = \{a\} \\ 0.9 & \text{if } A = \{b, c\} \\ 0 & \text{otherwise} \end{cases},$$

and

$$m_2(A) = \begin{cases} 0.1 & \text{if } A = \{a\} \\ 0.9 & \text{if } A = \{d, e\} \\ 0 & \text{otherwise} \end{cases}.$$

Then

$$m(A) = \begin{cases} 1 & \text{if } A = \{a\} \\ 0 & \text{otherwise} \end{cases}.$$

Thus, $\gamma = 0$, though $\gamma_1 = \gamma_2 = 0.9$.

The above analysis of coordination uncertainty for belief measures reinforces previous observations that results obtained by the Dempster rule of combination in information fusion are often questionable [10, 11].

6 Conclusions

When fuzzy measures are employed in the problem of fusing information from several sources, the fusion value involves uncertainty. This uncertainty is caused by the fact that, due to the nonadditivity of fuzzy measures, the fusion value depends on the way in which the contributions from individual sources are integrated. We express this uncertainty in terms of the difference between values obtained by the upper integral and the lower integral, and derive a special and very simple expression of this uncertainty for belief measures. We show that combining two belief measures by the Dempster rule of combination may either increase or decrease this uncertainty. In our future work, we intend to examine from this point of view other combination rules for belief functions. Our preliminary investigation seems to indicate that the uncertainty always increases when the alternative combination rule proposed by Yager [10] is employed instead of the Dempster rule

References

1. Choquet, G.: Theory of capacities, Annales de l'Institut Fourier 5, (1954) 131-295.
2. Halmos, P. R.: Measure Theory, Van Nostrand, New York (1967).
3. G. Shafer: A mathematical theory of Evidence, Princeton University Press, Princeton, New Jersey (1976).

4. Sugeno, M.: Theory of Fuzzy Integral and its Applications, Ph. D. dissertation, Tokyo Institute of Technology (1974).
5. Wang, Z. and Klir, G. J.: Fuzzy Measure Theory, Plenum, New York (1992).
6. Wang, Z. and Leung, K. S.: Uncertainty carried by fuzzy measures in aggregation, Proc. IPMU 2006, 105-112.
7. Wang, Z., Leung, K. S., and Klir G. J.: Integration on finite sets, International Journal of Intelligent Systems 21 (2006), 1073-1092.
8. Wang, Z., Li, W., Lee, K. H., and Leung, K. S.: Lower integrals and upper integrals with respect to nonadditive set functions, submitted to FSS.
9. Winston, W. L.: Operations Research— Applications and Algorithms, fourth edition, Duxbury Press (2004).
10. Yager, R. R., On the Dempster-Shafer framework and new combination rules, Information Sciences 41 (1987), 93-137.
11. Zadeh, L. A., A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination, AI Magazine 7 (1986), 85-90.
12. Zong, T., Shi, P., and Wang, Z.: The Choquet integral, lower integral, and upper integral on finite sets, Proc. IPMU 2006, 2456-2463.

Two-Input Fuzzy TPE Systems*

Joao P. Carvalho, Jose Tome, and Daniel Chang-Yan

IST - TU Lisbon, INESC-ID, R. Alves Redol, 9, 1000-029 Lisboa, Portugal
{joao.carvalho, jose.tome}@inesc-id.pt, dnlcyan@gmail.com

Abstract. Single input Fuzzy TPE systems, as proposed by Sudkamp and Hammel, allow a more efficient computational inference of a single input fuzzy rule base. This paper shows that it is possible to generalize single input Fuzzy TPE systems to 2-input systems, therefore extending its range of possible applications. The paper presents the details and proof of the extension validity, and shows benchmark results comparing the 2-input Fuzzy TPE with classical inference systems.

Keywords: 2-input fuzzy TPE, fuzzy inference computational efficiency.

1 Introduction

Sudkamp and Hammell's Fuzzy TPE systems [1] provide a way to accelerate the fuzzy inference procedure in 1-input fuzzy systems. It also makes the inference process independent from the number of involved linguistic terms, since instead of a sequential application of all rules in a rule base, it allows the direct access to the relevant rules using an indexation process and infers the rule result using previously computed constants. This approach is possible due to restrictions imposed on the involved linguistic terms: all membership functions must be triangular, complementary, and evenly spaced in the universe of discourse (UoD). However, Sudkamp's approach was developed for single-input systems, and could not be directly applied in systems with more inputs. Therefore its use was limited to a few particular problems. Through the years TPE systems have been used [2][3], but when more than one input is deemed necessary, alternative inference methods that do not implement a proper fuzzy inference mechanism, like FAMs, are used instead. By proper fuzzy inference mechanisms one means mechanisms that although computationally faster, do not produce the exact same result of classical fuzzy rule base inference processes. This paper shows that it is possible to generalize single input Fuzzy TPE systems to 2-input systems extending range of possible Fuzzy TPE systems applications.

2 Fuzzy TPE Systems

Sudkamp and Hammell's model is based on an even triangular partition of a single variable fuzzy universe of discourse (UoD), hence the name TPE – Triangular

* This work was supported in part by the FCT - Portuguese Foundation for Science and Technology under project POSI/SRI/47188/2002.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 539–548, 2007.
springerlink.com

Partition Evenly. The model bases its efficiency on the symmetry of the straight lines used to define the triangular membership functions. This section summarizes Sudkamp and Hammel's fuzzy TPE systems.

In this model, the membership function (mbf) of a linguistic term A_i can be defined by (1), where $\mu_{A_i}(x)$ is the membership of crisp input x in A_i, and $\mu_{A_i}(a_i) = 1$.

$$\mu_{A_i}(x) = \begin{cases} (x - a_{i-1})/(a_i - a_{i-1}) & if\ a_{i-1} \le x \le a_i \\ (-x + a_{i+1})/(a_{i+1} + a_i) & f\ a_i \le x \le a_{i+1} \\ 0 & otherwise \end{cases} \quad . \tag{1}$$

On a n-linguistic term fuzzy variable, all mbf are equal with the exception of the lower and upper linguistic terms. The mbf of these linguistic terms are respectively truncated to the left and to the right of their a_i points, i.e., $\mu_{A_1}(-1) = 1$ and $\mu_{A_n}(1) = 1$.

An additional condition is that all mbf are complementar in all UoD, i.e.:

$$\sum_{i=1}^{n} \mu_{A_i}(x) = 1 \quad , x \in UoD\,. \tag{2}$$

Equation (2) ensures the completeness of the rule base. Fig.1 shows a seven fuzzy linguistic term TPE set.

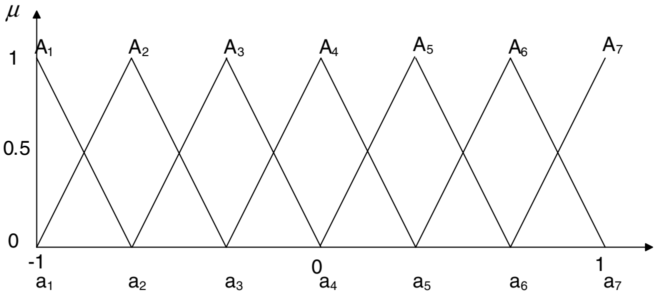


Fig. 1. Example of mbf that comply with Fuzzy TPE systems' restrictions

Within a fuzzy TPE system, one can divide the UoD into a set of $\left[a_i, a_{i+1}\right]$ intervals where $\mu_{A_i}(x) = 1 - \mu_{A_{i+1}}(x)$, $\mu_{A_i}(a_i) = 1$ and $\mu_{A_j}(a_i) = 0$ for $j \ne i$. Therefore, the membership degree within the interval can be defined by the two following straight line equations:

$$\mu_{A_i}(x) = \frac{-x + a_{i+1}}{a_{i+1} - a_i}\,. \tag{3}$$

$$\mu_{A_{i+1}}(x) = \frac{x - a_i}{a_{i+1} - a_i}. \tag{4}$$

Within a single-input/single-output TPE rule based fuzzy system, the fuzzy rules that involve antecedents A_i and A_{i+1} can be expressed as:

$$\begin{array}{l} \textit{if } X \textit{ is } A_i \textit{ then } Z \textit{ is } C_r \\ \textit{if } X \textit{ is } A_{i+1} \textit{ then } Z \textit{ is } C_s \end{array},$$

where C_r and C_s are also TPE linguistic terms. Defining c_r and c_s as the central points of C_r and C_s, and by using weighted averaging defuzzification, the result z specified by crisp input x is given by:

$$z = \frac{\mu_{A_i}(x)c_r + \mu_{A_{i+1}}(x)c_s}{\mu_{A_i}(x) + \mu_{A_{i+1}}(x)}. \tag{5}$$

By replacing (3) and (4) in (5) and simplifying one obtains:

$$z = \frac{x(c_s - c_r) + a_{i+1}c_r - a_i c_s}{a_{i+1} - a_i} = x\frac{c_s - c_r}{a_{i+1} - a_i} + \frac{a_{i+1}c_r - a_i c_s}{a_{i+1} - a_i}. \tag{6}$$

Equation (6) shows that only two constants are needed to compute z given any crisp input $x \in [a_i, a_{i+1}]$. These constants, $(c_s - c_r)/(a_{i+1} - a_i)$ and $(a_{i+1}c_s - a_i c_r)/(a_{i+1} - a_i)$, are determined by the fuzzy rule base and the midpoints of the triangular membership functions, and therefore independent of x. Thus a fuzzy TPE system can be represented by a table containing the constants associated with each input interval. Since all intervals are equal in size, one can directly address the constants associated with a given input x by using the function

$$trunc\left(\frac{2x + 2}{n - 1}\right) + 1. \tag{7}$$

On a fuzzy TPE system one can skip the inference of all rules in the rule base. Given input x, all that is needed is to get the appropriate constants from a table and apply (6) to find defuzzified output z. Therefore, inference time on a TPE fuzzy rule base is independent of the number of rules and linguistic term size, and TPE fuzzy inference is particularly well adapted to deal with large fuzzy rule bases.

3 Two-Input Fuzzy TPE Systems

A single input variable fuzzy rule based system has an obviously limited interest. Multiple input Fuzzy TPE systems are necessary if one wants to use these mechanisms in a significant range of applications. Although it is theoretically possible to extend the model to a higher number of inputs, in this paper we limit the extension to 2-input systems. One must note that in rule based fuzzy systems it is sometimes possible to organize a n-input rule base into several connected 2-input rule bases. This manipulation is particular to the system being modeled and relies on the interdependence of its input variables. Therefore, let us focus on the 2-input case.

Any 2-input fuzzy system rule base with n-linguistic term variables can be represented as a 2 dimensional table. Table 1, represents an excerpt of a generic 2-input fuzzy rule base, where all A_i, B_j and C_x are fuzzy linguistic terms defined by a fuzzy membership function. On such a system, any combination of input values activates at most four different rules of the rulebase.

Table 1. An excerpt of a 2-input Fuzzy Rule Base

X Y	A_i	A_{i+1}
B_j	C_r	C_t
B_{j+1}	C_s	C_u

if X is A_i *and Y is* B_j *then Z is* C_r

if X is A_i *and Y is* B_{j+1} *then Z is* C_s

if X is A_{i+1} *and Y is* B_j *then Z is* C_t

if X is A_{i+1} *and Y is* B_{j+1} *then Z is* C_u

Since TPE's membership functions are, by definition, triangular and symmetric, Table 1 can be replaced by a Numeric Inference (NI) table, which is an alternative representation where each linguistic term is replaced by its central point x-coordinate. Therefore, if one assumes the weighted averaging defuzzification method, one can deduce the following output equation for the rule base presented in Table 1, where c_r, c_s, c_t and c_u are the central point x-coordinates of the consequent mbfs:

$$z=\frac{\min(\mu_{A_i}(x),\mu_{B_j}(y))c_r+\min(\mu_{A_i}(x),\mu_{B_{j+1}}(y))c_s+\min(\mu_{A_{i+1}}(x),\mu_{B_j}(y))c_t+\min(\mu_{A_{i+1}}(x),\mu_{B_{j+1}}(y))c_u}{\min(\mu_{A_i}(x),\mu_{B_j}(y))+\min(\mu_{A_i}(x),\mu_{B_{j+1}}(y))+\min(\mu_{A_{i+1}}(x),\mu_{B_j}(y))+\min(\mu_{A_{i+1}}(x),\mu_{B_{j+1}}(y))} \quad (8)$$

In order to find the 2-dimensional equivalent of (6), consider two fuzzy variables A and B defined respectively by n and m TPE mbf. Fig. 2 shows a pictorial representation of A and B, where x and y are the crisp input values.

The 2-dimensional intervals we consider to find the 2-dimensional equation equivalent of (6) are represented by $\left[a_i, a_{i+1}\right]$ and $\left[b_j, b_{j+1}\right]$. The shadowed region in Fig. 2 shows a 2-dimensional interval. 2-dimensional interval indexation is based on (7), and results on (9):

$$
\begin{aligned}
interval_A &= \begin{cases} n-1 & ,if \;\; x=1 \\ trunc\left(\dfrac{(x+1)(n-1)}{2}\right)+1 & ,if \;\; x\neq 1 \end{cases} \\
interval_B &= \begin{cases} m-1 & ,if \;\; y=1 \\ trunc\left(\dfrac{(y+1)(m-1)}{2}\right)+1 & ,if \;\; y\neq 1 \end{cases}
\end{aligned}
\qquad (9)
$$

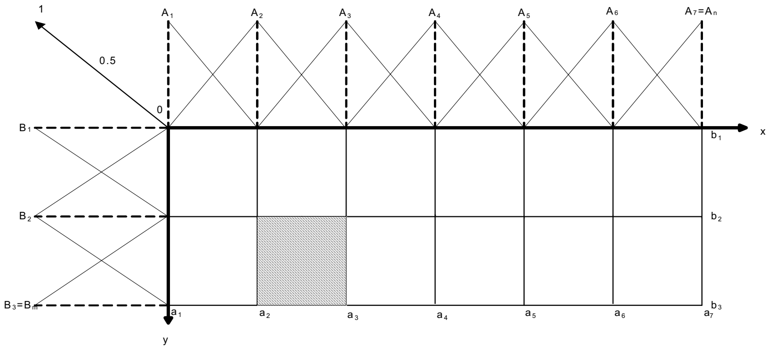


Fig. 2. Two-dimensional representation of TPE linguistic terms. The shadowed region represents a 2-dimensional interval.

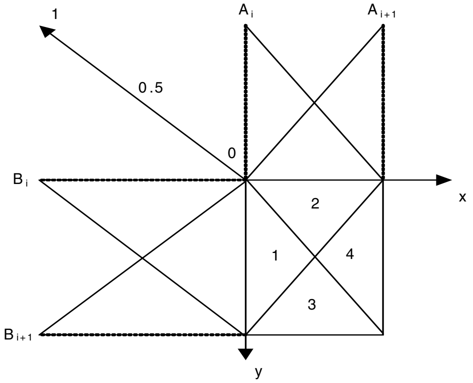


Fig. 3. The 4 different areas in each interval

One can divide each 2-dimensional interval in four different areas, as depicted in Fig.3. Each of these areas is characterized by a constant relation between the membership degrees of *x* and *y* in the pertaining linguistic terms. For example, in Area 1, all the following equations hold true:

$$\min(\mu_{A_i}(x), \mu_{B_j}(y)) = \mu_{B_j}(x),$$
$$\min(\mu_{A_i}(x), \mu_{B_{j+1}}(y)) = \mu_{B_{j+1}}(x),$$
$$\min(\mu_{A_{i+1}}(x), \mu_{B_j}(y)) = \mu_{A_{i+1}}(x),$$
$$\min(\mu_{A_{i+1}}(x), \mu_{B_{j+1}}(y)) = \mu_{A_{i+1}}(x).$$

Based on the characteristics of each of the 4 areas, it is possible to divide the defuzzification equation (8) into 4 different defuzzification equations, where the min() function is not necessary.

The 4 areas can be divided by two straight lines defined by the following equations:

$$y = \frac{(n-1)(x+1)}{m-1} + 2\times\left(\frac{interval_B - 1}{m-1} - \frac{interval_A - 1}{n-1}\right) - 1. \qquad (10)$$

$$y = \frac{2-(n-1)(x+1)}{(m-1)} + 2\times\left(\frac{interval_B - 1}{m-1} + \frac{interval_A - 1}{n-1}\right) - 1. \qquad (11)$$

Equations (10) and (11) can be used to create Table2, which divides the Universe of Discourse in intervals, and each interval in the 4 different areas.

Table 2. Area definition for each interval

	$y < \frac{(n-1)(x+1)}{(m-1)} + 2\times\left(\frac{interval_B - 1}{m-1} - \frac{interval_A - 1}{n-1}\right) - 1$	$y \geq \frac{(n-1)(x+1)}{(m-1)} + 2\times\left(\frac{interval_B - 1}{m-1} - \frac{interval_A - 1}{n-1}\right) - 1$
$y \geq \frac{2-(n-1)(x+1)}{(m-1)} + 2\times\left(\frac{interval_B - 1}{m-1} + \frac{interval_A - 1}{n-1}\right) - 1$	Area 1	Area 3
$y < \frac{2-(n-1)(x+1)}{(m-1)} + 2\times\left(\frac{interval_B - 1}{m-1} + \frac{interval_A - 1}{n-1}\right) - 1$	Area 2	Area 4

Since we are dealing with a TPE system, (3) and (4) are still valid for each fuzzy variable. Therefore, $\mu_{A_i}(x), \mu_{A_{i+1}}(x), \mu_{B_j}(y), \mu_{B_{j+1}}(y)$ can be defined by:

$$\begin{array}{ll} \mu_{A_i}(x) = \dfrac{-x + a_{i+1}}{a_{i+1} - a_i} & \mu_{B_j}(y) = \dfrac{-y + b_{j+1}}{b_{j+1} - b_j} \\ \mu_{A_{i+1}}(x) = \dfrac{x - a_i}{a_{i+1} - a_i} & \mu_{B_{j+1}}(y) = \dfrac{y - b_j}{b_{j+1} - b_j} \end{array}. \qquad (12)$$

Considering the 4 different areas defined in Fig.3 for each interval, one can replace (12) in (8) and manipulate the resulting expressions in order obtain the following defuzzification equations:

Area 1:

$$z_1 = \frac{\frac{-y+b_{j+1}}{b_{j+1}-b_j}c_r + \frac{y-b_j}{b_{j+1}-b_j}c_s + \frac{x-a_i}{a_{i+1}-a_i}c_t + \frac{x-a_i}{a_{i+1}-a_i}c_u}{\frac{-y+b_{j+1}}{b_{j+1}-b_j} + \frac{y-b_j}{b_{j+1}-b_j} + \frac{x-a_i}{a_{i+1}-a_i} + \frac{x-a_i}{a_{i+1}-a_i}} = \frac{y\frac{(c_s-c_r)}{b_{j+1}-b_j} + \frac{b_{j+1}c_r - b_j c_s}{b_{j+1}-b_j} + x\frac{(c_t+c_u)}{a_{i+1}-a_i} + \frac{-a_i(c_t+c_u)}{a_{i+1}-a_i}}{y.0 \quad + \quad 1 \quad + x\frac{2}{a_{i+1}-a_i} + \frac{-2a_i}{a_{i+1}-a_i}} \tag{13}$$

Area 2:

$$z_2 = \frac{\frac{-x+a_{i+1}}{a_{i+1}-a_i}c_r + \frac{y-b_j}{b_{j+1}-b_j}c_s + \frac{x-a_i}{a_{i+1}-a_i}c_t + \frac{y-b_j}{b_{j+1}-b_j}c_u}{\frac{-x+a_{i+1}}{a_{i+1}-a_i} + \frac{y-b_j}{b_{j+1}-b_j} + \frac{x-a_i}{a_{i+1}-a_i} + \frac{y-b_j}{b_{j+1}-b_j}} = \frac{y\frac{(c_s+c_u)}{b_{j+1}-b_j} + \frac{-b_j(c_s+c_u)}{b_{j+1}-b_j} + x\frac{(c_t-c_r)}{a_{i+1}-a_i} + \frac{a_{i+1}c_r - a_i c_t}{a_{i+1}-a_i}}{y\frac{2}{b_{j+1}-b_j} + \frac{-2b_j}{b_{j+1}-b_j} + \quad x.0 \quad + \quad 1} \tag{14}$$

Area 3:

$$z_3 = \frac{\frac{-y+b_{j+1}}{b_{j+1}-b_j}c_r + \frac{-x+a_{i+1}}{a_{i+1}-a_i}c_s + \frac{-y+b_{j+1}}{b_{j+1}-b_j}c_t + \frac{x-a_i}{a_{i+1}-a_i}c_u}{\frac{-y+b_{j+1}}{b_{j+1}-b_j} + \frac{-x+a_{i+1}}{a_{i+1}-a_i} + \frac{-y+b_{j+1}}{b_{j+1}-b_j} + \frac{x-a_i}{a_{i+1}-a_i}} = \frac{y\frac{-(c_r+c_t)}{b_{j+1}-b_j} + \frac{b_{j+1}(c_r+c_t)}{b_{j+1}-b_j} + x\frac{(c_u-c_s)}{a_{i+1}-a_i} + \frac{a_{i+1}c_s - a_i c_u}{a_{i+1}-a_i}}{y\frac{-2}{b_{j+1}-b_j} + \frac{2b_{j+1}}{b_{j+1}-b_j} + \quad x.0 \quad + \quad 1} \tag{15}$$

Area 4:

$$z_4 = \frac{\frac{-x+a_{i+1}}{a_{i+1}-a_i}c_r + \frac{-x+a_{i+1}}{a_{i+1}-a_i}c_s + \frac{-y+b_{j+1}}{b_{j+1}-b_j}c_t + \frac{y-b_j}{b_{j+1}-b_j}c_u}{\frac{-x+a_{i+1}}{a_{i+1}-a_i} + \frac{-x+a_{i+1}}{a_{i+1}-a_i} + \frac{-y+b_{j+1}}{b_{j+1}-b_j} + \frac{y-b_j}{b_{j+1}-b_j}} = \frac{y\frac{(c_u-c_t)}{b_{j+1}-b_j} + \frac{b_{j+1}c_t - b_j c_u}{b_{j+1}-b_j} + x\frac{-(c_r+c_s)}{a_{i+1}-a_i} + \frac{a_{i+1}(c_r+c_s)}{a_{i+1}-a_i}}{y.0 \quad + \quad 1 \quad + x\frac{-2}{a_{i+1}-a_i} + \frac{2a_{i+1}}{a_{i+1}-a_i}} \tag{16}$$

Note that all 4 equations can be expressed under the form

$$z = \frac{yH_1 + H_2 + xH_3 + H_4}{yH_5 + H_6 + xH_7 + H_8}, \tag{17}$$

where all H_{i} are constants. As in the single input case, the H_{i} constants are determined by the fuzzy rule base and by the central point of each linguistic term (a_i, b_j, c_l), and are independent of input crisp values x and y. Therefore, they can be computed only once, prior to the actual rule base inference process. Based on (9) and Table 2, one can directly index an inference matrix with dimensions [n-1]×[m-1]×[4]×[8], where in each position of the inference matrix one will have the previously defined H constant. From this inference matrix, one can directly compute the defuzzified output z. By using this inference process, one avoids the sequential computation of each single rule while simultaneously simplifying the defuzzification process. Therefore, as long as all H_{i} are previously computed, and with proper indexation of the input values, it is possible to obtain a very fast fuzzy inference process that does not grows exponentially with the increase of number of linguistic terms in each involved fuzzy variable.

4 Results

In order to test the computational efficiency of 2-Input Fuzzy TPE Systems, a comparison was made with traditional fuzzy rule based inference methods. Since the

Fuzzy TPE method computational inference time does not grow exponentially with the increase of number of linguistic terms in each involved fuzzy variable (contrary to the traditional methods), better results were to be expected when a larger number of membership functions were used. Therefore tests were made involving fuzzy variables with 3 and 7 linguistic terms, resulting in complete rule bases with 9 and 49 rules respectively. Table 3. and Table 4. show the used rule bases. Note that inference computational time is independent of actual rule content.

Table 3. Fuzzy rule base for 3 linguistic term's input variables

X *Y*	*L*	*M*	*H*
L	*L*	*M*	*H*
M	*M*	*H*	*M*
H	*H*	*M*	*L*

For the traditional fuzzy inference method computational implementation, one opted to use an individual array to represent each linguistic term membership function. This method is computationally faster than using straight line equations, since inferring $\mu_{A_i}(x)$ is done via a simple array indexation.

Table 4. Fuzzy rule base for 7 linguistic term's input variables

X *Y*	*VVL*	*VL*	*L*	*M*	*H*	*VH*	*VVH*
VVL	*VVL*	*VL*	*L*	*M*	*H*	*VH*	*VVH*
VL	*VL*	*L*	*M*	*H*	*VH*	*VVH*	*VH*
L	*L*	*M*	*H*	*VH*	*VVH*	*VH*	*H*
M	*M*	*H*	*VH*	*VVH*	*VH*	*H*	*M*
H	*H*	*VH*	*VVH*	*VH*	*H*	*M*	*L*
VH	*VH*	*VVH*	*VH*	*H*	*M*	*L*	*VL*
VVH	*VVH*	*VH*	*H*	*M*	*L*	*VL*	*VVL*

Using this method, the C code to compute a fuzzy rule like "If *x* is *A* and *y* is *B* Then *Z* is *C*", can be resumed as:

```
miu=MIN(is(A,x),is(B,y));  //If x is A And y is B
if (miu!=0){
    for (n=0; n<number_of_mbf; n++)   //Then z is C
         foutput[n] += miu*C[n];
}
```

where function ‘is()’ is simply:

```
float is(int LT, int inp){ //LT-Linguistic Term,inp-input
    return mbf[LT][inp];} //mbf-Array of all LT
```

The above mentioned code is repeated for each rule in the rule base. As a result one obtains an array representing fuzzy variable *z*.

The defuzzification method used is weighted averaging, which is also simple to implement:

```
for  (n=0;n<mbfsize;n++){
     mass+=foutput[n]*n;
     area+=foutput[n];
}
z=mass/area;
```

Although the defuzzification code is simple, it is not computationally efficient since the number of rules grows exponentially with the increase in the number of linguistic terms, and the code uses several cycles to run through the rule base. It is possible to vastly improve defuzzification computation time, but not without imposing restrictions to mbf shape (which is what fuzzy TPE does).

The Fuzzy TPE method accesses the active rules through direct indexation, and therefore avoids the necessity to test each single rule. Due to the restrictions in the shape of the membership functions, it also uses a much more efficient defuzzification method.

Table 5. shows the average inference computing time for the rule bases in Table 3. and Table 4.

Table 5. Average rule base inference computing time comparison (ns)

Method #LT/#rules	Classical Inference	Fuzzy TPE
3 / 9	44.4	6.6
7 / 49	58.7	7.2

The average inference time was obtained after computing 500000 inputs on a 1.86GHz PentiumM processor. One can see that Fuzzy-TPE inference is roughly 8 times faster than a classical fuzzy inference method, which can be considered a considerable improvement. However, one must not ignore that the involved computing times are so small that can often be considered a negligible factor in many applications. Therefore, due to the restrictions imposed to the linguistic terms, the use of 2-input Fuzzy-TPE systems can only be justified in applications where a large volume of information must be processed and where time is a major issue, like real-time control systems or computer chess.

5 Conclusions and Future Developments

It was shown that two-input Fuzzy TPE inference systems can be implemented and that they provide a significant improvement over classical fuzzy inference systems in what concerns computing performance. However, this improvement can only be useful in systems where time and performance are critical and where the restrictions imposed to the linguistic term's membership functions can be accepted.

N-input Fuzzy TPE systems are theoretically possible. However, as the number of inputs increases, the complexity of the defuzzification equation and the dimensionality of the computing matrixes increase exponentially. Therefore, n-input Fuzzy TPE systems viability over classical inference systems still needs to be proved on a future work.

References

1. Sudkamp,T., Hammell,R.J., "Interpolation, Completion, and Learning Fuzzy Rules", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, 2, 1994
2. Camacho, E., Berengel, M., Rubio, F., "Advanced Control of Solar Plants", Advances in Industrial Control, Springer-Verlag, 1997
3. Cross,V., Sudkamp,T., "Sparse Data and Rule Base Completion", Proceedings of the 2003 Conference of the North American Fuzzy Information Society, NAFIPS 2003, Chicago, 2003

Intelligent Decision Support System

Janos L. Grantner[1], George A. Fodor[2], Ramakrishna Gottipati[3], Norali Pernalete[4], and Sandra Edwards[1]

[1] Western Michigan University, Kalamazoo, MI, USA
janos.grantner@wmich.edu, sandra.edwards@wmich.edu
[2] ABB Process Automation, 721 59 Vasteras, Sweden
george.a.fodor@se.abb.com
[3] Dynamic Enterprise Solution, Inc., Libertyville, IL, USA
gottipati@dynamic-enterprise.net
[4] Cal Poly Pomona, Pomona, CA, USA
npernalete@csupomona.edu

Abstract. There is a need to develop an automated assessment and training procedure for children with eye-hand coordination problem. Such system is expected to reduce the burden and the associated cost of having a trained professional present at any assessment, or training session for each child. The intelligent decision support system is based upon a fuzzy automaton. By using qualitative (fuzzified) data from the previous test the system will make a decision on the complexity of the next test to be performed. A set of assessment tests, commonly used by occupational therapists, were chosen to implement the various functions using force, inertia and viscosity effects. A test bed has been designed for these tasks that consists of a six-degree-of-freedom force-reflecting haptic interface device called PHANToM along with the GHOST SDK Software, and the Intelligent Decision Support System software.

1 Introduction

Learning to write is an important occupation of children [5]. Problems with handwriting or drawing skills (graphomotor skills) are frequently the reason children in public schools are referred to occupational therapy services [7], [8]. Children with genetic anomalies or neurological disorders can also have problems with eye-hand coordination such as children with Down syndrome, cerebral palsy, and learning disabilities. All these diagnoses require intervention for eye-hand coordination and often for grip strength because these components are necessary for successfully performing activities of daily living such as dressing, feeding, drawing, and writing.

Assessing the eye-hand coordination skills of children with disabilities and making decisions on the next, more complex test by using crisp, quantitative terms may not be an optimum approach. Fuzzy logic [9] allows the aggregation of measured data and expert knowledge that is expressed in qualitative terms in a common mathematical model. A fuzzy automaton [10] can help in developing an intelligent decision system by recommending the sequence and complexity of the next test to be given. This recommendation will be based upon the results of the previously performed test and expert knowledge. The decision mechanism can be fine tuned as more test results become available.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 549–557, 2007.
springerlink.com

This paper is organized as follows: Section 2 provides the background of previous applications of robotics for eye-hand coordination assessment and treatment. Section 3 covers a brief description of the standard research design for dexterity assessment tasks and the development of the assistance functions for haptic execution of these tasks. Section 4 describes the Intelligent Decision Support System. Preliminary results and conclusions are presented in Sections 5 and 6, respectively.

2 Background

Extensive research is being done in the field of haptics to improve hand and arm movements. The concepts of motor control can be combined with robotics for a compelling application in rehabilitation and these robotics devices can be used for guiding patients along a desired trajectory [1]. One of the approaches taken was to expose a subject to a perturbing field to develop an internal model of the field as a relation between experienced limb states and forces [2]. In this study, the authors concluded that after-effects persisted after many trials using force field. Another study showed that haptic technologies can give rise to new generation of virtual environment applications for teaching motor skills and manual crafts [3]. The approach taken in that study is based upon a principle of rehabilitation in which simple movements can be improved by constant practice. Previous work by Bardorfer et all [4] shows labyrinths or mazes created in the virtual environment, in which the user has to move the pointer (ball) through it and could feel the reactive forces of the walls. It has been reported that eye-hand coordination in children with limited motor skills can be improved by using a Robotic Haptic Device [6].

3 Design of Experiments

The research approach is based upon the comparison of the performance of an experimental and a control group. The experimental group is exposed to standard occupational therapy tests as well as tests performed on the haptic robot. The control group works only on the standard tests. Initally, both groups are exposed to Standardized Occupational Therapy Assessment Tasks for one week. Then the experimental group works with the Haptic Device for eight weeks. During that timeframe the control group has no activities. Finally, both groups carry out again the Standardized Occupational Therapy Assessment Tasks for one week.

The initial (AInitial) and final (AFinal) data collection sessions are used in the assessment of the baseline performance and the occupational performance changes of the subjects. The Minnesota Handwriting Assessment Test has been selected to accomplish that because of its ability to measure the quality of handwriting (rate, legibility, space, line size and form). The child is required to copy words on the sample (“the quick brown fox jumped over the lazy dogs”) and they represent the entire alphabet. The letters are in mixed order. The subject’s handwriting sample is scored in each category. The score in each category (except for the rate) is based upon the error rate using a maximum score of 34. The rate is graded by counting the number of letters completed in 2.5 minutes.

The experimental group is assigned tasks using the Haptic Device. These sessions provides carefully selected tasks that make use of the Robotic Haptic Device for eye-hand coordination intervention. The hand grasp and function are visually identified and noted during every intervention by the therapist who closely monitors for any changes.

Each child will have two sessions of evaluation, $A_{Initial}$ and A_{Final} for 30 minutes each session. The child in the experimental group will receive intervention by means of the haptic robot once a week for 8 weeks, 20 minutes each. The child in the control group will not receive the intervention but will complete the assessments at the same time interval as the intervention group. Various labyrinth trajectories have been developed and tested, one of them is shown in Figure 1. The subject is required to move the ball along the specified trajectory. In addition to testing the free movement of the subject, various effects in form of assistant functions have been also implemented such as: velocity assistance, plane constraints, inertia effect, viscosity effect and force feedback effect.

The inertia effect is implemented by setting the damping co-efficient, mass, gravity and spring stiffness where the viscosity effect is implemented by setting a force opposite to the velocity of the PHANToM. The main idea of these effects is to increase the grip and hand strength in children during the task execution. The PHANToM robot used for the research is shown in Figure 2.

4 Intelligent Decision Support System

The functional block diagram of the Intelligent Decision Support System (IDSS) is given in Figure 3. In the present phase of the research only children of 5-year old have participated.

The system is made up of two major sections. The Main Controller section includes a computer workstation (PC), the PHANToM robot and the user interface software.

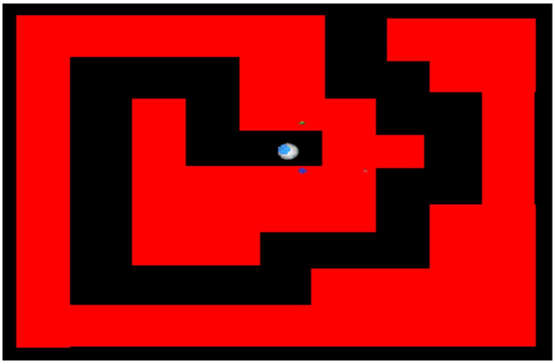


Fig. 1. Labyrinth Task User Interface

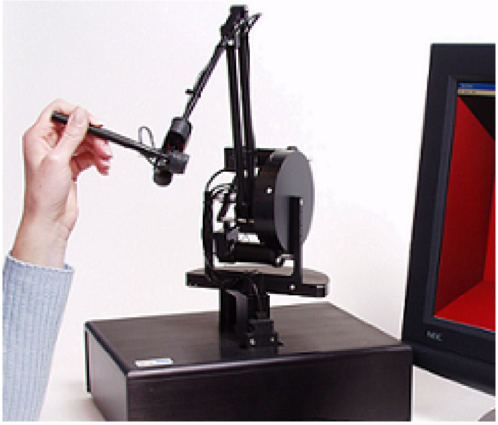


Fig. 2. PHANToM Haptic Robot

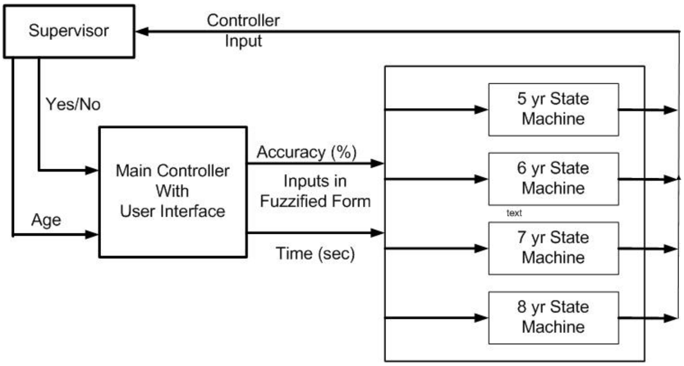


Fig. 3. Block Diagram of the IDSS

The other key section is a simulated fuzzy automaton that is configured to accommodate the testing of the 5-year old children group. The task-level flowchart of the Intelligent Decision Support System is depicted in Figure 4.

The general model of the fuzzy automaton being used in the IDSS is described in [10]. In this research, a simplified version is employed. Inference operations (e.g., to classify the eye-hand coordination skills of the children in a qualitative scale) are omitted. The state transition features of the automaton are used to devise the sequence and the complexity of the tests to be performed.

The controller unit consists of the Graphical User Interface (GUI), which allows the therapist to enter the age of the subject and grants the permission to execute the tests suggested by the Intelligent Decision Support System. The main control unit takes the age and assigns the first task to be performed by the subject. It then calculates the accuracy (with respect to the number of times the subject is out of the desired path) and the time (in seconds) taken to carry on the task, then fuzzifies the values and provides them as inputs to the IDSS. The IDSS then makes use of the fuzzy automaton model to recommend the next test to be performed by the child. Based upon its present state and the fuzzified results of the test just performed the fuzzy automaton makes a transition to a new state. In this new state when it becomes the present state only one two-valued output is set, out of a 12-element output vector. This output is then used to choose the next task to be performed by the subject. Depending on his/her performance the subject may get stuck at some particular task level, or may move up to a more sophisticated level. If a subject performs really well the sequence of states will be as follows: 1, 4, 7, 11, and Stop. The level of difficulty in a task increases as the state number increases. However, the approval by the therapist is required to proceed with any new test. The therapist can also abort the test at any point in time. The state transition graph of the fuzzy automaton for the 5-year old children group is shown in Figure 5.

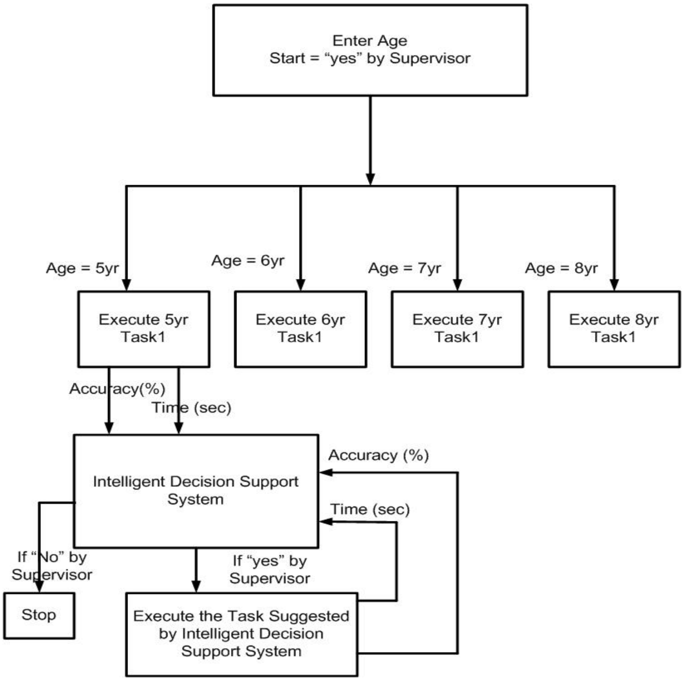


Fig. 4. Task Level Flowchart of the IDSS

The states are designated as follows: (1): initial state and no effects, or assistance, (2) and (3): design with maximum force assistance and minimum force assistance, respectively, (4): design with curves rather than with sharp edges in the path but no effects, or assistance, (5) and (6): design with minimum inertia effect, and medium inertia effect, respectively. The labyrinth for these states is illustrated in Figure 1. In State (7) a more difficult labyrinth is employed with no effects, or assistance. State (8) is a design with force assistance. States (9-11) are designs with minimum, medium and maximum inertia effects, respectively. State Stop is the last possible state in the

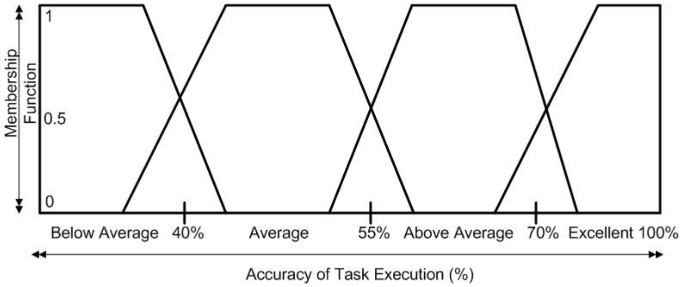


Fig. 5. membership Functions for Accuracy

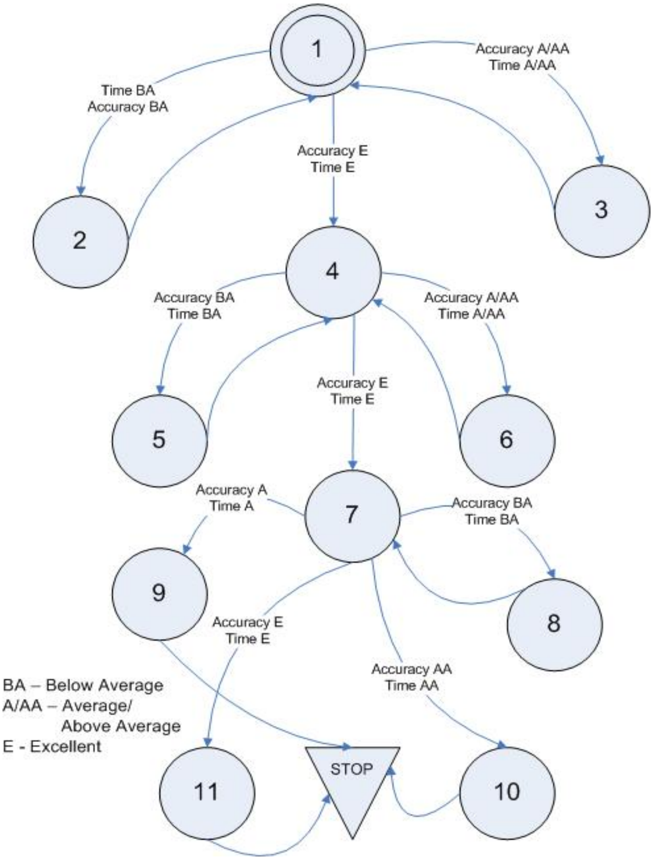


Fig. 6. State Transition Graph of the Fuzzy Automaton

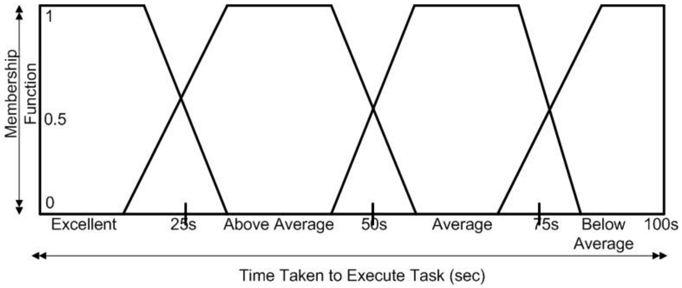


Fig. 7. Membership Functions for Time Taken

sequence of tests. The number of states visited depends upon the performance of the subject. This state transition graph has been developed using the help of an occupational therapy expert. The state transition conditions are subject to further experiments and refinements. The input membership functions used in evaluating the state transition conditions are given in Figures 6 and 7.

5 Simulation Results

For initial verification of the system a test bed has been developed using Matlab. Random inputs have been provided to the system and the actual state transitions have been verified against the state transition graph developed. The simulation results are shown in Figures 8-10. The raw input data (before fuzzification) for Accuracy and Time taken to perform the designated task are given in Figures 8 and 9, respectively. Figure 10 illustrates that all states have been reached during the trials.

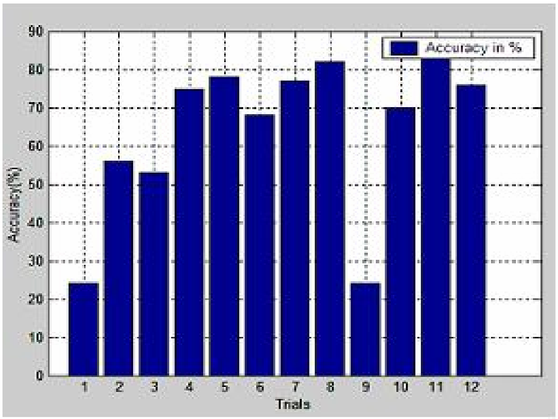


Fig. 8. Raw Input Data for Accuracy

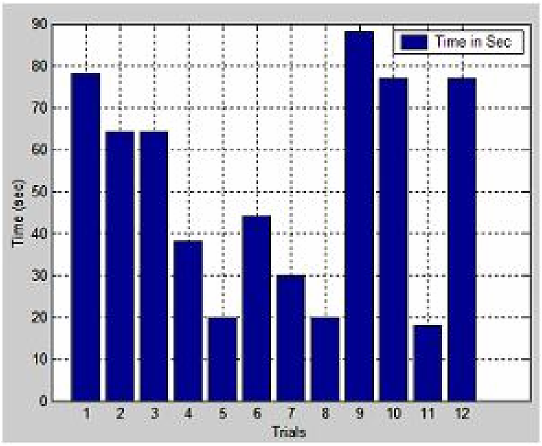


Fig. 9. Raw Input Data for Time Taken

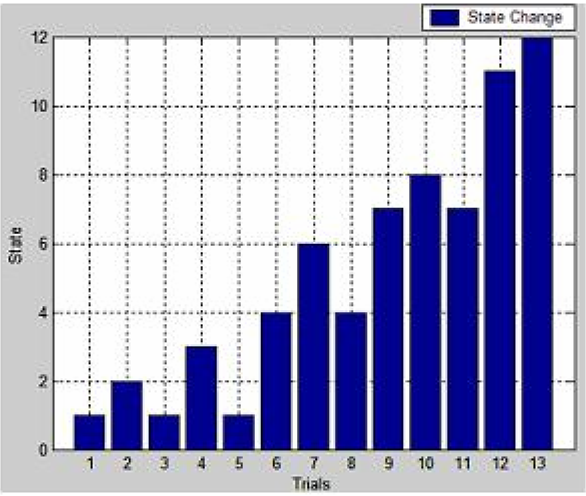


Fig. 10. States Visited During the Trials

6 Conclusions and Future Work

The initial results of a research to develop an Intelligent Decision Support System are presented. In the next phase the IDSS will be loaded as a reconfigurable agent into the Broker Architecture program of the Intelligent Fuzzy Controllers Laboratory [11], [12] at Western Michigan University. The Broker Architecture program and the IDSS system will be integrated with the PHANToM using the Ghost software and will be tested on children with eye-hand coordination problems. In the future the IDSS system will be extended to classify the skill-levels of the pool of subjects. It will be done by adding the fuzzy inference capability to the automaton. The If Then rules of the knowledge base will be constructed by interviewing experts of occupational therapy and also by extracting knowledge from the test data obtained from a larger pool of children.

The final objective for the IDSS system is to develop a distributed system using the Internet. There will be no need for a therapist be present at each location, e.g., in classrooms in a wide geographical area. Instead, only one therapist in a central location will monitor the tests performed at various sites and will make decisions with respect to the test procedures.

References

[1] H. Kerbs, N. Hogan, M. Aisen and B. Volpe, "Robot-aided neurorehabilitation," *IEEE Transactions on Rehabilitation Engineering*, Vol. 6, pp. 75-87, 1998.

[2] F. Mussa-Ivaldi A., J. Patton L., "Robots can teach people how to move their arm," International Conference on Robotics and Automation. pp. 300-305, 2000.

[3] G. Prisco M., C. Avizzano A., M. Calcara, S. Ciancio, S. Pinna and M. Bergamasco, "A virtual environment with haptic feedback for the treatment of motor dexterity disabilities," International Conference on Robotics and Automation. pp. 3721-3726, 1998.

[4] Aleš Bardorfer, Marko Munih, Anton Zupan, Alenka Primožič, "Upper Limb Motion Analysis Using Haptic Interface," IEEE/ASME Transactions on Mechatronics, VOL.6, NO.3, pp3721-3726 September 2001.

[5] Amundson, S.J. (1995). *Evaluation Tool of Children's Handwriting*. Homer, AK: OT KIDS.

[6] N Pernalete, R Gottipati, S Mikkilineni, S Edwards, E McCann, RV Dubey,W Yu, "Eye-Hand Coordination Assessment using a Robotics Haptic Device," International Conference on Robotics and Automation. 2004

[7] Edwards, S., Buckland, D. & McCoy-Powlen, J. (2002). *Developmental and Functional Hand Grasps*, New Jersey, Slack.

[8] Diekema, S. M., Deitz, J., & Amundson, S. J. (1998). Evaluation Tool of Children's Handwriting-Manuscript. *American Journal of Occupational Therapy,* 52, 248-254

[9] L. A. Zadeh, Fuzzy Sets, *Information and Control*, Vol. 8, 1965.

[10] J. L. Grantner, G. A. Fodor, "Fuzzy Automaton for Intelligent Hybrid Control Systems", 2002 World Congress on Computational Intelligence, WCCI'02/FUZZ-IEEE'02, May 12-17, 2002, Hilton Hawaiian Village Hotel, Honolulu, HI, in the CD Proceedings, ISBN: 0-7803-7281-6

[11] J. L. Grantner, G. A. Fodor, R. Gottipati, "Fuzzy Logic Enabled Software Agents for Supervisory Control", in the Proceedings of the FUZZ-IEEE'04 Conference, July 25-29, 2004, Budapest, Hungary.

[12] J. L. Grantner, "Intelligent Control Laboratory for Complex Hybrid System", DURIP'2001 Award (DAAD19-01-1-0431)

An Adaptive Location Service on the Basis of Fuzzy Logic for MANETs

Ihn-Han Bae and Yoon-Jeong Kim

School of Computer and Information Communication Eng., Catholic
University of Daegu, GyeongSan 712-702, Korea
ihbae@cu.ac.kr, kimyj2845@naver.com

Abstract. Location services are used in mobile ad hoc and hybrid networks either to locate the geographic position of a given node in the network or for locating a data item. One of the main usages of position location services is in location based routing algorithms. In particular, geographic routing protocols can route messages more efficiently to their destinations based on the destination node's geographic position, which is provided by a location service. In this paper, we propose an adaptive location service on the basis of fuzzy logic called FHLS (*Fuzzy Hierarchical Location Service*) for mobile ad hoc networks. The FHLS uses the adaptive location update scheme using the fuzzy logic on the basis of the mobility and the call preference of mobile nodes. The performance of the FHLS is to be evaluated by using a simulation, and compared with that of existing HLS scheme.

Keywords: Mobile Ad-hoc Network, Location Service, Fuzzy Logic.

1 Introduction

Mobile ad hoc networks (MANETs) enable wireless communication between mobile hosts without relying on a fixed infrastructure. In these networks the mobile hosts themselves forward data from a sender to a receiver, acting both as router and end-system at the same time. MANETs have a wide range of applications, e,g., range extension of WLAN access points, data transmission in disaster areas and inter-vehicular communication. Due to scare bandwidth, varying network connectivity and frequent topology changes caused by node mobility and transient availability, routing algorithms tailored for wired networks will not operate well if directly transposed to MANETs.

Since no fixed infrastructure of servers is assumed in MANETs, it is useful to devise a scheme through which various services offered within the network may be located. With the availability of such location services, it is tempting to adapt and exploit them for storing routing information. By storing the geographic location of mobile nodes in designed location servers in the network, it is possible to introduce a new family of routing algorithms that may potentially perform better than the traditional approach of discovering and maintaining end-to-end routes.

In this paper, we present an adaptive location service on the basis of fuzzy logic called FHLS (*Fuzzy Hierarchical Location Service*) for MANETs. FHLS divides the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 558–565, 2007.
springerlink.com

area covered by the network into a hierarchical of regions. The top-level region covers the complete network. A region is subdivided into several regions of the next lower level until the lowest level is reached. We call a lowest level region a cell. Using the hierarchy as a basis, the FHLS uses the adaptive location update scheme using the fuzzy logic on the basis of the mobility and the call preference of mobile nodes. The performance of the FHLS is to be evaluated by a simulation, and compared with that of existing HLS scheme.

The remainder of this paper is organized as follows. The next section provides an overview of related work. In Section 3, we present the FHLS algorithm, which serves as a basis for a routing algorithm. In Section 4, we undertake a simulation study for the FHLS. We finally conclude and describe future work in Section 5.

2 Related Work

In routing protocol of MANETs, the location service uses the location information of a node for packet routing. So, many researches for location management in MANETs were performed recently. A location service that uses flooding to spread position information is DREAM (*Distance Routing Effect Algorithm for Mobility*) [1]. With DREAM, each node floods its position information in the network with varying flooding range and frequency. The frequency of the flooding is decreased with increasing range. Thus, each node knows the location of each other node, whereas the accuracy of this information depends on the distance to the node.

The GLS (*Grid Location Service*) [2] divides the area containing the ad hoc network into a hierarchy of square forming a quad-tree. Each node selects one node in each element of the quad-tree as a location server. Therefore the density of location servers for a node is high in areas close to the node and becomes exponentially less dense as the distance to the node increases. The update and request mechanisms of GLS require that a chain of nodes based on node IDs is found and traversed to reach an actual location server for a given node. The chain leads from the updating or requesting node via some arbitrary and some dedicated nodes to a location server.

DLM (*Distributed Location Management Scheme*) [3] partitions the mobile node deployment region into a hierarchical grid with square of increasing size. The location service is offered by location servers assigned across the grid, storing node location information. Nodes that happen to be located in these regions offer the location service. The selection mechanism for the predetermined regions is carried out through a hash function, which maps node identifiers to region addresses. DLM distinguishes between a full length address policy and a partial length address policy. Under the full length address policy, location servers store the precise position of nodes. When nodes change regions due to mobility, it is necessary to update all location servers. Under the partial length address policy, the accuracy of node location information stored at the location servers increases along with the proximity of the location servers to the nodes. To the contrary of the full length address policy, several queries are necessary to locate a node. Nevertheless, the partial addressing scheme offers overall increasing performance, since it reduces the scope and frequency of location server updates due to node mobility.

HLS (*Hierarchical Location Service*) [4] divides the area covered by the network into a hierarchy of regions. The lowest level regions are called cells. Regions of one level are aggregated to form a region on the next higher level of the hierarchy. Regions on the same level of the hierarchy do not overlap. For any given node A, one cell in each level of the hierarchy is selected by means of a hash function. These cells are called the responsible cells for node A. As a node moves through the area covered by the network, it updates its responsible cells with information about its current position. When another node B needs to determine those cells that may potentially be responsible for A, it then queries those cells in the order of the hierarchy, starting with the lowest level region. There are two different methods for HLS to update location servers, the direct location scheme and the indirect location scheme. To update its location servers according to the direct location scheme, a node computes its responsible cells. Position updates are then sent to all responsible cells. The location information in the responsible cells is represented as a pointer to the position of the node. The network load can be reduced with the indirect location scheme where the location servers on higher hierarchy levels only know the region of the next lower level a node is located in. More precise location information is not necessary on higher levels.

3 Fuzzy Hierarchical Location Service

We present an adaptive location service on the basis of fuzzy logic called FHLS (*Fuzzy Hierarchical Location Service*) to minimize the sum of location update cost and paging cost. FHLS is adapted to the location update rate and call arrival rate in an environment where the characteristic of nodes movement changes all the time. The FHLS uses the fuzzy control logic for location update. The input parameters of the fuzzy logic are the linguistic variables of location update rate and call arrival rate for a mobile node, and the output is a direct location update scheme or an indirect location update scheme on the basis of a different level in hierarchical region.

3.1 Area Partitioning

FHLS partitions the area containing the ad-hoc network in cells. This partitioning must be known to all participating nodes. The shape and size of the cells can be chosen arbitrary according to the properties of the network. The only prerequisite is that a node in a given lowest-level cell must be able to send packets to all other nodes in the same cell.

The cells are grouped hierarchically into regions of different levels. A number of cells form a region of level one, a number of level-one regions forms a level-two region and so on. Regions of the same level must not intersect, i.e., each region of level n is a member of exactly one region of level n+1. An example for the area partitioning is shown in Fig. 1.

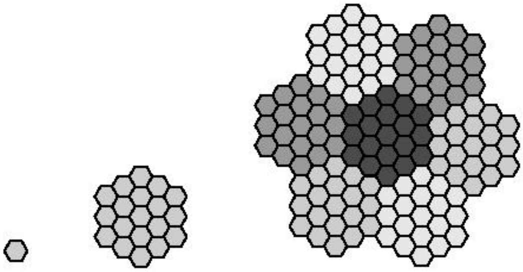


Fig. 1. Grouping cells to form regions: Cell, region on level one and region on level two

3.2 Location Service Cells

FHLS places location information for a node T in a set of cells. We call these cells the LSC (location service cells) of T. A node T selects one LRC for each level in the hierarchy. For a given level n, the LRC is selected according to the following algorithm:

1. Compute the set *S*(*T, n*) of cells belonging to the region of level n which contains T.
2. Select one of these cells with a hash function based on the characteristics of S and the node ID of T.

A possible hash function is the simple, modulo-based function: $H(T,n) = ID(T) \bmod \|S(T,n)\|$. With the number of calculated with this hash function, one of the cells in *S*(*T, n*) is chosen. As a result of the above selection, T has exactly one location service cell on each level n and it is guaranteed that the LRC for T of level n and node T share the same level-n region. An example for the selection of LSCs for a three-level hierarchy is shown in Fig. 2. All location service cells are candidates for location service. These candidate cells, tree-like structure are called candidate tree.

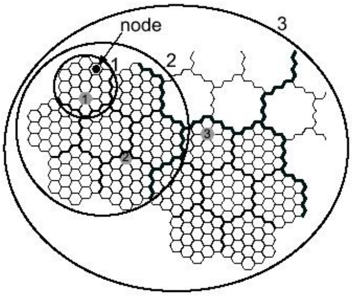


Fig. 2. Example for location service cells of a node

3.3 Location Updates

We propose the adaptive location update scheme using the fuzzy logic on the basis of the mobility and the call preference for mobile nodes. The input parameters of the fuzzy logic are the linguistic variables of location update rate and call arrival for a mobile node, and the output is a direct location update scheme or an indirect location update scheme on the basis of a different level in the hierarchy.

The mobility of a mobile node is measured by location update rate per minute. We map the update rate to the linguistic variable for the node mobility using the membership function as shown in Fig. 3.

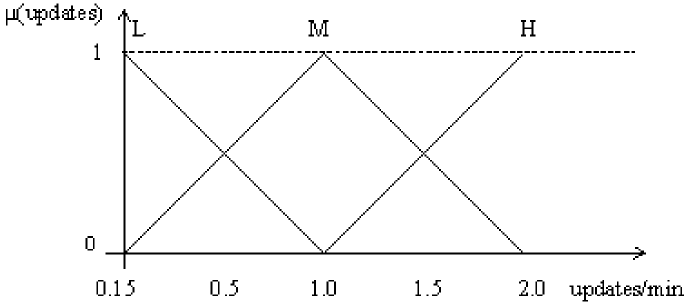


Fig. 3. The membership function and the linguistic variables for node mobility

The call preference of a mobile node is measured by call arrival rate per minute. We map the call arrival rate to the linguistic variable for the node preference using the membership function as shown in Fig. 4.

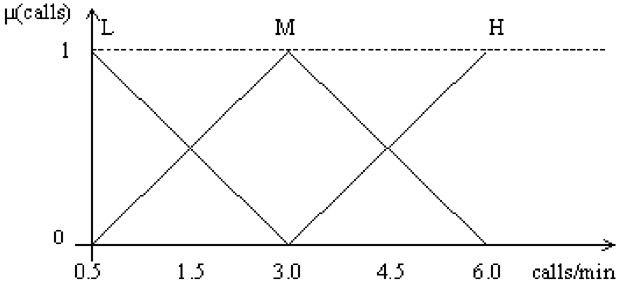


Fig. 4. The membership function and the linguistic variables for node preference

Fuzzy logic uses linguistic variable to map the input fuzzy variable to the output variable. This is accomplished by using IF-THEN rules [5, 6]. We use 9 fuzzy rules described in Table 1. According to the mobility and the preference for a mobile node, a direct location update scheme or an indirect location update scheme on the basis of a different level in the hierarchy is used for updating the location information of the mobile node. For a mobile node, if the linguistic variable for node mobility is M and the linguistic variable for node preference is M, FHLS uses the level-i indirect location update scheme for updating the location information of the mobile node, where the input parameter for FHLS, the level-i represents the level of the first location service cell in the hierarchy.

Table 1. Fuzzy control rules for location update

Mobility Preference	L	M	H
L	level-(i+1) direct location update scheme	level-(i+1) indirect location update scheme	level-(i+1) indirect location update scheme
M	level-i direct location update scheme	level-i indirect location update scheme	level-i indirect location update scheme
H	level-(i-1) direct location update scheme	level-(i-1) direct location update scheme	level-(i-1) indirect location update scheme
(Input variables) L – Low, M – Medium, H – High (Output variables) a direct location update scheme or an indirect location update scheme on the basis of a different level in the hierarchy			

3.4 Location Requests

To successfully query the current location of a target node T, the request of a source node S needs to be routed to a location server of T. When querying the position of T, S knows the ID of T and therefore the structure of the candidate tree defined via the hash function and T's ID. It furthermore knows that T has selected a LSC for each region it resides in. Thus, the request only needs to visit each candidate cell of the regions containing S.

S computes the cell that T would choose as a location service cell. If the location service cell were in the same level-one region, the S sends its request to this cell. When the request packet arrives at the first node A within the boundaries of the candidate cell, it is processed as follows:

1. Node A broadcasts the request to all nodes within the candidate cell. This is called cellcast request.
2. Any node that receives this cellcast request and has location information in its location database sends an answer to A.
3. If A receives an answer for its cellcast request, the request is forwarded to the target node T.
4. Otherwise A forwards the request to the corresponding candidate cell the next level.
5. With this mechanism, the request is forwarded from candidate cell to candidate cell until a location server for T is founded or the highest level candidate cell has been reached.

4 Performance Evaluation

We evaluate the performance of the FHLS in terms of the total cost that consists of location update and paging costs. Table 2 shows the parameter values for the simulation.

Table 2. Simulation parameters

Parameters	Values
The number of cells	64
Cell size (kilometer×kilometer)	2×2
Update rate per minute	random (0.15~2.0)
Call rate per minute	random (0.5~6.0)
Simulation time (minute)	100

The simulation result of total cost over time is illustrated in Fig. 5. As shown in the Fig. 5, the performance of the proposed FHLS is better than that of existing HLS. This is because the FHLS uses an adaptive location update scheme according to the characteristic of mobile node that is location update rate and call arrival rate. Also, the performance of the FHLS (level=4) that places the first location service cell on level 4 in the hierarchy is superior to that of the FHLS (level=3) that places the first location service cell on level 3 in the hierarchy. This is because as the level of the first location service cell increasing, the number of location service cells is decreased and the paging cost is reduced by the spatial locality in location query.

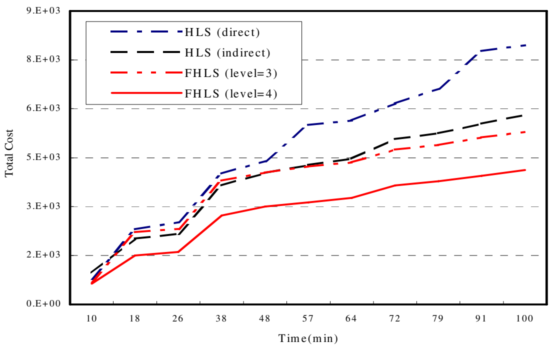


Fig. 5. Total cost for location service

Additionally, we know that the performance of the HLS (indirect) that uses indirect location update scheme is better than that of HLS (direct) that uses direct location update scheme.

5 Conclusions

In this paper, we have presented the Fuzzy Hierarchical Location Service (FHLS) for MANETs. FHLS uses the adaptive location update scheme using the fuzzy logic on the basis of the mobility and the call preference for mobile nodes. The input parameters of the fuzzy logic are the linguistic variables of location update rate and call arrival rate for a mobile node, and the output is a direct location update scheme or an indirect location update scheme on the basis of a different level in the hierarchy. The performance of the FHLS has been evaluated by using a simulation. Because the FHLS uses an adaptive location update scheme according to the characteristic of mobile node that is location update rate and call arrival rate, the performance of the proposed FHLS is better than that of existing HLS. Also, we know that the performance of the FHLS (level=4) that places the first location service cell on level 4 in the hierarchy is superior to that of the FHLS (level=3) that places the first location service cell on level 3 in the hierarchy. As part of our future work, we plan to evaluate our proposed FHLS in terms of various factors i.e., the number of cells, cell sizes, etc.

References

1. Basagni, S., Chlamtac, I., Syrotiuk, V. R., and Woodward, B. A.: A Distance Routing Effect Algorithm for Mobility (DREAM). Proceeding of the fourth annual ACM/IEEE International Conference on Mobile computing and networking (1998) 76-84
2. Li, J., Jannotti, J., DeCouti, D. S. J., Karger D. R. and Morris, R.: A Scalable Location Service for Geographic Ad Hoc Routing. Proceeding of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (2000) 120-130
3. Xue, Y., Li. B. and Nahrstedi. K.: A Scalable Location Management Scheme in Mobile Ad-hoc Networks. Proceeding of the IEEE Conference on Local Computer Networks (2002) 102-111
4. Kieb W., Fubler H, Widmer and Mauve M.: Hierarchical Location Services for Mobile Ad-Hoc Networks. Mobile Computing and Communication Review 8(4) (2004) 47-58
5. Baldwin, J. F.: Fuzzy logic and fuzzy reasoning. In Fuzzy Reasoning and It's Applications, E. H. Mamdani and B. R. Gaines (eds), Academic Press, London (1981)
6. Bae, I-H, Oh, S-J and Olariu, S.: A Fuzzy Logic-Based Location Management Method for Mobile Networks. software Engineering Research and Application, Springer-Verlag Berlin Heidelberg (2004) 304-319
7. Karp, B. N. and Kung, H. T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Proceeding of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (2000) 243-254

Part VIII

Intelligent Manufacturing and Scheduling

Fuzzy Logic Based Replica Management Infrastructure for Balanced Resource Allocation and Efficient Overload Control of the Complex Service-Oriented Applications*

Jun Wang, Di Zheng, Huai-Min Wang, and Quan-Yuan Wu

School of Computer Science, National University of Defence Technology,
Changsha, Hunan, China 410073
junwang@nudt.edu.cn

Abstract. For the complex service-oriented applications, the applications may be integrated by using the services across Internet, thus we should balance the load for the applications to enhance the resource's utility and increase the throughput. To overcome the problem, one effective way is to make use of load balancing. Kinds of load balancing middleware have already been applied successfully in distributed computing. However, they don't take the services types into consideration and for different services requested by clients the workload would be different out of sight. Furthermore, traditional load balancing middleware uses the fixed and static replica management and uses the load migration to relieve overload. However, for many complex service-oriented applications, the hosts may be heterogeneous and decentralized at all and load migration is not efficient for the existence of the delay. Furthermore, due to the global state uncertainty, there is no suitable mathematical model to characterize network behavior to predict the accurate task placement decision. Thus, we employ a fuzzy logic based autonomic replica management infrastructure to support fast response, hot-spot control and balanced resource allocation among different services. Corresponding simulation tests are implemented and their result s indicated that this model and its supplementary mechanisms are suitable to complex service-oriented applications.

Keywords: Web Service, Fuzzy Logic, Load Balancing, Adaptive Resource Allocation, Middleware.

1 Introduction

With the rapid development of the high performance applications, more and more complex problems can't be resolved by just one super computer. Users hope multiple distributed and heterogeneous resources can be organized by high speed networks to resolve the problems together. And application developers could often assume a target environment that was homogeneous, reliable, secure and centrally managed. So the

* This work was funded by the National Grand Fundamental Research 973 Program of China under Grant No.2005cb321804, the National High-Tech Research and Development Plan of China under Grant No.2004AA112020.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 569–579, 2007.
springerlink.com

web service was introduced to denote a proposed distributed computing infrastructure for advanced science and engineering. All the distributed resources including computing resources, storing resources, data resources and the information resources are integrated to support the development of different applications. Service-oriented applications provide users with a variety of services through the web interface. The services are executed by using heterogeneous back-end resources such as high performance systems, mass storage systems, database system etc. However, some complex applications may be integrated across the Internet by using the services and the distributed services and resources must be scheduled automatically, transparently and efficiently. Therefore, we must balance the load of the diverse resources to improve the utilization of the resources and the throughput of the systems. Currently, load balancing mechanisms can be provided in any or all of the following layers in a distributed system:

- **Network-based load balancing:** This type of load balancing is provided by IP routers and domain name servers (DNS). However, load balancing at these layers is somewhat limited by the fact that they do not take into account the content of the client requests.
- **OS-based load balancing:** At the lowest level for the hierarchy, OS-based load balancing is done by distributed operating system in the form of lowest system level scheduling among processors [3, 4].
- **Middleware-based load balancing:** This type of load balancing is performed in middleware, often on a per-session or a per-request basis. The key enterprise applications of the moment such as astronavigation, telecommunication, and finance all make use of the middleware based distributed software systems to handle complex distributed applications.

The middleware-based load balancing resides between network level and application level. There are different realizations of load balancing middleware. For example, stateless distributed applications usually balance the workload with the help of naming service [6]. But this scheme of load balancing just support static non-adaptive load balancing and can't meet the need of complex distributed applications. For more complex applications, the adaptive load balancing schema [7, 8, 9] is needed to take into account the load condition dynamically and avoid override in some node. However, many of the services are not dependable for loose coupling; high distribution and traditional load balancing middleware pay no attentions to the resource allocation. Furthermore, traditional load balancing middleware uses the fixed and static replica management and uses the load migration to relieve overload. However, to the complex service-oriented applications, the hosts may be heterogeneous and decentralized at all and load migration is not efficient for the existence of the delay. Furthermore, due to the global state uncertainty, there is no suitable mathematical model to characterize network behavior to predict the accurate task placement decision. Therefore, we put forward an fuzzy logic based autonomic replica management infrastructure based on middleware to support fast response, hot-spot control and balanced resource allocation among different services.

2 Architecture of the Load Balancing Middleware

Our middleware will directly address to the problems by providing load balancing for the service-oriented applications, preventing bottlenecks at the application tier, balancing the workload among the different services and enabling replication of service components in a scalable way to provide more access to the high performance back end resources. Furthermore, the group of replicas of each service can be used to standby for any fail over. The service components are object-based components and they can be distributed or remotely located in different resources. Our load balancing service is a system-level service and it is introduced to the application tier by using IDL [1, 2] interfaces. Figure 1 features the core components in our load balancing service s follows:

Service Replica Repository: Instances of services need to register with the Service Group. All the references of the groups are stored in the Service Replica Repository. A service group may include several replicas and we can add or remove replicas to the groups. The main purpose of the service group is to provide a view containing simple information about the references to the locations of all replicas registered with group. The uses need not to know where the replica is located.

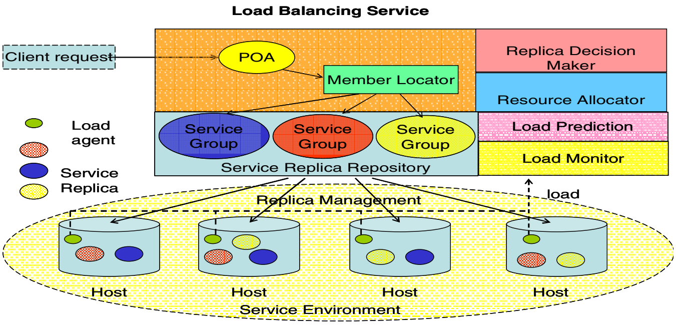


Fig. 1. Components of the Load Balancing Middleware

Member Locator: This component identifies which replica will receive the client request and it is also in charge of binding the clients to the identified replicas. The replica locator can be implemented portably using standard CORBA portable object adapter (POA) mechanisms.

Service Decision Maker: This component assigns the best replica in the group to service the request. The service decision maker is in charge of getting the best replica for the service and the client requests proceed with their normal procedure such as calling methods from the service replica. The decision maker makes decisions based on the algorithms configured in our load balancing policy [10].

Load Monitor: Load monitor collects load information from every load agent within certain time interval. The load information should be refreshed at a suitable interval so that the information provided is not expired. The hosts may have different workload when processing some certain client requests and the workload of the different hosts may fluctuate in different ways. Some hosts may be below threshold while the others may be above threshold or idle.

Load Agent: The purpose of load agent is to provide load information of the hosts it resides when requested by load monitor. As different services might have replicas in the same host, it is hard to presume the percentage of resource is being used by which service at particular moment. Therefore, a general metric is needed to indicate the level of available resources at the machine during particular moment.

Load Prediction: Many existing load balancing middleware use the *dampening* technology to make the load to be predicative. However, distributed systems are inherently difficult to manage and the dampening factor cannot be treated as static and fixed. The dampening factor should be adjusted dynamically according to different load fluctuate. So in this module we use the machine-learning based load prediction method where the system minimizes unpredictable behavior by reacting slowly to changes and waiting for definite trends to minimize over-control decisions.

Resource Allocator: The purpose of this component is to dynamically adjust the resource to achieve a balance load distribution among different services. In fact, we control the resource allocation by managing the replicas of different services. For example, this component makes decisions on some mechanisms such as service replication, services coordination, dynamic adjustment and requests prediction. In order to carry out service replication, we determine parameters such as initial number of replica, maximum number of replica, when to replicate, where to replicate etc.

3 Fuzzy-Logic Based Autonomic Replica Management

3.1 Motivation

In traditional load balancing middleware, the requests just are distributed among the different replicas to balance the workloads and the location as well as the number of the replicas is fixed. Therefore, it is a static resource allocation and the requests will be distributed to different hosts by the Service Decision Maker component according to the strategies plugged in it. The requests are treated in the same way and the workload can be effectively balanced when the requests are similar and even. However, the resource different requests needed are different at all and in some occasions such as 911 or world cup the number of some kinds of requests will increase fast while the number of them will be small in most of the days. When demand increases dramatically for a certain service, the quality of service (QoS) deteriorates as the service provider fails to meet the demand. At the same time, the hosts are unstable and some replicas may be in failure frequently. In these occasions many of the hosts will become overload and all the system may be unstable. To the traditional load balancing middleware, load migration will be made use of to relieve overload. However, to the complex service-oriented applications, the hosts may be heterogeneous and decentralized at all and load migration may be useless if there are no certain replicas in the less loaded hosts. Furthermore, to the stateful applications the control of the load migration is complex. Therefore we use the adaptive scheme to adjust the number of the replicas so as to realize the adaptive resource allocation.

At the same time, due to the global state uncertainty, there is no suitable mathematical model to characterize network behavior to predict the accurate task placement decision. Thus, in order to tackle the load balancing problem in such an environment, we employ fuzzy rules to model those sources that cause uncertainty in global states. The rule base contains a set of fuzzy if-then rules which defines the actions of the controller in terms of linguistic variables and membership functions of linguistic terms. The fuzzy inference engine applies the inference mechanism to the set of rules in the fuzzy rule base to produce a fuzzy set output. This involves matching the input fuzzy set with the premises of the rules, activation of the rules to deduce the conclusion of each rule that is fired, and combination of all activated conclusions using fuzzy set union to generate fuzzy set output. The purpose of the fuzzy logic based Resource Allocator component is as follows:

1. Managing the number and the location of the different services dynamically and efficiently to realize balanced resource allocation.
2. Bringing down the extra overhead.
3. Avoiding the overload.

3.2 Load Metrics

To the complex service-oriented applications, the hosts may be heterogeneous and there may be different kinds of service instances residing in the same host. How to measure the load is an important problem to be discussed. In traditional distributed systems, the load can be measured by the CPU utilization, the memory utilization, the I/O utilization, the network bandwidth and so on. At the same time, the load may be different for different applications. For example, the computing-centric applications may raise the CPU utilization faster while the data-centric applications may exhaust the memory faster. Different types of workloads and services are also closely related to the load balancing. The studies in [14] only discuss for I/O intensive workloads, and a new I/O-aware load-balancing scheme known for weighted average load-with Pre-emptive Migration is presented in [12]. Chen et al. [13] take the contents/services types into consideration for balancing workloads of scalable web server clusters, and the paper in [11] considers load sharing policies for memory intensive workloads on heterogeneous clusters. Hence we will propose our dynamic load balancing scheme, taken multiple resources and mixed workloads into consideration. The load index of each node is composed of composite usage of different resources including CPU, memory, and I/O, which are contributed to different workloads.

Therefore, the load of any node can be described as a triple $\left\langle K_1, K_2, K_3 \right\rangle$ and k_i is one of the load metrics as follows:

1. The CPU utilization.
2. The memory utilization.
3. The I/O utilization.

However, to different applications some certain load metric may be more important. Therefore, the compute capability of a node is a weighted function of load index and the total weighted value can be calculated with:

$$L_j = \sqrt{\sum_{i=1}^{3} a_i \, k_i^2} \quad (0 \le a_i \le 1, \ \sum_{i=1}^{3} a_i = 1) \tag{1}$$

In this equation, L_j denotes the load of host j and k_i denotes the percentage the according resource has been exhausted. For example, supposing the CPU utilization of the node j is ninety percent, then the k_1 of the node will be 0.9. Furthermore, a_i denotes the weighted value of the certain load metric and the value can be configured differently for diverse applications.

3.3 Implementations of the Autonomic Replica Management

In fact, to the service-oriented applications, the resource allocation is controlling the location and the number of the service replicas. At the beginning of the discussion let us give some definitions firstly. Let $H = \{h_1, h_2 ..., h_i\}$ where h_j represents the j^{th} node of the system and let $S = \{s_1, s_2 ..., s_l\}$ where s_k represents the k^{th} service of the system. Furthermore, let N_k represents the number of the replicas of the k^{th} service of the system. So the set of the replicas of the k^{th} service can be denoted by $R(S_k) = \{S_{k1}, ... S_{kN_k}\}$. At the same time, the host the m^{th} replica of the l^{th} service is residing in can be denoted by $H(S_{lm})$ whose load at time t is denoted by $L_{H(S_{lm})}(t)$.

The first problem is when to create the new replica. In normal conditions, new requests will be dispatched to the fittest replicas. But all the hosts the initial replicas residing in may be in high load and the new requests will cause them to be overload. So we should create new replicas to share the workload. We set the ***replica_addition*** threshold to help to make the decisions. For example, to the i^{th} service of the system, if the equation (**2**) can be true, then new replica will be created.

$$\forall x (L_{H(S_{ix})}(t) \ge replica\_addition) \;\; x \in R(S_i) \tag{2}$$

The second problem is where to create the new replica. As the load metrics we have discussed before we can compute the workloads of the hosts. Furthermore, the hosts may be heterogeneous and the workload of each host is different. In fact, we set the ***replica_deployment*** threshold for every host. According to the equation (**3**), if the workloads of some hosts don't exceed the threshold the new replica can be created on them. Otherwise no replica will be created because of the host will be overloaded soon and all the system will become unstable for the creation. Therefore the incoming requests should be rejected to prevent failures.

$$\exists x (L_{H_x}(t) < replica\_deployment) \;\; x \in \{1, ... i\} \tag{3}$$

The third problem is to create what kind of replicas. Because the applications may be composed of different services and the services may all need create new replicas.

However, the services may have different importance and we should divide them with different priorities. Therefore, we classify the services as high priority services, medium priority services and low priority services according to the importance of them. The services having different priorities may have different maximum replicas. For example, supposing the number of the hosts is *n*, then the maximum number of the high priority can be *n,* the maximum number of the medium priority service can be $\lfloor \frac{n}{2} \rfloor$ and the maximum number of the low priority service can be $\lfloor \sqrt{n} \rfloor$ .These configurations can be revised according to practical needs. Secondly, the Resource Allocator module maintains three different queues having different priorities. Each queue is managed by the FIFO strategy and the replicas of the services having higher priority will be created preferentially.

The last problem is the elimination of the replicas. For the coming of the requests may fluctuate. If the number of the requests is small, then monitoring multiple replicas and dispatching the requests among them is not necessary and wasteful. We should eliminate some replicas to make the system to be efficient and bring down the extra overhead. So we set the ***replica_elimination*** threshold to control the elimination of the replicas. For example, to the i^{th} service of the system, if the equation (4) can be true, then some certain replica will be eliminated.

$$\exists x(L_{H(S_{ix})}(t) < replica\_elimination) \;\; x \in R(S_i) \tag{4}$$

The elimination will be performed continuously until the equation (4) becomes false or the number of the replicas becomes one.

By the way we have discussed before, the resource can be allocated among the services efficiently. However, in some unusual occasions such as 911 and world cup some certain simple services will become hot spot. At the same time, to some services the client requests may fluctuate irregularly and the services may become hot spot for some certain time. For example, some services may have lots of client requests on Monday while on the other days there may only be a few client requests. Therefore, we should adjust the priority of the services according to the number of the incoming requests to avoid overload. The low priority services may have higher priority with the increasing client request and more replicas will be created to response the requests. At the same time, once the number of the requests decreases, the priority of these services will be brought down and the exceeding replicas will be eliminated.

4 Performance Results

As depicted in figure 2(a), our middleware StarLB is running on RedHat Linux 9/Pentium4/256/1.7G. The clients are running on Windows2000/Pentium4/512M/2G. Furthermore, to compare the results easier the Grid hosts are as same as each other and all of the hosts are running on Windows2000/Pentium4/512M/2G.

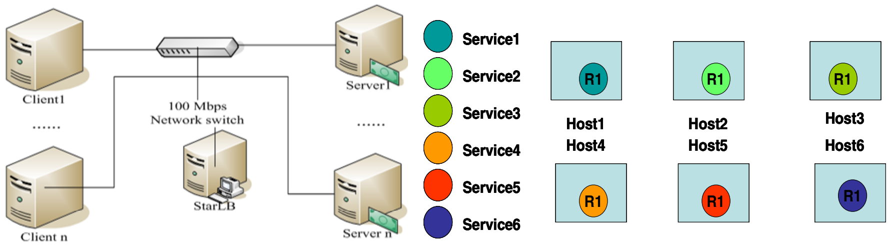


Fig. 2. (a) Load Balancing Experiment Testbed. (b) Initial Replicas of different Services.

At the beginning of this test, we used the services without the help of the autonomic replica management. Supposing there are six hosts and there are six different services. (This test just be used to analyze and present the mechanisms and more complex tests using many more hosts and services are ignored here.) Among these services there are two high priority services, two medium priority services and two low priority services. Among the six services the service1 and the service2 are high-priority services, the service3 and the service4 are medium-priority services and the remaining two are low-priority services. All the services have only one initial replica. Each replica resides in a host respectively and response to the client requests. The distribution of the replicas is as depicted in figure 2(b).

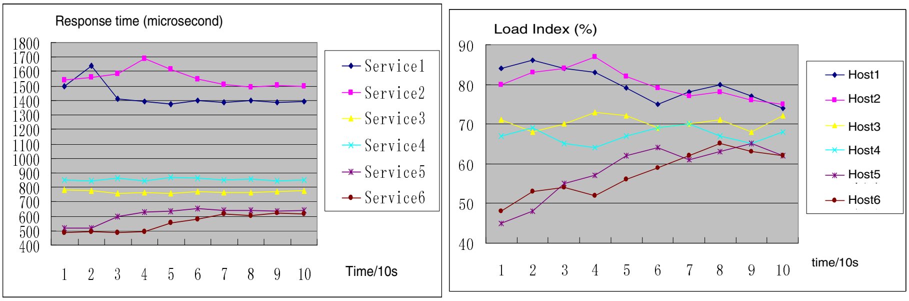


Fig. 3. (a) Response Time with Replica Management. (b) Load Index with Replica Management.

Furthermore, we set the low priority service can have at most two replicas, the medium priority service can have three replicas and the high priority service can have six replicas. As depicted in figure 3(a) and figure 3(b), according to our setting when the load index arrived at 85% new replica was created. From the two above broken lines in figure 4(a) we can see with the requests coming new replicas was created in the hosts the low priority services residing in. At the same time, because of the creation of new replicas the response time of the high priority services could be brought down and the throughput of these services was increased efficiently. Furthermore, the workload of all the hosts was balanced efficiently. All the creations are depicted in figure 4(a) and the services with higher priority may create new replicas more preferentially and have larger maximum replica number.

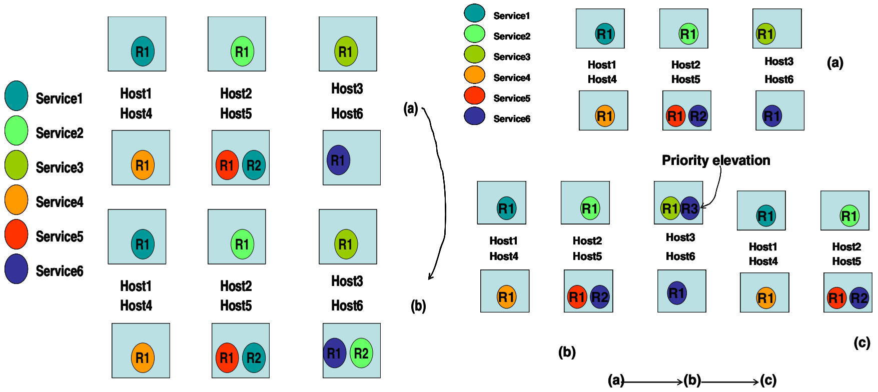


Fig. 4. (a) Creation and Elimination of the Replicas. (b) Creation and Elimination of the Replicas with Priority Elevation.

Then there is still a question to be discussed. That is the elimination of the extra replicas and the elevation of the priority. We deployed all the replicas as the initial state depicted in the figure 2(b) and made the service6 become hot spot. As the figure 5(a) and the figure 5(b) depicted, adding the number of the requests of the services6 gradually. Then the CPU utilization of the Host6 and the response time of the service6 increased too. According to our setting of the ***replica_addition*** threshold, when the Load index arrived at 85% new replica was created. For the load index of the Host5 was lowest, a new replica of the service6 was created in the Host5. By the creation of the new replica, the Load index of the Host6 decreased as well as the response time. At the same time, the response time of the service5 was just affected a little.

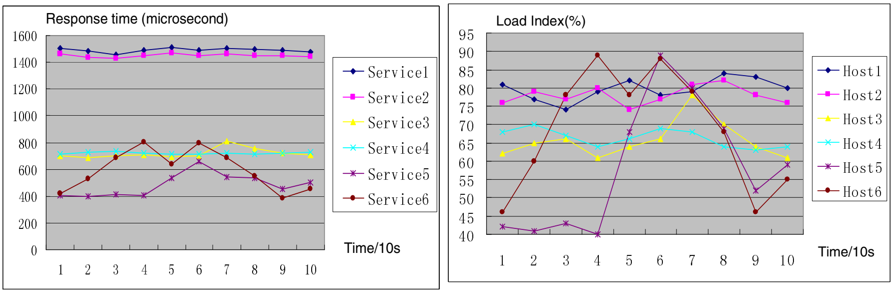


Fig. 5. (a) Response Time with Priority Elevation. (b) Load index with Priority Elevation.

However, because the requests for the service 6 kept increasing and the Load index arrived at 85% again. As we have discussed before the service6 is low priority service and the maximum number of the replicas is two. So the priority of the service6 should be elevated to allow the creation of the new replicas. Then the new replica was created in the Host3 and the requests could be distributed with the help of the new replica. At last, when we decreased the requests of the service6. The Load index decreased and too many replicas were not necessary. So the priority of the service6 should be decreased and unnecessary replicas should be eliminated. As depicted in the figure 5(b), when the Load index was below the ***replica_elimination*** threshold the replica in the Host3 was eliminated for the highest Load index among the three replicas. The remaining two replicas were keep dealing with the requests until the Load index shall become higher or lower. All the creation and elimination of the replicas with priority elevation are depicted in the figure 4(b).

Besides the discussion above, more complex tests have been completed. In fact, with the addition of the number of the hosts and the services, the resources can still be allocated efficiently and the workload of the resources can be balanced. Especially to the hot-spot services sharing the workloads with the extra replicas is much better than load migration among the overloaded servers.

5 Conclusions

For the complex service-oriented applications, the applications may be integrated by using the services across Internet, thus we should balance the load for the applications to enhance the resource's utility and increase the throughput. To overcome the problem, one effective way is to make use of load balancing. Kinds of load balancing middleware have already been applied successfully in distributed computing. However, they don't take the services types into consideration and for different services requested by clients the workload would be different out of sight. Furthermore, traditional load balancing middleware uses the fixed and static replica management and uses the load migration to relieve overload. However, for many complex service-oriented applications, the hosts may be heterogeneous and decentralized at all and load migration is not efficient for the existence of the delay. Thus, we employ a fuzzy logic based autonomic replica management infrastructure to support fast response, hot-spot control and balanced resource allocation among different services. Corresponding simulation tests are implemented and their result s indicated that this model and its supplementary mechanisms are suitable to complex service-oriented applications.

References

1. Object Management Group, The Common Object Request Broker: Architecture and Specification, 3.0 ed., June 2002.
2. M. Henning and S. Vinoski, Advanced CORBA Programming With C++. Addison-Wesley Longman, 1999.
3. R. Chow and T. Johnson, "Distributed Operating Systems and Algorithms", Addison Wesley Long, Inc., 1997.
4. Rajkumar Buyya. "High Performance Cluster Computing Architecture and Systems", ISBN7.5053-6770-6.2001.
5. S. M. Baker and B. Moon, "Distributed Cooperative Web Servers", In Proc. of The Eight International WWW Conference, Toronto, Canada, May 11-14, 1999.
6. IONA Technologies, "Orbix 2000." www.iona-iportal.com/suite/orbix2000.htm.
7. Othman, C. O'Ryan, and D. C, Schmidt, "The Design of an Adaptive CORBA Load Balancing Service," IEEE Distributed Systems Online, vol. 2, Apr, 2001.

8. Othman O, Schmidt DC. Issues in the design of adaptive middleware load balancing. In: ACM SIGPLAN, ed. Proceedings of the ACM SIGPLAN workshop on Languages, Compilers and Tools for Embedded Systems. New York: ACM Press, 2001. 205~213.
9. Othman O, O'Ryan C, Schmidt DC. Strategies for CORBA middleware-based load balancing. IEEE Distributed Systems Online, 2001, 2(3). http://www.computer.org/dsonline.
10. Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison-Wesley, 2002. 223~325.
11. Li Xiao, Xiaodong Zhang,and Yanxia Qu. "Effective load sharing on heterogeneous networks of workstations". In Proceedings of the 14th International Parallel and Distributed Processing Symposium. May 2000,pp, 431-438.
12. Xiao Qin, Hong Jiang, Yifeng Zhu, and David R.Swanson."Boosting performance for I/O-intensive workload by preemptive job migration in a cluster system". In Proceedings of the 15t Symposium on Computer Architecture and High Performance Computing. Nov 2003, pp, 235-243.
13. Tzung-Shi Chen, and Kuo-Lian Chen. "Balancing workload based on content types for scalable web server clusters". In Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04), 2004, 2:321-325.
14. Xiao Qin, Hong Jiang, Yifeng Zhu, and David R.Swanson. "A dynamic load balancing scheme for I/O-intensive applications in distributed systems". In Proceedings of the 2003 International Conference on Parallel Processing Workshops. Oct 2003, pp.79-86.

A Fuzzy-Neural Approach with BPN Post-classification for Job Completion Time Prediction in a Semiconductor Fabrication Plant*

Toly Chen

Department of Industrial Engineering and Systems Management, Feng
Chia University, 100, Wenhwa Road, Seatwen, Taichung City, Taiwan
tolychen@ms37.hinet.net
http://www.geocities.com/tinchihchen/

Abstract. Predicting the completion time of a job is a critical task to a semiconductor fabrication plant. Many recent studies have shown that pre-classifying a job before predicting the completion time was beneficial to prediction accuracy. However, most classification approaches applied in this field could not absolutely classify jobs. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For tackling these problems, a fuzzy-neural approach with back-propagation-network (BPN) post-classification is proposed in this study, in which a job is post-classified with some BPNs instead after predicting its completion time with a fuzzy BPN (FBPN). In this novel way, only jobs which estimated completion times are the same accurate will be clustered into the same category. To evaluate the effectiveness of the proposed methodology, production simulation is applied to generate test data. According to experimental results, post-classifying jobs might be very effective in enhancing the accuracy of job completion time prediction in a semiconductor fabrication plant.

1 Introduction

Predicting the completion time for every job in a semiconductor fabrication plant is a critical task not only to the plant itself, but also to its customers. After the completion time of each job in a semiconductor fabrication plant is accurately predicted, several managerial goals can be simultaneously achieved [5]. Predicting the completion time of a job is equivalent to estimating the cycle (flow) time of the job, because the former can be easily derived by adding the release time (a constant) to the latter. There are six major approaches commonly applied to predicting the completion/cycle time of a job in a semiconductor fabrication plant: multiple-factor linear combination (MFLC), production simulation (PS), back propagation networks (BPN), case based reasoning (CBR), fuzzy modeling methods, and hybrid approaches. Among the six approaches, MFLC is the easiest, quickest, and most prevalent in practical applications. The major disadvantage of MFLC is the lack of forecasting accuracy [5]. Conversely, huge amount of data and lengthy simulation time are two disadvantages of

* This work was support by the National Science Council, R.O.C.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 580–589, 2007.
springerlink.com

PS. Nevertheless, PS is the most accurate completion time prediction approach if the related databases are continuously updated to maintain enough validity, and often serves as a benchmark for evaluating the effectiveness of another method. PS also tends to be preferred because it allows for computational experiments and subsequent analyses without any actual execution [3]. Considering both effectiveness and efficiency, Chang et al. [4] and Chang and Hsieh [2] both forecasted the completion/cycle time of a job in a semiconductor fabrication plant with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy measured with root mean squared error (RMSE) was considerably improved with these BPNs. On the other hand, much less time and fewer data are required to generate an completion time forecast with a BPN than with PS. Chang et al. [3] proposed a k-nearest-neighbors based case-based reasoning (CBR) approach which outperformed the BPN approach in forecasting accuracy. Chang et al. [4] modified the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the fuzzy modeling method proposed by Wang and Mendel [22], called the WM method, with a simple genetic algorithm (GA) and proposed the evolving fuzzy rule (EFR) approach to predict the cycle time of a job in a semiconductor fabrication plant. Their EFR approach outperformed CBR and BPN in prediction accuracy. Chen [5] constructed a fuzzy BPN (FBPN) that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN especially in the efficiency respect. Chen's FBPN was applied to evaluate the achievability of a completion time forecast in Chen [6]. Recently, Chen [9] constructed a look-ahead FBPN for the same purpose, and showed that taking the future release plan into consideration did improve the performance. Chen and Lin [12] proposed a hybrid k-means (kM) and BPN approach for estimating job completion time in a semiconductor fabrication plant. Subsequently, Chen [10] constructed a kM-FBPN system for the same purpose. Similarly, Chen [8] combined SOM and BPN, in which a job was classified using SOM before predicting the completion time of the job with BPN. Chen [7] proposed a hybrid look-ahead SOM-FBPN and FIR system for job completion time prediction and achievability evaluation. The hybrid system outperformed many existing hybrid approaches with more accurate completion time estimation. Subsequently, Chen et al. [11] added a selective allowance to the completion time predicted using Chen's approach to determine the internal due date. The results of these studies showed that pre-classifying jobs was beneficial to prediction accuracy.

However, most classification approaches applied in this field could not absolutely classify jobs. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For tackling these problems, a fuzzy-neural approach with BPN post-classification is proposed in this study, in which a job is post-classified with some BPNs instead after predicting its completion time with a FBPN. In this novel way, only jobs which estimated completion times are the same accurate will be clustered into the same category. To evaluate the effectiveness of the proposed methodology, production simulation is applied to generate test data.

2 Methodology

The parameters used in this study are defined:

1. R_n: the release time of the n-th example/job.
2. U_n: the average plant utilization at R_n.
3. Q_n: the total queue length on the job's processing route at R_n.
4. BQ_n: the total queue length before bottlenecks at R_n.
5. FQ_n: the total queue length in the whole plant at R_n. Obviously, $FQ_n \geq Q_n \geq BQ_n$.
6. WIP_n: the plant work-in-process (WIP) at R_n.
7. $D_n^{(i)}$: the latenesses of the i-th most recently completed jobs.
8. $FDW_n^{(f)}$: the f-th nearest future discounted workload of job n.

The proposed methodology is composed of two steps. Firstly, a FBPN is constructed to predict the completion time of a job.

2.1 Job Completion Time Prediction with a FBPN

The configuration of the FBPN is established as follows:

1. Inputs: eleven parameters associated with the n-th example/job including U_n, Q_n, BQ_n, FQ_n, WIP_n, $D_n^{(i)}$ (i = 1~3), and $FDW_n^{(f)}$ (f = 1~3). These parameters have to be normalized so that their values fall within [0, 1]. Then some production execution/control experts are requested to express their beliefs (in linguistic terms) about the importance of each input parameter in predicting the cycle (completion) time of a job. Linguistic assessments for an input parameter are converted into several pre-specified fuzzy numbers. The subjective importance of an input parameter is then obtained by averaging the corresponding fuzzy numbers of the linguistic replies for the input parameter by all experts. The subjective importance obtained for an input parameter is multiplied to the normalized value of the input parameter.
2. Single hidden layer: Generally one or two hidden layers are more beneficial for the convergence property of the network.
3. Number of neurons in the hidden layer: the same as that in the input layer. Such a treatment has been adopted by many studies (e.g. [2, 5]).
4. Output: the (normalized) cycle time forecast of the example.
5. Network learning rule: Delta rule.
6. Transformation function: Sigmoid function,

$$f(x) = 1/(1 + e^{-x}). \tag{1}$$

7. Learning rate (λ): 0.01~1.0.
8. Batch learning.

The procedure for determining the parameter values is now described. A portion of the examples is fed as "training examples" into the FBPN to determine the parameter values. Two phases are involved at the training stage. At first, in the forward phase, inputs are multiplied with weights, summated, and transferred to the hidden layer. Then activated signals are outputted from the hidden layer as:

$$\tilde{h}_j = (h_{j1},\ h_{j2},\ h_{j3}) = 1/1 + e^{-\tilde{n}_j^h}\ , \tag{2}$$

where

$$\tilde{n}_j^h = (n_{j1}^h,\ n_{j2}^h,\ n_{j3}^h) = \tilde{I}_j^h(-)\tilde{\theta}_j^h\ , \tag{3}$$

$$\tilde{I}_j^h = (I_{j1}^h,\ I_{j2}^h,\ I_{j3}^h) = \sum_{all\ i} \tilde{w}_{ij}^h(\times)\tilde{x}_{(i)}\ , \tag{4}$$

and (−) and (×) denote fuzzy subtraction and multiplication, respectively; $\tilde{h}_j$'s are also transferred to the output layer with the same procedure. Finally, the output of the FBPN is generated as:

$$\tilde{o} = (o_1,\ o_2,\ o_3) = 1/1 + e^{-\tilde{n}^o}\ , \tag{5}$$

where

$$\tilde{n}^o = (n_1^o,\ n_2^o,\ n_3^o) = \tilde{I}^o(-)\tilde{\theta}^o\ , \tag{6}$$

$$\tilde{I}^o = (I_1^o, I_2^o, I_3^o) = \sum_{all\ j} \tilde{w}_j^o(\times)\tilde{h}_j\ . \tag{7}$$

To improve the practical applicability of the FBPN and to facilitate the comparisons with conventional techniques, the fuzzy-valued output $\tilde{o}$ is defuzzified according to the centroid-of-area (COA) formula:

$$o = \mathrm{COA}(\tilde{o}) = (o_1 + 2o_2 + o_3)/4. \tag{8}$$

Then the defuzzified output o is applied to predict the actual cycle time a, for which prediction error E and RMSE are calculated:

$$E = |o - a|. \tag{9}$$

$$RMSE = \sqrt{\sum (o-a)^2 / \text{number of examples}}. \tag{10}$$

Subsequently in the backward phase, the deviation between o and a is propagated backward, and the error terms of neurons in the output and hidden layers can be calculated, respectively, as

$$\delta^o = o(1-o)(a-o)\ , \tag{11}$$

$$\tilde{\delta}_j^h = (\delta_{j1}^h,\ \delta_{j2}^h,\ \delta_{j3}^h) = \tilde{h}_j(\times)(1-\tilde{h}_j)(\times)\tilde{w}_j^o\delta^o\ . \tag{12}$$

Based on them, adjustments that should be made to the connection weights and thresholds can be obtained as

$$\Delta\tilde{w}_j^o = (\Delta w_{j1}^o,\ \Delta w_{j2}^o,\ \Delta w_{j3}^o) = \eta\delta^o \tilde{h}_j , \tag{13}$$

$$\Delta\tilde{w}_{ij}^h = (\Delta w_{ij1}^h,\ \Delta w_{ij2}^h,\ \Delta w_{ij3}^h) = \eta\tilde{\delta}_j^h (\times)\tilde{x}_i , \tag{14}$$

$$\Delta\theta^o = -\eta\delta^o , \tag{15}$$

$$\Delta\tilde{\theta}_j^h = (\Delta\theta_{j1}^h, \Delta\theta_{j2}^h, \Delta\theta_{j3}^h) = -\eta\tilde{\delta}_j^h . \tag{16}$$

Theoretically, network-learning stops when RMSE falls below a pre-specified level, or the improvement in RMSE becomes negligible with more epochs, or a large number of epochs have already been run. In addition, to avoid the accumulation of fuzziness during the training process, the lower and upper bounds of all fuzzy numbers in the FBPN will no longer be modified if Chen's index [5] converges to a minimal value. Then test examples are fed into the FBPN to evaluate the accuracy of the network that is also measured with RMSE.

In addition, the fuzzy-valued output $\tilde{o} = (o_1, o_2, o_3)$ of the FBPN can be thought of as providing a weighted interval forecast for the actual cycle time a, and it becomes possible to further reduce RMSE with such weighted interval forecasts to the following value:

$$RMSE = \sqrt{\sum \min((o_1 - a)^2, (o_3 - a)^2) / \text{number of examples}} . \tag{17}$$

2.2 Job Post-classification

Jobs are post-classified after predicting their cycle times as follows:

1. The main set contains all jobs in the beginning.
2. Sort the data of the jobs in the main set in ascending order of the prediction error (E).
3. Divide the sorted data into two parts.
4. For each of the two parts, assign a code to all the jobs of the part.
5. Construct a new BPN for post-classification that can predict the code of a job according to the eleven attributes.
6. If a number of replications have been reached, stop; otherwise, go to step 7.
7. The main set contains all the jobs in the part which prediction error is higher.
8. Construct a new BPN to predict the cycle time of each job in the main set.
9. Go to step 2.

The methodology is shown in Fig. 1. When a new job is released into the plant, it is firstly classified by BPN #1. If the classification result is the job belongs to part #2, then BPN #2 is applied to classify the new job again; otherwise, FBPN #1 is applied to predict the cycle time of the new job. After the second classification, if the classification result is the job belongs to part #4, then BPN #3 is applied to classify the new job again; otherwise, FBPN #2 is applied to predict the cycle time of the new job, and so forth.

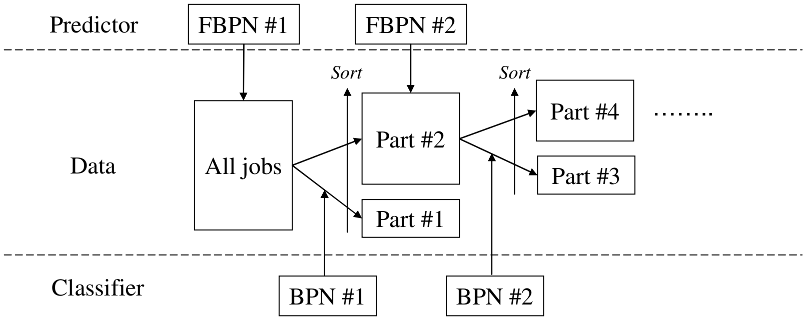


Fig. 1. The methodology architecture

3 A Demonstrative Example from a Simulated Plant

In practical situations, the history data of each job is only partially available in the plant. Further, some information of the previous jobs such as Q_n, BQ_n, and FQ_n is not easy to collect on the shop floor. Therefore, a simulation model is often built to simulate the manufacturing process of a real semiconductor fabrication plant [1-14]. Then, such information can be derived from the shop floor status collected from the simulation model [3]. To generate a demonstrative example, a simulation program coded using Microsoft Visual Basic .NET is constructed to simulate a semiconductor fabrication plant with the following assumptions:

1. Jobs are uniformly distributed into the plant.
2. The distributions of the interarrival times of machine downs are exponential.
3. The distribution of the time required to repair a machine is uniform.
4. The percentages of jobs with different product types in the plant are predetermined. As a result, this study is only focused on fixed-product-mix cases. However, the product mix in the simulated plant does fluctuate and is only approximately fixed in the long term.
5. The percentages of jobs with different priorities released into the plant are controlled.
6. The priority of a job cannot be changed during fabrication.
7. Jobs are sequenced on each machine first by their priorities, then by the first-in-first-out (FIFO) policy. Such a sequencing policy is a common practice in many foundry plants.
8. A job has equal chances to be processed on each alternative machine/head available at a step.
9. A job cannot proceed to the next step until the fabrication on its every wafer has been finished. No preemption is allowed.

The basic configuration of the simulated semiconductor fabrication plant is the same as a real-world semiconductor fabrication plant which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. A trace report was generated every simulation run for verifying the simulation model. The simulated average cycle times have also been compared with the actual values to validate the simulation model. Assumptions

(1)~(3), and (7)~(9) are commonly adopted in related studies (e.g. [2-5]), while assumptions (4)~(6) are made to simplify the situation. There are five products (labeled as A~E) in the simulated plant. A fixed product mix is assumed. The percentages of these products in the plant's product mix are assumed to be 35%, 24%, 17%, 15%, and 9%, respectively. The simulated plant has a monthly capacity of 20,000 pieces of wafers and is expected to be fully utilized (utilization = 100%). Jobs are of a standard size of 24 wafers per job. Jobs are released one by one every 0.85 hours. Three types of priorities (normal, hot, and super hot) are randomly assigned to jobs. The percentages of jobs with these priorities released into the plant are restricted to be approximately 60%, 30%, and 10%, respectively. Each product has 150~200 steps and 6~9 reentrances to the most bottleneck machine. The singular production characteristic "reentry" of the semiconductor industry is clearly reflected in the example. It also shows the difficulty for the production planning and scheduling people to provide an accurate due-date for the product with such a complicated routing. Totally 102 machines (including alternative machines) are provided to process single-wafer or batch operations in the plant. Thirty replicates of the simulation are successively run. The time required for each simulation replication is about 12 minute on a PC with 512MB RAM and Athlon™ 64 Processor 3000+ CPU. A horizon of 24 months is simulated. The maximal cycle time is less than 3 months. Therefore, four months and an initial WIP status (obtained from a pilot simulation run) seemed to be sufficient to drive the simulation into a steady state. The statistical data were collected starting at the end of the fourth month. For each replicate, data of 30 jobs are collected and classified by their product types and priorities. Totally, data of 900 jobs can be collected as training and testing examples. Among them, 2/3 (600 jobs, including all product types and priorities) are used to train the network, and the other 1/3 (300 jobs) are reserved for testing.

3.1 Results and Discussions

To evaluate the effectiveness of the proposed methodology and to make comparisons with four approaches – MFLC, FBPN, CBR, and kM-FBPN, all the five methods were applied to five test cases containing the data of full-size (24 wafers per job) jobs with different product types and priorities. The convergence condition was established as either the improvement in RMSE becomes less than 0.001 with one more epoch, or 1000 epochs have already been run. The minimal RMSEs achieved by applying the five approaches to different cases were recorded and compared in Table 1. RMSE is adopted instead of mean absolute error (MAE) because the same measure has been adopted in the previous studies in this field (e.g. [2-5]), namely, to facilitate comparison. As noted in Chang and Liao [5], the k-nearest-neighbors based CBR approach should be fairly compared with a BPN (or FBPN) trained with only randomly chosen k cases. MFLC was adopted as the comparison basis, and the percentage of improvement on the minimal RMSE by applying another approach is enclosed in parentheses following the performance measure. The optimal value of parameter k in the CBR approach was equal to the value that minimized RMSE [5].

Table 1. Comparisons of the RMSEs of various approaches

RMSE	A(normal)	A(hot)	A(super hot)	B(normal)	B(hot)
MFLC	185.1	106.01	12.81	302.86	79.94
FBPN	171.82(-7%)	89.5(-16%)	11.34(-11%)	286.14(-6%)	76.14(-5%)
CBR	172.44(-7%)	86.66(-18%)	11.59(-10%)	295.51(-2%)	78.85(-1%)
kM-FBPN	157.78(-15%)	54.93(-48%)	9.48(-26%)	197.1(-35%)	42.01(-47%)
proposed methodology	117.85(-36%)	51.28(-52%)	10.28(-20%)	144.82(-52%)	42.23(-47%)

According to experimental results, the following discussions are made:

1. From the effectiveness viewpoint, the prediction accuracy (measured with RMSE) of the proposed methodology was significantly better than those of the other approaches in most cases by achieving a 20%~52% (and an average of 41%) reduction in RMSE over the comparison basis – MFLC. The average advantage over CBR was 30%.
2. The effect of pre-classification is revealed by the fact that the prediction accuracy of kM-FBPN was considerably better than that of FBPN without pre-classification in all cases with an average advantage of 25%.
3. The effect of post-classification is revealed by the fact that the prediction accuracy of the proposed methodology was considerably better than that of FBPN without post-classification in all cases with an average advantage of 32%.
4. The propose methodology surpassed kM-FBPN in most cases, which revealed that post-classification might be more effective than pre-classification for job cycle/completion time prediction in a semiconductor fabrication plant. However, in some cases the effect was not so obvious and therefore a more sophisticated post-classification mechanism has to be developed to further enhance the performance.

In the case of product A with super hot priority, the performance of the proposed methodology was worse than that of the pre-classifying kM-FBPN approach, indicating that post-classifying did not guarantee a better result. Both ways of job classification have to be considered for safety. In addition, there might be a trade-off between the performance of job classification and that of job completion time prediction. Such a relationship needs to be investigated with more experiments.

4 Conclusions and Directions for Future Research

Most classification approaches applied to job completion time prediction in a semiconductor fabrication plant could not absolutely classify jobs. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For these reasons, to further enhance the effectiveness of job completion time prediction in a semiconductor fabrication plant, a fuzzy-neural approach with BPN post-classification is proposed in this study, in which a job is post-classified with some BPNs instead after predicting its completion time with a FBPN. In this novel way, only jobs which estimated completion times are the same accurate will be clustered into the same category. For evaluating the effectiveness of the proposed methodology and to make comparisons with some existing approaches, PS is applied in this study to generate test data. According to experimental results, the prediction accuracy (measured with RMSE) of the proposed methodology was significantly better than those of the other approaches in most cases by achieving a 20%~52% (and an average of 41%) reduction in RMSE over the comparison basis – MFLC. The effect of post-classification was also obvious. Nevertheless, a more sophisticated post-classification mechanism has to be developed to further enhance the performance.

However, to further evaluate the advantages and disadvantages of the proposed methodology, it has to be applied to plant models of different scales, especially a full-scale actual semiconductor fabrication plant. In addition, the proposed methodology can also be applied to cases with changing product mixes or loosely controlled priority combinations, under which the cycle time variation is often very large. These constitute some directions for future research.

References

1. Barman, S.: The Impact of Priority Rule Combinations on Lateness and Tardiness. IIE Transactions 30 (1998) 495-504
2. Chang, P.-C., Hsieh, J.-C.: A Neural Networks Approach for Due-date Assignment in a Wafer Fabrication Factory. International Journal of Industrial Engineering 10(1) (2003) 55-61
3. Chang, P.-C., Hsieh, J.-C., Liao, T. W.: A Case-based Reasoning Approach for Due Date Assignment in a Wafer Fabrication Factory. In: Proceedings of the International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, British Columbia, Canada (2001)
4. Chang, P.-C., Hsieh, J.-C., Liao, T. W.: Evolving Fuzzy Rules for Due-date Assignment Problem in Semiconductor Manufacturing Factory. Journal of Intelligent Manufacturing 16 (2005) 549-557
5. Chen, T.: A Fuzzy Back Propagation Network for Output Time Prediction in a Wafer Fab. Applied Soft Computing 2/3F (2003) 211-222
6. Chen, T.: A Fuzzy Set Approach for Evaluating the Achievability of an Output Time Forecast in a Wafer Fabrication Plant. Lecture Notes in Artificial Intelligence 4293 (2006) 483-493
7. Chen, T.: A Hybrid Look-ahead SOM-FBPN and FIR System for Wafer-lot-output Time Prediction and Achievability Evaluation. International Journal of Advanced Manufacturing Technology (2007) DOI 10.1007/s00170-006-0741-x
8. Chen, T.: A Hybrid SOM-BPN Approach to Lot Output Time Prediction in a Wafer Fab. Neural Processing Letters 24(3) (2006) 271-288
9. Chen, T.: A Look-ahead Fuzzy Back Propagation Network for Lot Output Time Series Prediction in a Wafer Fab. Lecture Notes in Computer Science 4234 (2006) 974-982
10. Chen, T.: Applying an Intelligent Neural System to Predicting Lot Output Time in a Semiconductor Fabrication Factory. Lecture Notes in Computer Science 4234 (2006) 581-588
11. Chen, T., Jeang, A., Wang, Y.-C.: A Hybrid Neural Network and Selective Allowance Approach for Internal Due Date Assignment in a Wafer Fabrication Plant. International Journal of Advanced Manufacturing Technology (2007) DOI 10.1007/s00170-006-0869-8
12. Chen, T., Lin, Y.-C.: A Hybrid and Intelligent System for Predicting Lot Output Time in a Semiconductor Fabrication Factory. Lecture Notes in Artificial Intelligence 4259 (2006) 757-766

13. Chung, S.-H., Yang, M.-H., Cheng, C.-M.: The Design of Due Date Assignment Model and the Determination of Flow Time Control Parameters for the Wafer Fabrication Factories. IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part C 20(4) (1997) 278-287
14. Foster, W. R., Gollopy, F., Ungar, L. H.: Neural Network Forecasting of Short, Noisy Time Series. Computers in Chemical Engineering 16(4) (1992) 293-297
15. Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
16. Hung, Y.-F., Chang, C.-B.: Dispatching Rules Using Flow Time Predictions for Semiconductor Wafer Fabrications. In: Proceedings of the 5th Annual International Conference on Industrial Engineering Theory, Applications and Practice, Taiwan (2001)
17. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed Representation of Fuzzy Rules and Its Application to Pattern Classification. Fuzzy Sets and Systems 52(1) (1992) 21-32
18. Lin, C.-Y.: Shop Floor Scheduling of Semiconductor Wafer Fabrication Using Real-time Feedback Control and Prediction. Ph.D. Dissertation, Engineering-Industrial Engineering and Operations Research, University of California at Berkeley (1996)
19. Piramuthu, S.: Theory and Methodology – Financial Credit-risk Evaluation with Neural and Neuralfuzzy Systems. European Journal of Operational Research 112 (1991) 310-321
20. Ragatz, G. L., Mabert, V. A.: A Simulation Analysis of Due Date Assignment. Journal of Operations Management 5 (1984) 27-39
21. Vig, M. M., Dooley, K. J.: Dynamic Rules for Due-date Assignment. International Journal of Production Research 29(7) (1991) 1361-1377
22. Wang, L.-X., Mendel, J. M.: Generating Fuzzy Rules by Learning from Examples. IEEE Transactions on Systems, Man, and Cybernetics 22(6) (1992) 1414-1427
23. Weeks, J. K.: A Simulation Study of Predictable Due-dates. Management Science 25 (1979) 363–373

Enhanced Genetic Algorithm-Based Fuzzy Multiobjective Strategy to Multiproduct Batch Plant Design

Alberto A. Aguilar-Lasserre[1], Catherine Azzaro-Pantel[2], Luc Pibouleau[2], and Serge Domenech[2]

[1] Division of Research and Postgraduate Studies, Instituto Tecnologico de Orizaba, Av. Instituto Tecnologico 852, Col Emiliano Zapata. 09340 Orizaba, Veracruz, México
[2] Laboratoire de Génie Chimique-UMR 5503 CNRS/INP/UPS 5, Rue Paulin Talabot- BP1301, 534-61-5252
aaguilar@itorizaba.edu.mx,
{Catherine.AzzaroPantel, Luc.Pibouleau, Serge.Domenech}@ensiacet.fr

Abstract. The design of such plants necessary involves how equipment may be utilized, which means that plant scheduling and production must form an integral part of the design problem. This work proposes an alternative treatment of the imprecision (demands) by using fuzzy concepts. In this study, we introduce a new approach to the design problem based on a multi-objective genetic algorithm, taking into account simultaneously maximization of the net present value $\widetilde{NPV}$ and two other performance criteria, i.e. the production delay/advance and a flexibility criterion. The methodology provides a set of scenarios that are helpful to the decision's maker and constitutes a very promising framework for taken imprecision into account in new product development stage. Besides, a hybrid selection method Pareto rank-tournament was proposed and showed a better performance than the classical Goldberg's wheel, systematically leading to a higher number of non-dominated solutions.

Keywords: Multiobjective Optimization, Genetic Algorithm, Fuzzy Arithmetic, Batch Plant Design.

1 Introduction

In recent years, there has been an increased interest in the design of batch plant due to the growth of specialty chemical, biochemical, pharmaceutical and food industries. The most common form of batch plant design formulation considered in the literature is a deterministic one, in which fixed production requirements of each product must be fulfilled. However, it is often the case that no precise product demand predictions are available at the design stage (Shah and Pantelides, 1992).

The market demand for such products is usually changeable, and at the stage of conceptual design of a batch plant, it is almost impossible to obtain the precise information on the future product demand over the lifetime of the plant. Nevertheless, decisions must be taken on the plant capacity. This capacity should be able to balance

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 590–599, 2007.
springerlink.com

the product demand satisfaction and extra plant capacity in order to reduce the loss on the excessive investment cost or than on market share due to the varying product demands (Cao and Yuan 2002).

The key point in the optimal design of batch plants under imprecision concerns modeling of demand variations. The most common approaches treated in the dedicated literature represent the demand uncertainty with a probabilistic frame by means of Gaussian distributions. Yet, this assumption does not seem to be a reliable representation of the reality, since in practice the parameters are interdependent, leading to very hard computations of conditional probabilities, and do not follow symmetric distribution rules.

In this work, fuzzy concepts and arithmetic constitute an alternative to describe the imprecise nature on product demands. For this purpose, we extended a multiobjective genetic algorithm, developed in previous works (Aguilar et al. 2005), taking into account simultaneously the maximization of the net present value $N\tilde{P}V$ and two other performance criteria, i.e. the production delay/advance and a flexibility criterion. The paper is organized as follows. Section 2 is devoted to process description and problem formulation. Section 3 presents a brief overview of fuzzy set theory involved in the fuzzy framework within a multiobjective genetic algorithm. The presentation is then illustrated by some typical results in Section 4.

2 Process Description and Problem Formulation

2.1 Problem Statement

In conventional optimal design of a multiproduct batch chemical plant, a designer specifies the production requirements for each product and total production time for all products. The number, required volume and size of parallel equipment units in each stage are to be determined in order to minimize the investment cost (Huang and Wang 2002).

The designers must not only satisfy technico-economic criteria, but also respect some due dates. In this framework, this study introduces a new design approach to maximize the net present value $N\tilde{P}V$ and two other performance criteria, i.e. the production delay/advance and a flexibility criterion. Such an optimal design problem becomes a multi-objective optimization problem (MOOP).

In order to specify the production requirements for each product and total production time for all products, it is almost impossible to obtain some precise information. Indeed, the ability of batch plants to deal with irregular product demand patterns reflecting market uncertainties or seasonal variations is one of the main reasons for the recently renewed interest in batch operations. In this paper, we consider an alternative treatment of the imprecision of the demand by using fuzzy concepts. A genetic algorithm was implemented for solving this problem, since it has demonstrated to be efficient in multi-objective optimization problems.

2.2 Assumptions

The model formulation for batch plant design problems adopted in this paper is based on Modi's approach (Modi and Karimi 1989). It considers not only treatment in batch stages, which usually appears in all types of formulation, but also represents semi-continuous units that are part of the whole process (pumps, heat exchangers,...). Besides, this formulation takes into account mid-term intermediate storage tanks. So, a batch plant is finally represented by series of batch stages (B), semi-continuous stages (SC) and storage tanks (ST). The model is based on the following assumptions:

1. The devices used in a same production line cannot be used twice by one same batch.
2. The production is achieved through a series of single product campaigns.
3. The units of the same batch or semi-continuous stage have the same type and size.
4. All intermediate tank sizes are finite.
5. If a storage tank exists between two stages, the operation mode is "Finite Intermediate Storage". If not, the "Zero-Wait" policy is adopted.
6. There is no limitation for utility.
7. The cleaning time of the batch items is included into the processing time.
8. The item sizes are continuous bounded variables.

2.3 Model Formulation

The model considers the synthesis of *I* products treated in *J* batch stages and *K* semi-continuous stages. Each batch stage consists of *mj* out-of-phase parallel items with same size *Vj*. Each semi-continuous stage consists of *nk* out-of-phase parallel items with same processing rate *Rk* (i.e. treatment capacity, measured in volume unit per time unit). The item sizes (continuous variables) and equipment numbers per stage (discrete variables) are bounded. The *S-1* storage tanks, with size *Vs**, divide the whole process into *S* sub-processes.

a) Economic criterion evaluation: The net present value method ($N\tilde{P}V$) of evaluating a major project allows to consider the time value of money (1). Essentially, it helps find the present value in "today's value money" of the future net cash flow of a project. Then, this amount can be compared with the amount of money needed to implement the project. When using this formula, the values of the number of periods (*n*), *d*iscount rate (*r*) and tax rate (*a*) take respectively the following classical values 5, 10% and 0 (computation before tax). In order to calculate investment cost *(Cost)* (2), the working capital *(f)*, revenue ($\tilde{V}_P$), operation cost ($\tilde{D}_P$) and depreciation *(A_P)* are introduced.

$$Max\left(N\tilde{P}V\right) = -Cost - f + \sum_{p=1}^{n} \frac{(\tilde{V}_p - \tilde{D}_p - A_p)(1-a) + A_p}{(1+r)^n} \tag{1}$$

$$Cost = \sum_{j=1}^{J} (m_j a_j V_j^{\alpha_j}) + \sum_{k=1}^{K} (n_k b_k R_k^{\beta_k}) + \sum_{s=1}^{S} (c_s V_s^{\gamma_s}) \tag{2}$$

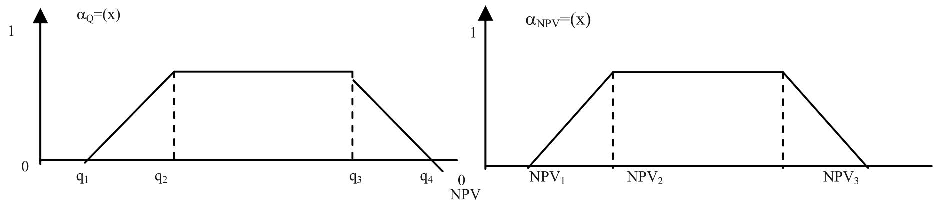


Fig. 1. Two trapezoidal fuzzy numbers, $\tilde{Q} = (q_1,q_2,q_3,q_4)$, $N\tilde{P}V$ =(NPV_1,NPV_2,NPV_3,NPV_4)

The proposed approach involves arithmetic operations on fuzzy numbers and quantifies the imprecision of the demand by means of fuzzy sets (trapezoidal). In this case, the flat line over the interval (q_2,q_3) represents the precise demands with an interval of confidence at level α=1, while the intervals (q_1,q_2) and (q_3,q_4) represent the "more or less possible values" of the demand. The result of the *net present value* ($N\tilde{P}V$) is treated and analyzed through fuzzy numbers. The demand and the net present value are fuzzy quantities as shown in figure 1.

b) Computation of the criterion penalizing the delays and advances of the production time necessary for the synthesis of all the products: for this purpose, we must compare the time horizon $\tilde{H}$ represented by a fuzzy expression (rectangle) and the production time $\tilde{H}_i$ (trapezoidal). For the comparison of fuzzy numbers, Liou and Wang's method (Liou and Wang 1992) was adopted.

The production time necessary to satisfy each product demand must be less than a given time horizon, but due to the nature of the fuzzy numbers, eight different cases for determination of the second criterion may occur. The different cases are reported in figure 2.

The temporal criterion selected is called "common surface", representing the intersection between the sum of the production time (trapezoid) and the horizon of time to respect (rectangle). The calculation of the criterion depends on each case: for example, case1 illustrate the solutions which arrive just in time.

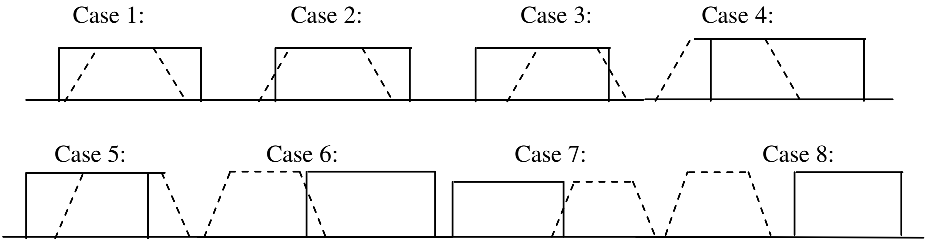


Fig. 2. Eight cases for the minimization of a criterion that penalizes the delays and advances of the time horizon necessary for the synthesis of all the products

The criterion relative to the advances (2, 4, 6 and 8) or to the delays (3, 5 and 7) is calculated by the formulas 3 and 4 respectively. The corresponding mathematical expressions of the objective functions are proposed as follows:

$$Max\,(Criterion\ of\ advance) = Common\ surface\ \ x\ \ \omega \tag{3}$$

$$Max\,(Criterion\ of\ delay) = Common\ surface\ \ x\ \ 1/\,\omega \tag{4}$$

The penalization term is equal to an arbitrary value of ω for an advance and $1/\omega$ for a delay in order to penalize more delays than advances. A sensitivity analysis leads to adopt a value of 3 for ω.

c) Flexibility index: Finally, an additional criterion was computed in case of an advance (respectively a delay), representing the additional production (the demand not produced) that the batch plant is able to produce. Without going further in the detailed presentation of the computation procedure, it can be simply said that a flexibility index is computed by dividing the potential capacity of the plant by its actual value.

The problem statement involves four forms of different constraints as reported in literature (Modi and Karimi 1989): Dimension constraints, time constraint, constraint on productivities and the size of intermediate storage tanks.

3 A Fuzzy Decision-Making Approach for Multiproduct Batch Plant Design

3.1 Overview of Multiobjetive Genetic Algorithm Approach

The GA implemented in this study uses quite common genetic operators. The proposed GA procedure implies the following steps:

1) ***Encoding of solution.*** The encoding of the solutions was carried out dividing the chromosome, i.e. a complete set of coded variables, into two parts. The first one deals with the items volumes, which are continuous in the initial formulation. Nevertheless, they were discretized here with a 50 unit range, while their upper and lower bounds were preserved. The second part of the chromosome handles the number of equipment items per stage: the value of these discrete variables is coded directly in the chromosome.

2) ***Initial population creation.*** The procedure of creating the initial population corresponds to random sampling of each decision variable within its specific range of variation. This strategy guarantees a population various enough to explore large zones of the search space.

3) ***Fitness Evaluation.*** The optimization criterion considered for fitness evaluation involves the *net present value* (NPV) and two other performance criteria, i.e. the production delay/advance and a flexibility criterion. Traditionally, a GA uses a fitness function, which must be maximized. The fitness for these criteria is equal to their directly calculated values.

4) ***Selection Procedure.*** The multi-objective aspects are taken into account in the selection procedure. A hybrid selection method Pareto rank-tournament was proposed and showed a better performance than the classical Goldberg's wheel, systematically leading to a higher number of non-dominated solutions.

The method of tournament preferentially selects the non-dominated individuals of case 1, thus successively and in a consecutive way the procedure selects then if need be cases 2, 3, 4, 5, 6 and 7 (preference for the solutions with greatest common surfaces between the sum of the horizons of time "trapezoid" and the horizon of time to respect "rectangle") until arriving at case 8 (without common surface).

The procedure of selection of the hybrid selection method Pareto rank-tournament into account a Pareto set following the criteria of Pareto dominance are then implemented on the population of individuals and makes it possible to extract the set of the non-dominated Pareto's optimal solutions.

5) ***Crossover.*** Two selected parents are submitted to the crossover operator to produce two children. The crossover is carried out with an assigned probability, which is generally rather high. If a randomly generated number is superior to the probability, the crossover is performed. Otherwise, the children are copies of the parents.

6) ***Mutation.*** The genetic mutation introduces diversity in the population by an occasional random replacement of the individuals. The mutation is performed on the basis of an assigned probability. A random number is used to determine if a new individual will be produced to substitute the one generated by crossover. The mutation procedure consists of replacing one of the decision variable values of an individual, while keeping the remaining variables unchanged. The replaced variable is randomly chosen, and its new value is calculated by randomly sampling within its specific range.

7) ***Registration of all non-dominated individuals in Pareto set.*** Pareto's sort procedure is carried out at the end of the algorithm over all the evaluated solutions; at the end of the procedure, the whole set of the non dominated Pareto's optimal solutions, are obtained.

3.2 Treatment of an Illustrative Example

We consider an example to illustrate the approach fuzzy-AG based on arithmetic operations on fuzzy numbers and quantifying the imprecision of the demand. The example was initially presented by Ponsich and al. (2004): the plant, divided into two sub-processes, consists of six batch stages to manufacture three products.

The GA parameters are the following ones: Population size 200 individuals, number of generations 400 iterations, crossover probability 40%, mutation probability 30% and the stop criterion considered in this study concerns a maximum number of generations to reach.

For the considered example, table 1 shows the values for processing times, size factor for the units, cost data, and the production requirement for each product quantifying the imprecision of the demand by means of fuzzy numbers representing the "more or less possible values".

For the construction of the trapezoid which represents the request for each product, the original values of the demand were used as a reference. To determine the support and the core, one calculated a percentage of opening taking as reference the demand of the original data is computed.

Table 1. Data used in example

		Processing time $\tau_{i,j}$ (h)						***Size factors (1/kg)***					
		B1	*B2*	*B3*	*B4*	*B5*	*B6*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*
Minimum size =250 l	**A**	1.15	3.98	9.86	5.28	1.2	3.57	8.28	6.92	9.7	2.95	6.57	10.6
Maximum size = 10 000 l	**B**	5.95	7.52	7.01	7	1.08	5.78	5.58	8.03	8.09	3.27	6.17	6.57
	C	3.96	5.07	6.01	5.13	0.66	4.37	2.34	9.19	10.3	5.7	5.98	3.14
	γ_i	0.4	0.29	0.33	0.3	0.2	0.35						

Unit price for product i ($/Kg)			***Coefficients $c_{i,j}$***						Q$_1$=(419520. 428260, 441370, 445740) Q$_2$=(311040, 319140, 330480, 336960) Q$_3$=(247680, 258000, 263160,268320) H = (5760, 5760, 6240, 6240)
	C_P	C_O	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	Cost of mixer=$\$250V^{0.6}$
A	0.70	0.08	0.2	0.36	0.24	0.4	0.5	0.4	Cost of reactor=$\$250V^{0.6}$
B	0.74	0.1	0.15	0.5	0.35	0.7	0.42	0.38	Cost of extractor=$\$250V^{0.6}$
C	0.80	0.07	0.34	0.64	0.5	0.85	0.3	0.22	Cost of centrifuge=$\$250V^{0.6}$
									(Volume V in liter)

	Operating cost factors					
	B1	*B2*	*B3*	*B4*	*B5*	*B6*
C_E	20	30	15	35	37	18

4 Typical Results

4.1 Monocriterion Case

For the monocriterion case, the best individuals, that are the surviving ones, are chosen with the Goldberg's roulette. Each individual is represented by a slice of the roulette wheel, proportional to its fitness value. Since the criteria are represented by fuzzy numbers, they were defuzzified (the defuzzified value was calculated with the Liou and Wang's method) in the roulette wheel.

GA typical results obtained with $\tilde{NPV}$ as the only criterion to consider are presented in Table 2 (ten runs were performed to guarantee the stochastic nature of the GA). In particular, the value of the best individual of each generation and the average value of the function objective of the population take a traditional form of regular increase, to stabilize itself finally at the end of the research.

4.2 Multi-objective Optimization

The multi-objective resolution strategy for multiproduct batch plant design uses two genetic algorithms (bicriteria and tricriteria optimizations are considered).

4.2.1 Bicriteria Case: NPV- The Delays and Advances of the Time Horizon

The first bicriteria analysis takes into account the NPV and the criterion which represents the advances or delays of the time horizon with the hybrid selection method Pareto rank-tournament.

Table 2. Monocriterion ($\tilde{NPV}$): Fuzzy optimal design of batch plant

Product	B_{is} kg	T_{Li} h	*Criterion: The net present value ($)*	*Information Complémentaire*
A	943.3	4.8	$\tilde{N}PV$ =[722225.3 803765.7 892941.6 969060.6]	$\tilde{V}_p$=[721977.6 742345.6 764042.2 782142.4] [$] $\tilde{D}_p$= [232095.9 238513.7 245540.9 251437.2] [$] I = 698877.8 [$] A_p= 139775.5 [$] f= 104831.6 [$] V_s = 1505.2 [l] $\sum \tilde{H}_i$ = [5759.9 5925.4 6097.3 6239.9] [h]
B	1145.5	7.3		
C	899.7	6.6		

To treat the bicriteria optimization, the analysis is performed on the solution of case 2 with the best NPV and on those for case 3 having, on the one hand, larger common surfaces and, on the other hand, the best NPV. The example did not provide any solution of case 1, because the rectangle which represents the horizon of time to respect is larger than the trapezoids obtained for the sum of times of production. Table 3 shows the results of the chosen solutions.

Table 3. Fuzzy optimal design of batch plant for case 2 (advance) and cases 3 (delays)

	NPV (Mean Value)	Common surface	$\tilde{NPV}$ ($) and production time $\Sigma \tilde{H}$ (h)
Cas 2	860358	653	$\tilde{NPV}$ = [736120, 817367, 906144, 981801] $\Sigma \tilde{H}$ = [5758, 5916, 6089, 6238]
Cas 3a	860550	163	$\tilde{NPV}$ = [736327, 817565, 906332, 981976] $\Sigma \tilde{H}$ = [5772, 5930, 6104, 6253]
Cas 3b	861823	129	$\tilde{NPV}$ = [737489, 818728, 907495, 983139] $\Sigma \tilde{H}$ = [5883, 6045, 6222, 6374]

4.2.2 Bicriteria Case: NPV- Flexibility Index

The second bicriteria analysis takes into account the NPV and the criterion which represents the flexibility index of the configuration chosen to produce a possible additional demand. Figure 3 exhibits 277 non dominated solutions of the advance cases "(2, 4, 6 and 8) and of the delay cases (3, 5 and 7).

Two solutions of case 2: the first is the solution with the best benefit and the second configuration has the best index of flexibility. The third corresponds to case 3 with the best index of flexibility.

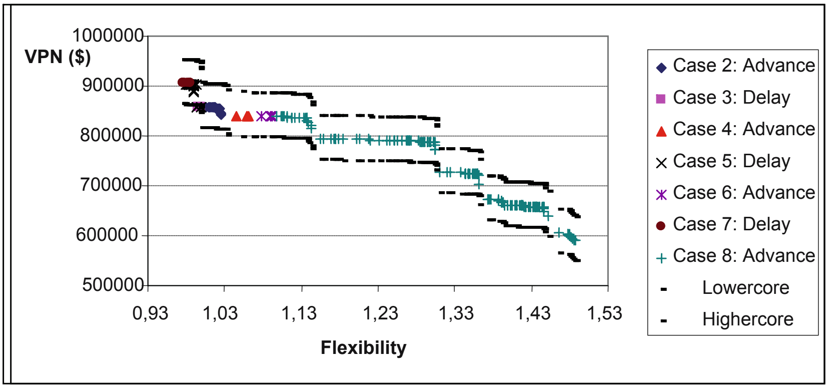


Fig. 3. Bicriteria results: NPV- Flexibility Index

4.2.3 Comparison Between a Hybrid Selection Method (Pareto Rank-Tournament) and the Classical Goldberg's Wheel for the Bicriteria Optimization Case

To evaluate the performance of the selection method of the fuzzy genetic algorithm (combination of Pareto set and tournament), it is proposed to test the algorithm modified with the classical Goldberg's wheel. The same number of surviving individuals is chosen for each criterion (two roulettes). The same parameters for AG with the procedure of Pareto rank-tournament, are used with Goldberg's wheel.

A higher number of non dominated individuals are obtained with the hybrid method because the selection provides compromise solutions between the two criteria and the values of their criteria are better than those obtained with the Goldberg's wheel.

4.2.4 Tricriteria Case: NPV- Delays/Advances- Flexibility Index

Lastly, the fuzzy optimal design of batch plant takes into account simultaneously the three criteria, i.e., NPV, criteria of the advances or delays (common surface) and index of flexibility. The method proposes a sufficiently large range of compromise solutions making it possible to the decision's maker to tackle the problem of the final choice, with relevant information for his final choice.

To analyse the results obtained with the tricriteria case, 6 non dominated solutions are adopted: 3 of case 2, 2 of case 4 and 1 of case 6. These solutions were selected by considering the same policy as for the bicriteria case.

5 Conclusions

For the most common form of design of a multiproduct batch chemical plant, the designers specify the production requirement of each product and the total production time. However, no precise product demand predictions are generally available. For this reason, an alternative treatment of the imprecision by using fuzzy concepts is introduced in this paper.

In this study, we have introduced a fuzzy-AGs approach to solve the problem of multi-objective optimal design of a multiproduct batch chemical plant. The results obtained on the treated example have shown that three different scenarios were obtained as a fuzzy decision-making approach. The analysis tended to be helpful for decision making.

Its benefits can be summarized as follows:

- Fuzzy concepts allow us to model imprecision in cases where historical data are not readily available, i.e. for demand representation;
- The models do not suffer from the combinatorial explosion of scenarios that discrete probabilistic uncertainty representation exhibit;
- Another significant advantage is that heuristic search algorithms, namely genetic algorithms for combinatorial optimization can be easily extended to the fuzzy case;
- The hybrid selection method Pareto rank-tournament was proposed and showed a better performance than the classical Goldberg's wheel, systematically leading to a higher number of non-dominated solutions.
- Multiobjective concepts can also be taken into account.
- The proposed approach thus constitutes an efficient and robust support to assist the mission of the designer, leading to a quite large set of compromise solutions.

Finally, this framework provides an interesting decision-making approach to design multiproduct batch plants under conflicting goals.

References

1. Aguilar and al. (2005) Modélisation des imprécisions de la demande en conception optimale multicritère d'ateliers discontinus, *Proceedings of the* SFGP (*Société Français de Génie de Procédés*). Toulouse, France
2. Cao D. and Yuan X. (2002) Optimal design of batch plants with uncertain demands considering switch over of operating modes of parallel units, *Ind. Eng. Chem.*, pp 4616-4625
3. Huang H.,Wang W. F. (2002) Fuzzy decision-making design of chemical plant using mixed-integer hybrid differential evolution, *Computers & Chemical Engineering*, Volume 26, Issue 12, 15, pp 1649-1660
4. Liou T.S. and Wang M.J (1992) Ranking fuzzy numbers with integral value, Fuzzy Sets System, vol. 50, pp 247
5. Modi A. K. and Karimi I.A. (1989) Design of multiproduct batch processes with finite intermediate storage, Computer and Chemical Engineering, 13, pp 127-138
6. Shah N. and Pantelides C.C. (1992) Design of multipurpose batch plants with uncertain production requirements, *Ind. Eng. Chem. Res.*, pp 1325-1337

Extension or Resolution: A Novel Approach for Reasoning in Possibilistic Logic

Yin Minghao and Jigui Sun

College of Computer Science & Technology, Jilin University, Changchun, China, 130017

Abstract. Resolution principle is the rule of inference at the basis of most procedures for automated reasoning both in classic logic and possibilistic logic. In these procedures, deduction starts by inferring new clauses by resolution, and goes on until the empty clause is generated or consistency of the set of clauses is proven, e.g. because no new clauses can be generated. Theorem proving using extension rule is a novel approach for reasoning in classic logic advanced by us recently. The key idea is to use the inverse of resolution and to use the inclusion-exclusion principle to circumvent the problem of space complexity. In this paper, we proposed a possibilistic extension of "extension rule", which we called possibilistic extension rule. We showed that it outperformed possibilistic resolution-based method in some cases. And it is potentially a complementary method to possibilistic resolution rule.

1 Introduction

Resolution principle is the rule of inference at the basis of most procedures for automated reasoning. In these procedures, the input formula is first translated in an equivalent set of clauses. Then deduction starts by inferring new clauses by resolution, and goes on until the empty clause is generated or consistency of the set of clauses is proven, e.g. because no new clauses can be generated. Based on classic resolution principle, D. Dubois and H. Prade presented the possibilistic extension of resolution principle [1, 2] for reasoning in possibilistic logic [3]. Since its introduction, reasoning based on possibilistic resolution principle has been widely studied. The aim of this paper is to challenge possibilistic resolution principle by using the inverse of possibilistic resolution. Thus, we try to compute the inconsistency degree of a given possibilistic knowledge base by deduction of the set of all the possibilistic maximum terms. This work is motivated by [4]. In [4], we present a novel method for theorem proving in classic logic, we called extension rule. We can compare possibilistic resolution principle and possibilistic extension rule here first. If we want to compute a given set of possibilistic clauses' inconsistent degree, the empty clause is deduced for one of its classic cut set. While in possibilistic extension-based method, one of its cut set needs to consist of all the possibilistic maximum terms.

This paper explores what happens if we use the inverse of possibilistic resolution. Our contributions are:

(1) We use the inverse of possibilistic resolution together with the inclusion-exclusion principle to reason in possibilistic logic – this has never been done before.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 600–607, 2007.
springerlink.com

This may be considered a new framework for reasoning in possibilistic logic that is worth exploring.

(2) What makes our method promising is that it leads to a new method for knowledge compilation in possibilistic logic, which is advanced by H. Prade etc. recently [6].

We assume that the readers are familiar with possibilistic logic and we won't discuss it in length in this paper. For more detail, we refer to [3].

2 Possibilistic Extension Rules

In this paper, we focus on possibilistic extension rules in the standard version of possibilistic logic, which handles possibilistic formula corresponding to certainty-qualified statements. In the rest of the paper, well-formed possibilistic formula will be denoted by (φ,α), where φ is a classical logic formula and α is the lower bound certainty valuation of the formula. We use $\sum$ to denote a possibilistic knowledge base in clause form, (c,α) to denote a possibilistic clause, M to denote the set of all atoms appearing inΣ, and T, $\perp$ denote tautology and contradiction respectively.

Definition 1 (possibilistic reduction rule, SPL-R)**.** Given two sets of possibilistic clauses, where C = {(c,α), (c,β)|$\alpha\geq\beta$}, C'= {(c,α)}, we call the operation proceeding from C to C' the *possibilistic reduction rule* on C. We call C' the result of the possibilistic reduction rule.

Obviously, C is logically equivalent to C', because (c,α) $\wedge$ (c,β) $\models$ (c,α); (c,α)$\models$(c,β).

Definition 2 (possibilistic extension rule, SPL-ER)**.** Given a single possibilistic clause (c,α) and a set M:

C'={ $(c\vee b,\alpha)$, $(c\vee\neg b,\alpha)$ | "b" is an atom, b$\in$M and "b" does not appear in c}

We call the operation proceeding from (c,α) to C' the *possibilistic extension rule* on (c,α). We call C' the result of the possibilistic extension rule. For example, given a possibilistic clause ($p\vee q$, 1) and the set of atoms {p, q, r}, the result of (SPL-ER) is {($p\vee q\vee r$, 1), ($p\vee q\vee\neg r$, 1)}. Notice that the extension rule is just the inverse of resolution.

Theorem 1. Given an arbitrary SPL clause (c,α), (c,α) is logically equivalent to its result of the rule (SPL-ER).

Proof. Suppose C' = { $(c\vee b,\alpha)$, $(c\vee\neg b,\alpha)$ } is the result of rule (SPL-ER) on (c,α), we can deduce (c,α) from C' by one step of possibilistic resolution. On the other hand, since N($c\vee b$) $\geq$ max($N(c)\vee N(b)$) = max($\alpha\vee N(b)$) $\geq\alpha$ where N is necessity measure induced by a possibilistic distribution satisfying (c,α), so (c,α) $\models (c \vee b,\alpha)$. Similarly, $(c,\alpha) \models (c \vee \neg b,\alpha)$. Thus, we can deduce C' from (c,α).

3 Reasoning Using Possibilistic Extension Rules

In this section, we will show how to use (SPL-ER) and (SPL-R) for reasoning in possibilistic logic. First we will define *possibilistic maximum term (PMT).*

Definition 3. A possibilistic clause is a *possibilistic maximum term (PMT)* on a set M iff its classical projection contains all atoms in M in either positive form or negative form.

The following theorem is used to tell how to compute the inconsistent degree of $\sum$. Let _*INC*($\sum$) denote the inconsistent degree of F.

Theorem 2. Given a possibilistic knowledge base $\sum$ in clause form, with its set of atoms M (|M|=m). If all the possibilistic clauses in $\sum$ are all PMT on M, then _*INC*($\sum$)>0 iff its classical projection contains 2^m clauses.

Let Σ' be the result of rule (SPL-R) from $\sum$ where no rule (SPL-R) can be applied any more and $\sum$' contains 2^m clauses, then the inconsistent degree of Σ can be computed as: _*INC*($\sum$) = inf (α_i | (c_i, α_i)∈$\sum$').

Proof. According to theorem 3.1 in [7], the classical projection of $\sum$ is UNSAT iff the number of clauses of it equals 2^m. Thus _*INC*($\sum$)> 0 iff its classical projection contains 2^m clauses. According to [1], given a possibilistic knowledge base $\sum$, _*INC*($\sum$)= $\max\limits_{\Sigma' \subseteq \Sigma, \_INC(\Sigma')>0} \min\{\alpha \mid (\Phi,\alpha) \in \Sigma'\}$. Since removing any possibilistic clause in $\sum$' leads that the new set of possibilistic clauses to be consistent, _*INC*($\sum$) = inf (α_i | (c_i, α_i)∈$\sum$').

According to theorem 1, if we want to compute the inconsistent degree of a set of possibilistic clauses, we can proceed by finding an equivalent set of possibilistic clauses such that all the clauses in it are PMT by using the rule (SPL-ER), and then prune it using the rule (SPL-R). And then, using Theorem 1 we can compute its inconsistent degree. We call this process *reasoning based on the Extension rule in possibilistic logic.*

In fact, reduced to classical prepositional logic, the theorem is: "if each clauses in the CNF contains all variables (so called canonical implicates in [1]), then the problem is UNSAT if the number of clauses is 2^m", which is really straightforward. Here we show an example to make things clear.

Example 1. Given a set of possibilistic clauses C = {($p \vee q$, 1), ($\neg p \vee q$, 0.8), ($\neg r \vee \neg q$,0.4), (r 0.6), ($\neg r$,0.3)}, we apply rule (SPL-ER) and rule (SPL-R) to C. We can get a new SPL clauses set C'= {($p \vee q \vee r$, 1), ($p \vee q \vee \neg r$, 1), ($\neg p \vee q \vee r$, 0.8), ($\neg p \vee q \vee \neg r$, 0.8), ($p \vee \neg r \vee \neg q$, 0.4), ($\neg p \vee \neg r \vee \neg q$, 0.4), ($p \vee \neg q \vee r$, 0.6), ($\neg p \vee \neg q \vee r$, 0.6)}. Since the number of PMTs equals 2^m, C is partially inconsistent, and its inconsistent degree is 0.4.

The above process is rather inefficiency and is hardly helpful in practice, because for each problem it needs exponential time and space. So it's clear that a direct use of (SPL-ER) (generate all the PMT and prune it using SPL-R) is infeasible. However, think twice the problem, we can find that we needn't list (know) all the clauses but just count the clauses. If we know the total number of a given cut subset of $\sum$ is 2^m, we can know that this subset's is inconsistent; otherwise this subset is consistent. In fact we could use the famous "inclusion-exclusion principle" to compute the inconsistent degree. But we need to introduce this theorem first.

Theorem 4. Given any two possibilistic clauses (c_1,α) and (c_2, β), C_1 and C_2 are the set of possibilistic maximum terms extended by (c_1,α) and (c_2, β) respectively. For arbitrary clause (c_1',α)$\in C_1$, and arbitrary clause (c_2', β)$\in C_2$, we can't apply rule (SPL-R) on (c_1',α) and (c_2', β) iff there are complementary literals in (c_1,α) and (c_2, β).

Proof. (Sufficiency) If there are complementary literals in (c_1,α) and (c_2, β), say A and ¬A, then for each possibilistic clause in C_1, it contains A, and for each clauses in C_2, it contains ¬A. So, for each clause in C_1, and for each clause in C_2, their classic projection cannot be the same. According to definition 1, we can't apply rule (SPL-R).

(Necessary) Assume that there are no complementary literals in (c_1,α) and (c_2, β) and the set of literals appearing in (c_1,α) are L = {l_1, …, l_r}, the set of literals appearing in (c_2, β) are L' = {l'_1, …, l'_s}. Without losing generality, let L" = L $\cup$ L' = {l''_1, …, l''_t }. Obviously, ($l''_1 \vee \ldots \vee l''_t$, α) can be extended by (c_1,α), and ($l''_1 \vee \ldots \vee l''_t$, β) can be extended by (c_2, β). So we can apply rule (SPL-R) on them to prune one possibilistic clause. So, if we cannot apply rule (SPL-R) on C_1 and C_2, there are complementary literals in (c_1,α) and (c_2, β).

Example 2. Consider the clauses ($p \vee q$, 1) and ($\neg p \vee q$, 0.8). For each clause extended by ($p \vee q$, 1), it must contain p, and for each clause extended by ($\neg p \vee q$, 0.8), it must contain ¬p. So we can't apply rule (SPL-R) on their results. Consider the clauses ($p \vee q$, 1) and (r 0.6). By applying rule (SPL-ER) on the first clause, we have ($p \vee q \vee r$, 1); by applying rule (SPL-R), we have ($p \vee q \vee r$, 0.6). So we can apply rule (SPL-R) on them to prune ($p \vee q \vee r$, 0.6).

Lemma 5. Given two possibilistic clauses (c_1,α) = ($l_1 \vee \ldots \vee l_r$, α) and (c_2, β) = ($l'_1 \vee \ldots \vee l'_s$, β), and there are no complementary literals in (c_1,α) and (c_2, β). Let M be set of atoms in $\sum$ and |M| = m. Let L = {l_1,…, l_r}$\cup${l'_1, …, l'_s} = {l''_1, …, l''_t}. Suppose C_1 is the set of possibilistic maximum terms extended by (c_1,α), C_2 is the set of possibilistic maximum terms extended by (c_2, β), C_3 is the set of clauses pruned by C_1 and C_2 by applying rule (SPL-R). Then

$$|C_1| = 2^{m-r};\ |C_2| = 2^{m-s};\ |C_3| = 2^{m-t}$$

We will use an example to illustrate lemma 6:

Example 3. Let C be the set of possibilistic clauses mentioned in example 1. Obviously, M = {p, q, r}. For the clause ($p \vee q$, 1), according to lemma 6, the number of PMTs extended by this clause is $2^{3-2} = 2$; for the clause (r, 0.6), the number of PMTs extended by this clause is 2^{3-1}. For the clauses ($p \vee q$, 1) and (r, 0.6), the number of clauses pruned by applying rule (SPL-R) is $2^{3-(2+1)}$.

Remark 1. Given two clauses in $\sum$, let P_1 denote the set of possibilistic maximum terms we can get from c_1, and P_2 denote the set of possibilistic maximum terms we can get from c_2. We change the definition of intersection slightly, i.e. $P_i \cap P_j = \{(c_i, \beta) | (c_i, \beta) \in P_i, (c_i, \alpha) \in P_j, \alpha \geq \beta\}$. Notice that reduced to classic set, the definition of intersection doesn't change. The semantics of "intersection" defined here is just to show which clause will be pruned by rule (SPL-R) after extended.

Now we show how to count possibilistic maximum terms here. Given a set of possibilistic clauses $\sum = \{c_1, c_2, \ldots, c_n\}$, let M be the set of atoms which appear in $\sum$ (|M|=m) and let $|c_i|$ be the set of atoms which appear in c_i . Let P_i be the set of all the possibilistic maximum terms we can get from c_i by using rule (SPL-ER), and let S be the number of distinct maximum terms we can get from $\sum$, where no (SPL-R) rule can be applied on S. By using the intension-exclusion rule, we will have:

$$S = \sum_{i=1}^{n} |P_i| - \sum_{1 \le i < j \le n} |P_i \cap P_j| + \sum_{1 \le i < j < l \le n} |P_i \cap P_j \cap P_l| - \ldots + (-1)^{n+1} |P_1 \cap \ldots \cap P_n| \qquad (1)$$

Where $|P_i| = 2^{m-|c_i|}$.

$$|P_i \cap P_j| = \begin{cases} 0, \text{There are complement ary literals in } c_i \text{ } and \text{ } c_j. \\ 2^{m-|c_i \cup c_j|}, \text{ Otherwise.} \end{cases}$$

This leads to algorithm SPL-ER below.

Algorithm SPL-ER
Let $\sum = \{c_1, c_2, \ldots, c_n\}$ be a set of possibilistic clauses, and let $V = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ be the distinct value appearing in $\sum$ in a down-top order

```
i = n; $\_INC(\Sigma) = 0$;
While i>0
    Let $\Sigma_{\alpha i}$ denote $\alpha_I$-cut set of $\Sigma$,
    Call algorithm SPL-ER' to judge $\Sigma_{\alpha i}$
    If $\_INC(\Sigma_{\bar{\alpha} i}) > 0$ then $\_INC(\Sigma) = \alpha_i$ ,return;
    Else i = i-1;
End While;
7. return;
```

Algorithm SPL-ER'
$\sum = \{c_1, c_2, \ldots, c_n\}$ be a set of possibilistic clauses, and let M (|M|=m) be its set of atoms.

```
i=1, sum=0
While i<=n
    For all sets L that contain i clauses
        Union = the union of all the clauses in L
        If no complementary literals in Union
            then number=$2^{m-|Union|}$
        Else Number=0
    End For
    i:=i+1
End While
If sum=$2^m$ then _$INC(\sum)>0$;
Else _$INC(\sum)=0$;
18.Return;
```

4 Extension vs. Resolution

It is clear that the worst-case time complexity of Algorithm SPL-ER is exponential. However there are cases in which Algorithm SPL-ER is tractable. For example, if each pair of clauses in a set contains complementary literal(s), then the complexity of Algorithm SPL-ER will be polynomial in the number of clauses and the number of values appearing in $\sum$ because only the first n terms in Formula (1) are nonzero terms and need to be computed. Intuitively, in this case resolution-based methods are likely to be inefficient since there are potentially many resolutions that need to be performed. So this leads to a new method for knowledge compilation in possibilistic logic. The idea is that we can always find an equivalent set of possibilistic clauses that do satisfy the above condition. Another extreme case is that if there are no complementary literals at all. In this case we would have to compute all the 2^m terms in order to decide its inconsistency using the possibilistic extension rule. But its inconsistency can be decided immediately by using a resolution-based method.

Let us visualize a spectrum of possibilistic reasoning problems: at one end (say the left) of the spectrum are problems for which there are complementary literal(s) for any pair, at the other end (say the right) of the spectrum are problems for which there are no complementary literals. It is likely that a (SPL-ER)-based method will be more efficient at the left end and a possibilistic resolution-based method at the right end. However, the practical problems always fall in between these extremes. Then this leads an open problem: when to use extension and when to use resolution. In this section we propose a roughly idea to solve this question.

Definition 4 (complementary factor)**.** Given a set of clauses $\Sigma = \{c_1, c_2, \ldots, c_n\}$, the *complementary factor* of the set is the ratio of the number of pairs that contain complementary literal(s) to the number of all the pairs in the set. That is $\frac{S}{n(n-1)/2}$, where S stands for the number of pairs that contain complementary literal(s).

Intuitively, the higher the complementary factor of a problem is, the nearer the problem is to the left end of the spectrum and the more efficient Algorithm SPL-ER can be expected to be; the lower the complementary factor of a problem is, the more efficient resolution can be expected to be.

In this sense, SPL-ER and be combined with any resolution-based possibilistic reasoning systems. Before reasoning, we should compute complementary factor first, If it is high, we call SPL-ER, else we call resolution-based methods.

5 Experimental Results

Experimental Results are shown in this section. The aim of the experiments is to show the relationship between the efficiency of SPL-ER and complementary factor. In other words, we are mainly concerned with how the complementary factor influences our algorithm in practice. We make the experiments on a Dell PC with CPU P4-2.8G, 512M memories. We implemented SPL-ER using Visual C++ in the operating system Windows XP Professional Edition.

The problem instances are obtained by a random generator with five parameters. It takes as an input the number of variable n, the number of clauses m and the length of each clause k, the value of a given clause v and obtains each clause randomly by choosing k variables from the set of n variables and by determining the polarity of each literal with probability p=0.5. The results given are mean values on 20 experiments per experiment and the inconsistent degree in each experiment is the same; in some instances the digits following the decimal point have been deleted. It can be seen from the results that when the complementary factor is low, the performance of our algorithms is relatively low. When complementary factor is high, our algorithms have a relatively good performance.

Table 1.

(No. of clauses, variables, length of clauses)	Complementary factor	Seconds
(200,15,15)	0.93	0.062
(200,15,15)	0.82	0.422
(200,15,15)	0.75	2.094
(200,15,15)	0.72	3.625
(200,15,15)	0.44	14.625
(200,15,15)	0.30	19.156
(200,15,15)	0.25	32.047
(400,15,15)	0.88	7.719
(400,15,15)	0.66	48.656.
(400,15,15)	0.53	68.469
(400,15,15)	0.37	103.750

6 Conclusion

This paper explores what happens if we use the inverse of possibilistic resolution by proposing a new method for reasoning in possibilistic logic. It can be considered, in a sense, a method dual to possibilistic resolution based reasoning method. What makes it promising is that it can be considered a framework for knowledge compilation for possibilistic knowledge base. This paper is a first attempt to do this work. Of course further work must be done to obtain more refined results.

Acknowledgements

This research was National Natural Science Foundation of China (Grant No. 60473003), Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20050183065) the Science and Technology Development Program of Jilin Province of China (Grant No. 20040526), the Outstanding Youth Foundation of Jilin Province of china (Grant No. 20030107) and Science Foundation for Yong Teachers of Northeast Normal University (Grant No. 20051001).

References

1. D. Dubois, H. Prade (1987) Necessity measures and the resolution principle, IEEE Trans. On Sys., Man and Cyber., 17:474-478.
2. D. Dubois, H. Prade, 1990. Resolution Principles in possibilistic logic. Int. Journal of Approximate Reasoning, 4(1):1-21.
3. D. Dubois, J. Lang, H. Prade (1991) Fuzzy sets in approximate reasoning-Part 2: logical approaches. Fuzzy sets and systems, 40:203-244.
4. Lin Hai, Sun Jigui, Zhang Yiming (2003) Theorem Proving Based On Extension Rule. Journal of Automated Reasoning, 31(1), 31:11-21.
5. Lin Hai, Sun Jigui (2004) Knowledge Compilation Using Extension Rule. Journal of Automated Reasoning, 32(2): 93-102.
6. Salem Benferhat and Henri Prade (2006) Compiling possibilistic knowledge bases, in proceedings of ECAI 2006.
7. D. Cayrol, D. Dubois, H. Prade (1995) Practical model-based diagnosis with qualitative possibilistic uncertainty, Proceedings of UAI 1995, pp 68-76.
8. D. Dubois, etc (2003) Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach, Artificial Intelligence 148: 219-260.
9. Selman, B., & Kautz, H. A (1991) “Knowledge compilation using Horn approximation.” In Proceedings of AAAI-1991, pp 904-909.
10. A. del Val (1994) Tractable databases: How to make propositional unit resolution complete through compilation, In J. Doyle, E. Sandewall, and P. Torasso, editors, Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning, pp 551-561.
11. R. Schrag (1996) Compilation for critically constrained knowledge bases, In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), pp 510-515.
12. P. Marquis, (1995), Knowledge compilation using theory prime implicates. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), pp 837- 843.
13. A. del Val, (2000), On some tractable classes in deduction and abduction. Artificial Intelligence, 116:297-313.

Applying Genetic-Fuzzy Approach to Model Polyester Dyeing

Maryam Nasiri[1], S. Mahmoud Taheri[2], and Hamed Tarkesh[3]

[1] Department of Textile Engineering
[2] School of Mathematical Sciences
[3] Department of Industrial & Systems Engineering,
Isfahan University of Technology, Isfahan 84156, Iran. Maryam Nasiri
mn_iutacir@yahoo.com

Abstract. Genetic-Fuzzy (GF) approach is used to model color yield in the high temperature (HT) polyester dyeing as a function of dye concentration, temperature, and time. The proposed method is compared with statistical regression model for their prediction powers using the Mean Square Errors (MSE). The results show that the GF model is preferable to the statistical regression model in at least two ways. First, it enjoys a better predictive power regarding to its minimum MSE. Second, the statistical regression model fails to satisfy the standard requirements.

Keywords: Fuzzy Inference System, Genetic Algorithm, Statistical Regression, Modeling, Polyester Dyeing, Disperse Dye.

1 Introduction

Many real world problems are so complex that classical computing methods fail to deal with them efficiently. An alternative approach to deal with these problems would be soft-computing. This approach includes the fuzzy set theory, the evolutionary search methods, and neural networks. The Fuzzy Inference System (FIS) is a powerful tool for modeling real systems; its unknown parameters, however, can be adjusted when properly coupled with evolutionary search methods such as genetic algorithms.

Genetic algorithm (GA) [1] is a parallel search method that can be used to find the near optimum structure of an FIS. Such a structure is formed by its membership functions and inference operations. The FIS knowledge base is usually defined by an expert and a genetic algorithm may be used to accompany the expert experiences for achieving higher performance.

Once soft computing methods were developed, many attempts were made to apply them in textile research. Sztandera and Pastore reviewed soft computing methods [2]. Hung and Hue used FIS to control continuous dyeing [3]. Jahmeerbacusa et al studied fuzzy dye bath pH control in exhaust dyeing [4]. Marjoniemi and Mantysalo applied Adaptive Neuro Fuzzy Inference Systems (ANFIS) in modeling dye solution and concentration [5,6]. Tavanai et al used fuzzy regression method to model the color yield in dyeing [7]. Callhof and Wulfhort applied ANFIS [8], Veit et al employed Neural Networks [9], and Nasiri used fuzzy regression in texturing [10]. Guifen et al used

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 608–617, 2007.
springerlink.com

neural network techniques to predict the Warp Breakage rate in weaving [11]. Predicting the properties of Melt Spun Fibers by neural network was studied by Chung-Feng [12]. Jeng-Jong used genetic algorithm to obtain the best combination of weaving parameters for woven fabric designs [13]. Sette et al used soft computing techniques to Fiber-to-Yarn production process [14]. Blaga studied the application of genetic algorithms to knitting technology [15]. An automatic textile sales forecast using fuzzy treatment of explanatory variables was used by Thomassey et al [16]. Peeva and Kyosev examined fuzzy relational calculus theory with applications in various engineering subjects such as textile [17].

In this paper, FIS is used to model the dyeing process. The genetic algorithm is then applied to tune the weights of FIS rules with the objective of achieving lower MSE levels in test data.

The paper is arranged along the following lines. Section 2 provides a description of dyeing process. Methods of modeling are discussed in Section 3. The performance of the proposed method is compared with that of statistical regression in Section 4. Finally, a brief conclusion will be presented in Section 5.

2 Dyeing

The most important man-made fiber is polyester which is produced by the melt-spinning process [18]. The dyeing of polyester is limited only to disperse dyes. However, the penetration of disperse dyes is hampered by the compact structure of polyester fibers under normal dyeing conditions. Dyeing of polyester fibers, thus, requires special conditions such as high temperature (≈130ºC), dry heat (190 - 220ºC), or the use of a carrier in the dye bath. The chemical structure of disperse dyes contains polar groups but there are no ionic groups present, leading to their very low solubility in water [19, 20].

Temperature, time, and disperse dye concentration are the primary factors affecting color yield when dyeing polyester. The relative importance of these factors can be seen in models representing color yield as a function of these parameters. These models may also have applications in processing and cost minimization. The main objective of this study is to develop a model of color yield for polyester dyed in the high temperature dyeing process. Statistical regression and Genetic-Fuzzy System are used in developing the model. The color yield is designated as K/S, representing the ratio of the light absorbed by an opaque substrate to the light scattered from it. This ratio is calculated from Kubelka-Munk relation [21]:

$$(K/S)_{\lambda}=(1-R_{\lambda})^{2}/(2R_{\lambda}) \quad (1)$$

3 Modeling of Polyester Dyeing

The present study aims to model variations of color yield of C.I. Disperse Blue 266 versus time, temperature, and disperse dye concentration in the high temperature (HT) polyester dyeing process using the GF approach and the statistical regression method. A total number of 120 polyester samples were dyed according to the conditions in Table 1. The models based on statistical regression and GF approaches were developed and compared. Finally, the models were used for 120 polyester samples, which hade been dyed with C.I. Disperse Brown 1 according to conditions in Table 1. It should be mentioned that C.I. Disperse Brown 1 has a chemical structure similar to that of C.I. Disperse Blue 266.

Table 1. Dyeing conditions

Dye Concentration (%owf)	0.75	1.5	3	4.5	6		
Temperature $(°C)$	100	110	115	120	120	125	130
Time (min)	12	24	36	48			

3.1 Modeling by Statistical Regression

A general multiple regression for modeling the color yield can be considered to take the following form:

$$Y = A_0 + A_1X_1 + A_2X_2 + A_3X_3 \qquad (2)$$

The least squares method is used to obtain the coefficients in the above equation. The statistical regression model is then obtained as fallows:

$$K/S=-79.4+0.71Conc+0.11Time+1.54Temp$$

To verify the necessary statistical regression conditions, the following four conditions can be used [22]:

1. Linear form for the normal plot of the residuals.
2. I chart of residuals should lie between the upper and lower control limits without any specific pattern.
3. The residuals histogram should be of an approximately normal form.
4. Residuals versus fitted values should show no specific pattern.

The information on the statistical regression model for C.I. Disperse Blue 266 is shown in Figure 1. It is seen that this statistical model cannot be accepted with regard to the four conditions mentioned above.

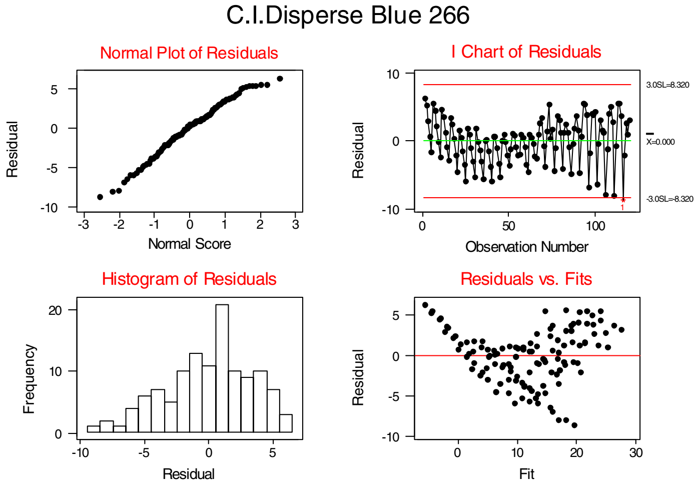


Fig. 1. Plot of Residuals for K/S of C.I. Disperse Blue 266

The statistical regression model was obtained for C.I. Disperse Brown 1 along the following lines:

$$K/S=-74.7+0.64Conc+0.12Time+3.46Temp$$

This model, similar to that for C.I. Disperse Blue 266, fails to meet the necessary conditions.

3.2 Variations of Color Yield in Terms of Time, Temperature, and Dye Concentration

Figures 2 and 3 show variations in K/S function for the variables temperature, time and concentration in the case of C.I. Disperse Blue 266. The following observations can be made in these figures.

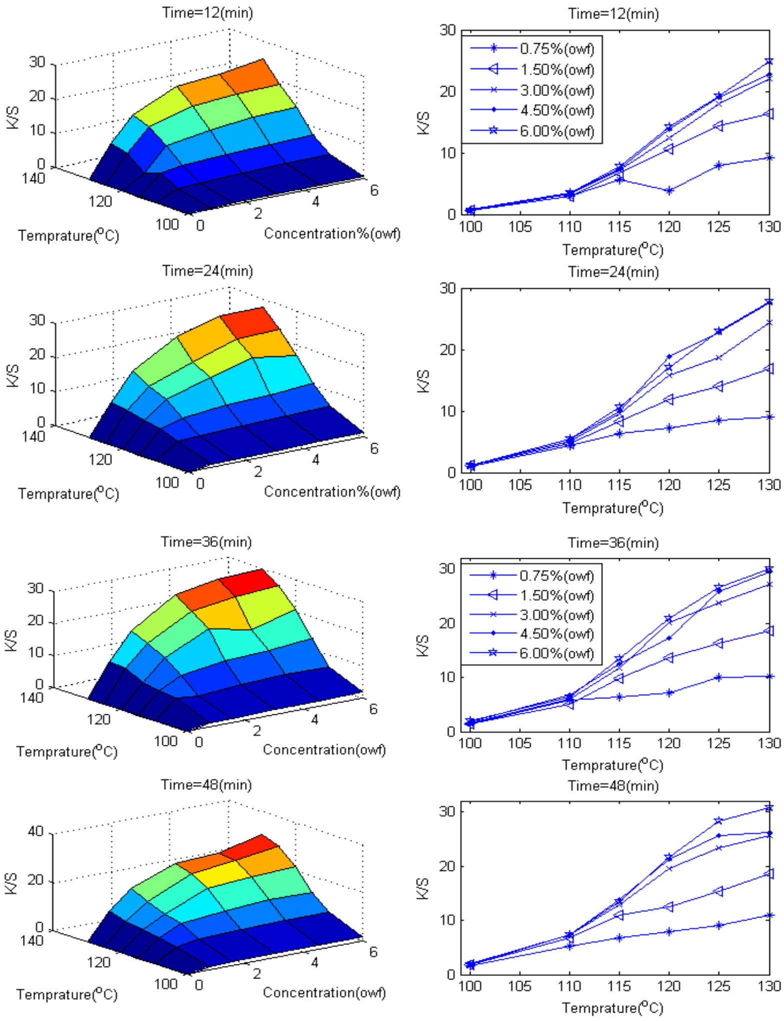


Fig. 2. K/S in terms of temperature and concentration

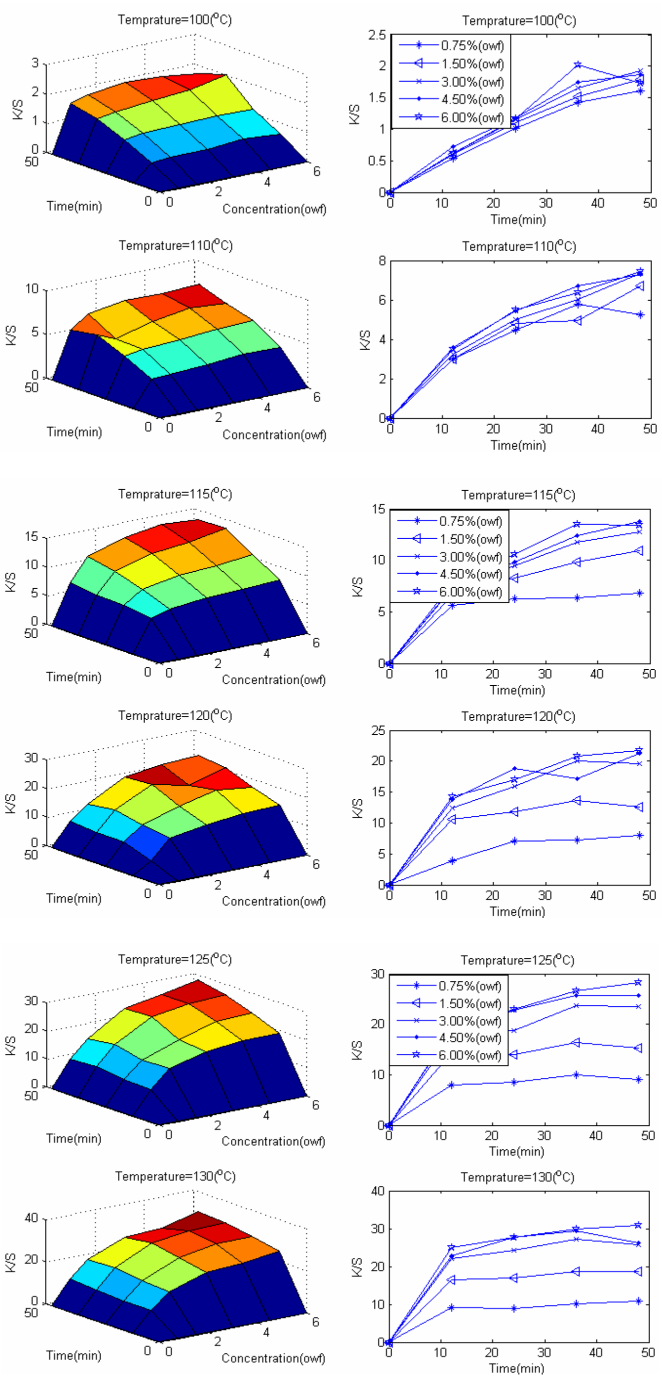


Fig. 3. K/S in terms of time and concentration

According to Figure 2, which shows the effect of temperature on K/S values of dyed samples at times 12, 24, 36 and 48 (min), there is a mild upward trend in K/S variations with increasing temperature, but a sharp rise at higher concentrations.

Figure 3 shows the effect of time on color yield of dyed samples at different temperatures of 100,110,115,120,125 and 130 °c. It is seen that the K/S values of the samples rise steadily with time and reach a peak especially for lower dyestuff concentrations, a trend which seems to continue onwards from a temperature of 115°c upwards,. In the same way, this figure indicates that differences in K/S values of samples dyed at different dyestuff concentrations are more significant at higher temperatures.

In view of the above figures, especially Figure 2, it is clearly observed that dye concentration has a greater effect on K/S value than time does. In conclusion, the results show that temperature and concentration are more effective than time in the dyeing process.

3.3 Genetic-Fuzzy Approach to Modeling the Dyeing Process

The Fuzzy Inference System, representing the color yield of C.I. Disperse Blue 266 as a function of time, temperature, and disperse dye concentration in the high temperature (HT) polyester dyeing, was developed along the following lines [23,24]:

First, membership functions for input and output variables have been determined. Tables 2 and 3 show the ranges of input and output variables, defined in terms of polyester dyeing conditions. The Gaussian membership function was used for all input and output variables. The values for mean and standard deviation of membership function for each variable are given in Tables 2 and 3.

Table 2. Parameters of fuzzy set for input variables

Fuzzy set	Concentration		Time		Temperature	
	Mean	Std	Mean	Std	Mean	Std
Low	1.28	0.848	13.1	14.2	100	9.61
Medium	3.38	0.891	-	-	119	1.64
High	5.87	1.33	44.4	11.7	129	3.97

Table 3. Parameters of fuzzy set for output variable

Fuzzy set	Color yield (K/S)	
	Mean	Std
Very low	3.29	0.625
Low	4.11	1.42
Medium	12.8	1.69
High	19.3	1.86
Very high	29.8	223

In the second stage, the nine rules below were defined according to the physical and chemical structure of polyester fiber, HT dyeing of polyester, and the behavior of 120 samples dyed in C.I. Disperse Blue 266:

1. If (temperature is low) and (time is low) and (concentration is low), then (K/S is very low).
2. If (temperature is medium) and (concentration is high), then (K/S is high).
3. If (temperature is high) and (concentration is low), then (K/S is medium).
4. If (temperature is high) and (concentration is medium), then (K/S is high).
5. If (temperature is low) and (time is high) and (concentration is low), then (K/S is very low).
6. If (temperature is high) and (concentration is high), then (K/S is very high).
7. If (temperature is medium) and (time is low) and (concentration is high), then (K/S is medium).
8. If (temperature is medium) and (time is high) and (concentration is high), then (K/S is high).
9. If (Temperature is low) and (time is low) and (concentration is high), then K/S is low.

Finally, the weights of the rules were tuned using a genetic algorithm. The population size of GA was 100 divided in 5 sub-populations. The Line recombination with mutation features [25] method was used for a crossover, and mutation of real-valued

w_1	w_2	w_3	…	w_9

Fig. 4. The chromosome structure of the genetic algorithm

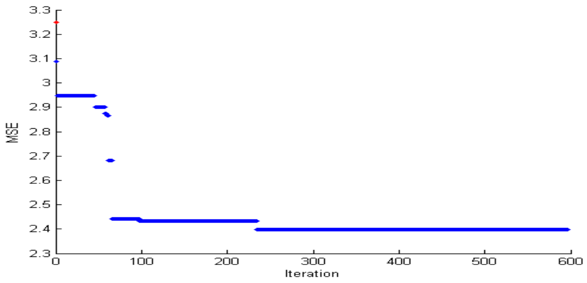


Fig. 5. The convergence of GA to a near optimum FIS

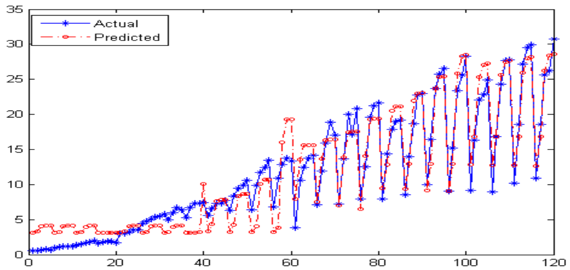


Fig. 6. K/S of dyed samples and predicted K/S by GF

population [26] method was used for mutation. The crossover rate, mutation rate, and insertion rate were adjusted to 0.8, 0.11 and 0.95, respectively. Figure 4 shows the structure of chromosomes in this application. The convergence behavior of GA for a sample run is shown in Figure 5.

Figure 6 shows the results for the GF approach applied to 120 dyed samples of C.I. Disperse Blue 266.

4 Comparison of Statistical Regression and Genetic-Fuzzy Approaches

In a manner similar to the one described above, the proposed method was applied to C.I. Disperse Brown with no changes in the FIS designed for the previous dye but with changes in rule weights. An MSE of 4.187 was obtained, which is smaller than the values for the statistical regression and the fuzzy inference system approaches. Table 4 shows a comparison of the predictive powers by different models, using Mean Square Errors (MSE).

Table 4. Parameters of fuzzy set for output variables

MSE	Blue	Brown
Statistical Regression	3.382	4.474
Fuzzy Inference System	3.307	4.446
Genetic-Fuzzy Approach	2.410	4.187

On the basis of the above considerations, it can be said that the GF approach provides an appropriate method to predict the color yield.

5 Conclusion

Soft computing has been applied to model the dyeing process. We have predicted the color yield in terms of time, temperature, and disperse dye concentration in the high temperature (HT) polyester dyeing process using statistical regression and Genetic-Fuzzy approaches. We have found that the prediction performance is best for the GF approach. Using GA for simultaneous tuning of both rule weights and the parameters of membership functions may lead to better results. This can be further investigated in future studies.

Acknowledgments. The authors feel greatly indebted to Dr. Hossein Tavanai for his supervision during the first author's work on her MS dissertation when he lent his full support and knowledge to obtain the data used here and to study the behavior of the dyes. This study was partially supported by the Fuzzy Systems and Its Applications Center of Excellence, Shahid Bahonar University of Kerman, Iran.

References

1. Goldberg, D.: Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA (1989).
2. Sztandera, L.M., Pastore, C.: Soft Computing in Textile Sciences, Springer, Berlin (2003)
3. Hung, C.C., Yu, W.H.: Control of Dye Concentration, PH, and Temperature in Dyeing Processes, Textile Research Journal 69 (1999) 914-918.
4. Jahmeerbacusa, M.I., Kistamahb, N., Ramgulamb, R.B., Fuzzy Control of Dyebath PH in Exhaust Dyeing, Coloration Technology, 120 (2004) 51-55
5. Marjoniemi, M., Mantysalo, E.: Neuro-Fuzzy Modeling of Spectroscopic Data. Part A-Modelling of Dye Solutions, J.S.D.C 113 (1997) 13-17
6. Marjoniemi, M., Mantysalo, E.: Neuro-Fuzzy Modeling of Spectroscopic Data. Part B–Dye Concentration Prediction, J.S.D.C 113 (1997) 64-67.
7. Tavanai, H., Taheri, S.M., Nasiri, M.: Modeling of Colour Yield in Polyethylene Terphethalate Dyeing with Statistical and Fuzzy Regression, Iranian Polymer Journal 14 (2005) 954-968.
8. Callhof, C., Wulfhort, B.: Neuro-fuzzy Networks- A Tool For the Development of New Yarn Contacting Exemplary on Friction Discs, Man Made Fiber Yarn Book 95 (1999).
9. Veit, D., Batista-de –Souza, P., Wulfhorst, B., Application of a Neural Network in the False Twist Texturing Process, Chemical Fibers International 48 (1998) 155-156
10. Nasiri, M.: Fuzzy Regression Modeling of False-Twist Texturing Yarn, IFSA 2005 Conference, Beijing, China, (2005) 142-146.
11. Guifen, Y., Jiansheng, G., Yongyuan, Z.: Predicting the Warp Breakage Rate in Weaving by Neural Network Technicqes, Textile Research Journal 75 (2005). 274-278.
12. Chung-Feng, K.J., Kun-Iuan, H., Yi-Shiuan, W.: Using Neural Network Theory to Predict the Properties of Melt Spun Fibers, Textile Research Journal 74 (2004) 840-843.
13. Jeng-Jong, L.: A Genetic Algorithm for Searching Weaving Parameters for Woven Fabrics, Textile research journal 73 (2003) 105-112.
14. Sette, S., Boullart, L., Van Langenhof, L.: Building a Rule Set for the Fiber-to-Yarn Production Process by Means of Soft Computing Techniques, Textile Research Journal 70 (2000) 375-386.
15. Blaga, M., Draghici, M.: Application of Genetic Algorithms in Knitting Technology, The Journal of the Textile Institute, 97 (2004) 175-178.
16. Thomassey, S., Happiette, M., Castelain,J.M.: An automatic Textile Sales Forecast Using Fuzzy Treatment of Explanatory Variables, European Journal of Operational Research 16 (2005) 275-284.
17. Peeva K., Kyosev Y., Fuzzy Relational Calculus, Advanced in Fuzzy Systems- Application and Theory 22 (2005).
18. Goorden Cook, J.: Handbook of Textile Fibers, man-made fiber, Merro Technical Library (1984)
19. Johnon, A.: The Theory of Coloration of Textiles, Second Ed., Bardford: S.D.C. Pub. (1989).
20. Nunn, D.M.: The Dyeing of Synthetic Polymer and Acetate Fibers, Dyers Company Publication Trust (1979).
21. Baumann W., et al.: Determination of Relative Colour Strength and Residual Colour Difference by Means of Reflectance Measurements, J. Soc. Dyers and Colour, 103 (1987) 100-105.
22. Montgomery, D.C., Peek C.A.: Introduction to Linear Regression Analysis, Second Ed., J. Wiley (1991).

23. Zimmermann H. J.: Fuzzy Set Theory and Its Applications, Third Ed. Kluwer Akademic Publishers (1996).
24. Pedrycz, W., Gomidee, F., An Introduction to Fuzzy Sets, Analysis and Design, Prentice-Hall (1998)
25. Muehlenbein, H.: The Breeder Genetic algorithm- a Provable Optimal search Algorithm and Its Application, IEE Colloquium, Application of Genetic Algorithm, London (1994).
26. Muehlenbein, H., Schlierkamp-Voosen, D.: Predictive Models For the Breeder Genetic Algorithm: I. Continuous Parameter Optimization, Evolutionary Computation 1 (1993) 25-49.

Gear Fault Diagnosis in Time Domains by Using Bayesian Networks

Yuan Kang[1], Chun-Chieh Wang[2], and Yeon-Pun Chang[1]

[1] Department of Mechanical Engineering, Chung Yuan Christian University
yk@cycu.edu.tw
[2] R&D Center for Membrane Technology, Chung Yuan Christian University,
200, Chung Pei Rd., Chung Li 32023, Taiwan

Abstract. Fault detection in gear train system is important in order to transmitting power effectively. The artificial intelligent such as neural network is widely used in fault diagnosis and substituted for traditional methods. In rotary machinery, the symptoms of vibration signals in frequency domain have been used as inputs to the neural network and diagnosis results are obtained by network computation. However, in gear or rolling bearing system, it is difficult to extract the symptoms from vibration signals in frequency domain which have shock vibration signals. The diagnosis results are not satisfied by using artificial neural network, if the training samples are not enough. The Bayesian networks (BN) is an effective method for uncertain knowledge and less information in faults diagnosis. In order to classify the instantaneous shock of vibration signals in gear train system, the statistical parameters of vibration signals in time-domain are used in this study. These statistical parameters include kurtosis, crest, skewness factors etc. There, based on the statistical parameters of vibration signals in time-domain, the fault diagnosis is implemented by using BN and compared with two methods back-propagation neural network (BPNN) and probabilistic neural network (PNN) in gear train system.

1 Introduction

Gearboxes are widely used in rotary machinery in order to transmitting power. The definition of the gear fault diagnosis is differentiating faults from vibration signals by expert knowledge or experiences when the gear system broke down. The vibration signals always carry the important information in the machine. It is important to extract the symptoms of the vibration signals from accelerometers and use these symptoms to classify fault by artificial inference in the gear system. Although the most common method of fault diagnosis in rotary machinery is used frequency spectral analysis, it is difficult to extract the symptoms from vibration signals in frequency domain which have shock vibration signals in gear train system. Dyer and Stewart [1] use statistical analysis of vibration signals in time-domain to fault diagnosis in rolling element bearing. The symptoms of vibration signals in time-domain could display clearly for fault diagnosis in gear train system.

The traditional fault diagnosis may be influenced by human subjectivity and experience, and the weighting parameters also designed by human thought. Because the rules given by various experts are different, the diagnosis results are not consistent. Recently, BPNN is a kind of neural network which is widely used to solve fault diagnosis

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 618–627, 2007.
springerlink.com

problems, which has been innovated by Rumelhart and McClelland [2] and developed to be a diagnosis method by Sorsa et al. [3]. Kang et al. [4] extracted frequency symptoms of vibration signals to detect faults of electric motor by using BPNN. However, it is hard to identify the faults by using frequency spectrum analysis in gear train system, because the components of frequency spectrum are complex and ambiguous. And when training samples are not enough, the diagnosis results are not accurate for diagnostic samples which are not existence within trained samples. The variations of vibration signals in time-domain are distinct to differentiate faults of gear train system.

Many artificial neural networks (ANNs) have been proposed before, however, the slow repeated iterative process and poor adaptation capability for data restrains the ANNs applications. An effective and flexible probabilistic neural network (PNN) could overcome these drawbacks. PNN presented in [5] is different from other supervised neural network, since the weightings don't alternate with training samples. The output values of PNN are obtained by once forward network computation. Lin et al. [6] used PNN to identify faults in dissolved gas content, while using the gas ratios of the oil and cellulosic decomposition to create training examples.

Based on Bayesian principle, the probability can estimate by prior probability of previous samples. This method is well used to these problems when the information is not enough. BN have proven useful for a variety of monitoring and predictive purposes. Applications have been documented mainly in the medical treatment [7], gas turbine engines [8], and industrial fault diagnosis [9]. Chien et al. [10] constructs a BN on the basis of expert knowledge and historical data for fault diagnosis on distribution feeder. On the other hand, Giagopulos et al. [11] use Bayesian statistical framework to estimate the optimal values of the gear and bearing model parameters and be achieved by combining experimental information from vibration measurements with theoretical information to build a parametric mathematical model.

The statistical parameters of vibration signals in time-domain including waveform, crest, impulse, allowance, kurtosis and skewness parameters are proposed by Heng and Nor [12]. Due to the statistical parameters could represent the explicit symptoms for gear train system and the classification for uncertainty information by using BN, in this study, combined these methods to diagnosis faults in gear train system. These six statistical parameters of vibration signals in time-domain are used for fault diagnosis with BPNN, PNN and BN methods. Also, BN results are compared with both methods BPNN and PNN in fault diagnosis of gear train system.

2 Statistical Parameters of Vibration

The six statistical parameters of time history have been defined by Heng and Nor [12] as follows:

(a) The waveform factor shows the indication of shift in time waveform and determined by

$$Waveform\ (X_W) = \frac{r.m.s\ value}{mean\ value} = \frac{X_{rms}}{|\overline{X}|} \tag{1}$$

(b) The crest factor shows the indication of peak height in time waveform and determined by

$$Crest\ (X_C\ ) = \frac{max\ peak}{r.m.s\ value} = \frac{max\left|X\right|}{X_{rms}} \tag{2}$$

(c) The impulse factor shows the indication of shock in time waveform and determined by

$$Impulse\ (X_I\ ) = \frac{max\ peak}{mean\ value} = \frac{max\left|X\right|}{\left|\bar{X}\right|} \tag{3}$$

(d) The allowance factor shows the indication of plenty in time waveform and determined by

$$Allowance\ (X_A\ ) = \frac{max\ peak}{X_r} = \frac{max\left|X\right|}{X_r}\ , X_r = \left(\left(1/N\right)\sum\nolimits_{n=1}^{N} x(n)^{1/2}\right)^2 \tag{4}$$

where $x(n)$ is time waveform of vibration signal.

(e) The skewness and kurtosis factors are both sensitive indicators for the shape of the signal, which are relative to third and fourth moment of signal distribution in time-domain, respectively. The skewness factor corresponds to the moment of third order norm of vibration signal, which is determined by

$$Skewness\ (X_S\ ) = \frac{M_3}{\sigma_3} \tag{5}$$

$$M_3 = \left(1/N\right)\sum\nolimits_{n=1}^{N}(\ x(\ n\ ) - \bar{X}\ )^3\ ,\ \sigma = (\ \left(1/N\right)\sum\nolimits_{n=1}^{N}(\ x(\ n\ ) - \bar{X}\ )^2\ )^{1/2} \tag{6}$$

where M_3 is the third order moment and σ is standard deviation. The kurtosis factor corresponds to the moment of fourth order norm of vibration signal, which is determined by

$$Kurtosis\ (X_K\ ) = \frac{M_4}{\sigma_4}\ ,\ M_4 = \left(1/N\right)\sum\nolimits_{n=1}^{N}(\ x(\ n\ ) - \bar{X}\ )^4 \tag{7}$$

where M_4 is the fourth order moment. For kurtosis parameter, the values are 1.5 and 1.8 for sine and triangle wave signals in time history, respectively; for crest parameter, the values are 1.7 and 1.0 for triangle and square wave signals in time history, respectively. Although the simple signals could be checked easily with theoretical values of statistical parameters, the vibration signals of gear train system is complex and could not checked for different gear faults.

In this study, the training samples are obtained by simulating corresponding gear faults on experimental rotor-bearing system. Besides the normal gear (O_1) signals, there are two kinds of faults which including tooth breakage fault (O_2) and wear fault (O_3). There are ten training samples for each fault and is listed in Table 1. Each training sample have six statistical parameter of vibration signals which including waveform, crest, impulse, allowance, kurtosis and skewness parameters. Similarly, there are five diagnostic samples for each fault and is listed in Table 2.

Table 1. The training samples of statistical parameters of vibration signals

Training sample	Parameter Test No.	Waveform	Crest	Impulse	Allowance	Kurtosis	Skewness
1	1	1.28748	4.16475	5.36206	6.42607	2.94202	-1.38656
	2	1.33577	4.42759	5.91424	7.23393	3.45039	-1.43746
	3	1.26744	4.90187	6.21281	7.36275	2.8115	-1.36702
	4	1.26649	5.35327	6.77986	8.01719	2.93045	-1.39855
	5	1.28831	4.25197	5.47787	6.55841	3.03643	-1.39166
	6	1.27132	5.16768	6.56977	7.80874	2.89777	-1.38012
	7	1.25515	4.72966	5.93644	6.99951	2.77261	-1.38922
	8	1.2625	6.26067	7.9041	9.34875	2.88097	-1.39203
	9	1.27562	4.9493	6.31342	7.50522	2.95553	-1.39151
	10	1.27298	5.63142	7.16872	8.52778	2.92079	-1.3766
2	1	1.27916	4.82934	6.17751	7.34722	3.03417	-1.39694
	2	1.30721	8.2694	10.80986	12.91118	5.376	-1.48271
	3	1.32963	10.52056	13.98841	16.84124	5.03708	-1.43339
	4	1.27883	5.11315	6.53884	7.79305	3.1635	-1.4335
	5	1.25876	6.35699	8.0019	9.45222	2.94453	-1.42286
	6	1.29752	10.30627	13.37261	15.88942	6.08271	-1.46424
	7	1.36039	11.89035	16.17552	19.47433	7.94966	-1.39051
	8	1.3236	9.8308	13.01206	15.63566	4.64462	-1.3523
	9	1.30498	9.86201	12.86971	15.36947	4.70865	-1.39747
	10	1.26212	5.49845	6.93971	8.19017	2.92291	-1.39796
3	1	1.68451	11.44801	19.28425	25.26632	14.76671	-0.99271
	2	1.68947	10.91075	18.43343	23.91418	16.44452	-0.95952
	3	1.70213	10.60673	18.05407	23.92433	14.38681	-1.01801
	4	1.58438	10.36547	16.42281	21.0378	11.10712	-1.05348
	5	1.53747	11.87461	18.2568	22.97949	12.0315	-1.10121
	6	1.7177	11.00257	18.89916	25.00289	15.44512	-1.07667
	7	1.63984	11.55684	18.95141	24.56436	14.65595	-1.05854
	8	1.54895	10.27717	15.91886	20.20572	9.62844	-1.08054
	9	1.60122	10.12086	16.20569	20.79312	11.47277	-0.99938
	10	1.67369	10.68908	17.89025	23.13652	16.18965	-0.99548

Table 2. The diagnostic samples of statistical parameters of vibration signals

Diagnostic sample	Parameter Test No.	Waveform	Crest	Impulse	Allowance	Kurtosis	Skewness
1	1	1.28083	4.9557	6.34742	7.57411	2.92901	-1.37241
	2	1.27573	4.24116	5.41058	6.45568	2.86223	-1.38532
	3	1.2797	4.25527	5.44549	6.50587	2.91428	-1.39213
	4	1.25417	3.88207	4.86876	5.75323	2.73299	-1.39766
	5	1.25061	4.49917	5.62672	6.62779	2.65127	-1.36925
2	1	1.28311	7.37059	9.45725	11.26441	3.39816	-1.41329
	2	1.31473	7.68456	10.10314	12.19807	3.79101	-1.43346
	3	1.29129	6.41499	8.28359	14.92663	3.12193	-1.38309
	4	1.29454	6.84103	8.85602	10.61262	3.44279	-1.42621
	5	1.29993	7.1357	9.27591	11.12575	3.66325	-1.45471
3	1	1.36765	11.41049	19.02875	24.67334	15.31862	-0.95438
	2	1.53771	10.51913	16.17538	20.65024	8.64434	-1.0645
	3	1.54835	11.44328	17.71818	22.44819	11.37889	-1.12111
	4	1.62486	11.48614	18.66335	24.16117	12.91246	-1.06472
	5	1.60378	11.81417	18.94734	24.71282	10.43235	-1.05344

3 Bayesian Networks

According to Bayes' theorem [13] for a hypothesis *H* and an evidence *E* been giving the conditional (posterior) probability can be determined by

$$P(H|E)=\frac{P(E|H)\times P(H)}{P(E)} \tag{8}$$

where $P(H|E)$ is the posterior probability for *H* giving the information that the evidence *E* is true, $P(H)$ is the prior probability of *H*, and $P(E|H)$ is the probability for evidence *E* giving the information that the hypothesis *H* is true.

Perrot et al. [14] used Gaussian probability distribution law to conditional probability density function. The equation of a Gaussian probability law generalized to *n* dimensions is

$$P(E_i|H_j)=\frac{1}{(2\pi)^{i/2}(\det\Gamma[j])^{1/2}}\exp[-\frac{1}{2}(X_i-E_j)\Gamma[j]^{-1}(X_i-E_j)^T] \tag{9}$$

where X_i and E_j are *i*-component row vector for statistical parameters and mean values for *j*-th category. $\Gamma[j]$ is the $i\times i$ covariance matrix for *j*-th category. In this study, a Bayesian network is designed with six statistical parameters (*i=1,2,...,6*) and three gear faults (*j=1,2,3*). The equation of covariance matrix $\Gamma[j]$ as follow:

$$\Gamma[j]=\begin{bmatrix} L_{11} & \cdots & L_{1n} \\ L_{21} & \cdots & L_{2n} \\ & & \\ L_{m1} & \cdots & L_{mn} \end{bmatrix},\quad L_{mn}=\sum_{k=1}^{10}x_m^{(j)}(k)\square x_n^{(j)}(k)-\frac{(\sum_{k=1}^{10}x_m^{(j)}(k)\square x_n^{(j)}(k))^2}{10} \tag{10}$$

where *m=n=1,2,…,6*. L_{mn} is the covariance value between x_m and x_n. Three probability density values $P(E|H_j)$ are obtained by Eq. (9) with statistical parameters of diagnostic sample. The max probability density value $P(E|H_j)$ represents the decision classification belongs to *j*-th category. And the posterior for *j*-th category as follow:

$$P(H_j|E)=\frac{P(E|H_j)}{\sum_{j=1}^{3}P(E|H_j)} \tag{11}$$

4 Back-Propagation Neural Network

A BPNN [3] is designed to fault diagnosis in gear train system which is composed by six input nodes, forty hidden nodes and three output nodes. The inputs and outputs variables are arrayed into a vector as express by

$$(X_W,X_C,X_I,X_A,X_K,X_S,O_1,O_2,O_3) \tag{12}$$

where six elements of inputs represent waveform (X_W), crest (X_C), impulse (X_I), allowance (X_A), kurtosis (X_K) and skewness (X_S) factors, and three elements of outputs including normal gear (O_1), tooth breakage fault (O_2) and wear fault (O_3).

Table 3. The membership function of statistical parameters of vibration signals in time-domain

Level / Parameter	VS (**1** 0 0 0 0)	S (0 **1** 0 0 0)	M (0 0 **1** 0 0)	L (0 0 0 **1** 0)	VL (0 0 0 0 **1**)
Waveform	$X_W<1.2665$	$1.2665\le X_W<1.2749$	$1.2749\le X_W<1.2942$	$1.2942\le X_W<1.3296$	$X_W\ge 1.3296$
Crest	$X_C<4.8293$	$4.8293\le X_C<5.6314$	$5.6314\le X_C<7.9098$	$7.9098\le X_C<10.1209$	$X_C\ge 10.1209$
Impulse	$X_I<6.1474$	$6.1474\le X_I<7.1687$	$7.1687\le X_I<10.1717$	$10.1717\le X_I<13.3726$	$X_I\ge 13.3726$
Allowance	$X_A<7.2594$	$7.2594\le X_A<8.5278$	$8.5278\le X_A<12.1927$	$12.1927\le X_A<15.8894$	$X_A\ge 15.8894$
Kurtosis	$X_K<2.8810$	$2.8810\le X_K<2.9820$	$2.9820\le X_K<3.5942$	$3.5942\le X_K<5.3926$	$X_K\ge 5.3926$
Skewness	$X_S<-1.4205$	$-1.4205\le X_S<-1.3920$	$-1.3920\le X_S<-1.3847$	$-1.3847\le X_S<-1.3215$	$X_S\ge -1.3215$

Since these statistical parameters have been used as inputs to the BPNN which used sigmoid function to activated function, the inputs and outputs of diagnostic network can be normalized into a range from 0 to 1 and are classified into several levels except for the skewness factor which is normalized into a range from -1 to 0. Too many levels are chosen caused the redundant computation, contrariwise, too few levels are chosen caused the ambiguous classification. In order to obtain the explicit classification, in this study, these statistical parameters are classified into five levels, one fifth, two fifths, three fifths and four fifths of total sample can be designated to be level limits of very small(VS), small(S), medium(M), large(L) and very large(VL) levels. Thus, the five levels of membership functions are expressed in Table 3. Thus, there are thirty neurons(F_i) in fuzzy layer.

The connections between the fuzzy and hidden layers, the hidden and output layers are weighting coefficient w_{ij} and w_{jk}, respectively. The input and output of the j-th neuron in the hidden layer respectively can be expressed by

$$net_j=\sum_{i=1}^{30} w_{ij}F_i\,,\ O_j=f(net_j)=1/(1+exp(-net_j)) \tag{13}$$

where F_i is the i-th fuzzy neuron, w_{ij} is the weighting coefficient between the i-th neuron in the fuzzy layer and the j-th neuron in the hidden layer. $f(\cdot)$ is the sigmoid function adopted to be the activation function. The output of BPNN can be expressed by

$$net_k=\sum_{j=1}^{m} O_j w_{jk}\ ,\ k=1,2,3,\ O_k=f(net_k)=1/(1+exp(-net_k)) \tag{14}$$

where m is the neuron number in the hidden layer, w_{jk} is the weighting coefficient between the j-th neuron in the hidden layer and the k-th neuron in the output layer.

The BPNN is trained using the error between the actual output and the ideal output, to modify w_{ij} and w_{jk} until the output of BPNN is close to the ideal output with an acceptable accuracy. On the basis of the gradient descent method for the minimization of error, the correction increments of weighting coefficients are defined to be proportional to the slope, related to the changes between the error estimator and the weighting coefficients as

$$\Delta w_{ij}=-\eta\frac{\partial E}{\partial w_{ij}}\,,\ \Delta w_{jk}=-\eta\frac{\partial E}{\partial w_{jk}} \tag{15}$$

where η is the learning rate used for adjusting the increments of weighting coefficients and controls the convergent speed of the weighting coefficients. E is the error estimator of the network and is defined by

$$E = \frac{1}{2}\sum_{p=1}^{N} E_p = \frac{1}{2}\sum_{p=1}^{N}\sum_{k=1}^{3}\left(T_k - O_k\right)_p^2 \tag{16}$$

where N is the total number of training samples. $(T_k)_p$ is the ideal output of the p-th sample and $(O_k)_p$ is the actual output of the p-th sample. Substituting equation (16) into equation (15) and executing derivations give the increments of weighting coefficients as

$$\Delta w_{jk} = \eta\delta_k O_j = \eta \cdot f'\left(net_k\right)\cdot\left(T_k - O_k\right)_p \cdot O_j \tag{17}$$

for the output layer, and

$$\Delta w_{ij} = \eta\delta_j O_i = \eta \cdot w_{jk} \cdot \delta_k \cdot f'(\, net_j \,)\cdot O_i \tag{18}$$

where $f'(\cdot)$ is the first derivation of $f(\cdot)$.

5 Probabilistic Neural Network

A PNN [5] is designed to fault diagnosis in gear train system and the input and output nodes are the same as the BPNN. PNN is different with other supervised neural network, since the weighting coefficients values don't alternate with training samples. Because the numbers of the hidden nodes represents the training samples, the connections between the input and hidden layers are initial weighting coefficient w_{ij} which indicates the symptom sets of the training samples. The connections between the hidden and output layers are weighting coefficient w_{jk} which indicates the fault sets of the training samples. The input of the j-th neuron in the hidden layer can be expressed by

$$net_j = \sum_{i=1}^{6}(X_i\text{-}w_{ij}\,)^2\,,\, j=1,2...,30 \tag{19}$$

where X_i is the i-th input which including waveform (X_W), crest (X_C), impulse (X_I), allowance (X_A), kurtosis (X_K) and skewness (X_S) factors, sequentially. w_{ij} is the weighting coefficient between the i-th input in the input layer and the j-th neuron in the hidden layer. The output of the j-th neuron in the hidden layer can be obtained by

$$O_j = exp(\text{-}net_j/2\sigma^2\,) \tag{20}$$

where σ is the smoothing parameter of gauss function. The output of PNN can be expressed by

$$net_k = \frac{1}{N_K}\sum_{j=1}^{30} O_j w_{jk}\ ,\ k=1,2,3 \tag{21}$$

where w_{jk} is the weighting coefficient between the j-th neuron in the hidden layer and the k-th neuron in the output layer, and also indicates the fault sets of the training samples. Then, the output O_k of the PNN can be obtained by

$$net_k = \max_r net_r,\ O_k = 1\,,\, r=1,2,3 \tag{22}$$

6 Case Studies

6.1 Experiment Set-Up

The experiment equipment of gear train system is shown in Figure 1. It consists of a motor, a converter and a pair of spur gears in which the transmitting gear has 46 teeth and the passive gear has 30 teeth. The vibration signals are measured in vertical direction from two accelerometers that mounted on the bearing housing of gear train system.

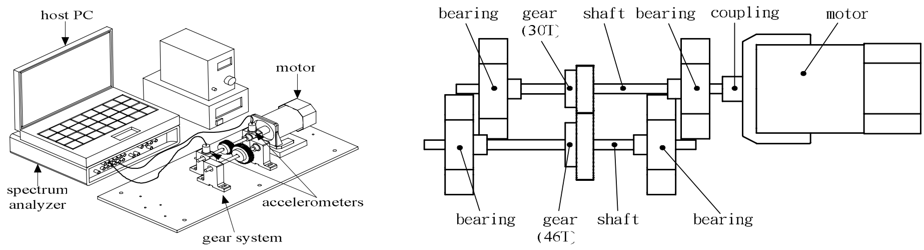


Fig. 1. The experimental setup of gear train system

6.2 Neural Network Diagnosis

When the BPNN accomplish training, using the six statistical parameters and normalized by membership function to compute with the trained weighting coefficients w_{ij} and w_{jk} and obtained output values. Each value represents a kind of gear fault and the output value represents the degree of certainty for corresponding fault. If the value is closed to 1, which represent the possibility of the fault is high.

In Table 4, the diagnosis results by using BPNN are obtained with the trained weighting coefficients w_{ij} and $w_{jk.}$ For fifteen diagnostic samples, there are two and one wrong diagnosis results by using BPNN and PNN, respectively. Although, there are several advantages of artificial neural network such as the weightings of neural network are obtained from neural computation and the diagnosis results by using artificial neural network are objective than traditional expert's experiences, the diagnosis results are not accurate completely when the diagnostic samples are not within the range of trained samples. Because the membership function is designed with subjective experiences, the diagnosis results by using BPNN with membership function could not obtained the accuracy results completely.

6.3 Bayesian Networks Diagnosis

According to Eq. (9), the diagnosis results by using BN are listed on Table 4. In BN classifier, for the three kinds of diagnostic samples, the posterior probabilities for gear faults are calculated and the decision is made for the maximum one. There is no wrong diagnosis result by using BN. It is accurate and easy to classify each gear fault for diagnostic samples by using BN method than both methods BPNN and PNN.

Table 4. The certainty of diagnosis result due to BN, BPNN and PNN

Diagnostic sample	Faults / Test No.	O_1			O_2			O_3		
		BN	BPNN	PNN	BN	BPNN	PNN	BN	BPNN	PNN
1	1▲	0.942	0.088	1	0.058	0.943	0	0	0.133	0
	2	0.971	0.582	1	0.029	0.068	0	0	0.177	0
	3	0.965	0.527	1	0.035	0.042	0	0	0.055	0
	4	0.982	0.971	1	0.018	0.030	0	0	0.031	0
	5	0.977	0.958	1	0.023	0.499	0	0	0.022	0
2	1▲	0.229	0.935	0	0.771	0.110	1	0	0.008	0
	2	0.109	0.047	0	0.891	0.830	1	0	0.001	0
	3*	0.001	0.230	1	0.999	0.525	0	0	0.013	0
	4	0.305	0.255	0	0.695	0.579	1	0	0.026	0
	5	0.145	0.270	0	0.855	0.575	1	0	0.006	0
3	1	0	0.029	0	0	0.024	0	1	0.981	1
	2	0	0.029	0	0	0.024	0	1	0.981	1
	3	0	0.029	0	0	0.024	0	1	0.981	1
	4	0	0.029	0	0	0.024	0	1	0.981	1
	5	0	0.029	0	0	0.024	0	1	0.981	1

BN: Bayesian networks, BPNN: Back-propagation neural network, PNN: Probabilistic neural network., ▲: Wrong result by using BPNN, ∗: Wrong result by using PNN.

7 Conclusions

This study used six statistical parameters of vibration signals in time-domain to fault diagnosis in gear train system. The diagnosis results by using BN are better accurate than BPNN and PNN in Table 4.

Besides, it spends little time to obtain the diagnosis results than BPNN and PNN, because the connections weightings between inputs and outputs in BN are probability computation which differs from BPNN and PNN. Thus, the fault diagnosis results in gear train system by using BN not only represent high accuracy but also have fewer calculations than both methods BPNN and PNN.

Acknowledgements

This study was supported by grant number NSC92-2623-7-033-003-ET from National Science Council and by the Center-of-Excellence Program on Membrane Technology from Ministry of Education, Taiwan, R.O.C.

References

1. Dyer, D., Stewart, R. M.: Detection of Rolling Element Bearing Damage by Statistical Vibration Analysis. Journal of Mechanical Design 100 (1978) 229-235
2. Rumelhart, D. E. and McClelland, J. L.: Parallel Distributed processing, Explorations in the Microstructure of Cognition, MIT Press (1986)

3. Sorsa, T., Koivo, H. N., Koivisto, H.: Neural Networks in Process Fault Diagnosis. IEEE Transactions on Systems 21 (1991) 815-825
4. Kang, Y., Wang, C. C., Chang, Y. P., Hsueh, C. C., Chang, M. C.: Certainty Improvement in Diagnosis of Multiple Faults by Using Versatile Membership Functions for Fuzzy Neural Networks. Lecture Notes in Computer Science 3973 (2006) 370-375
5. Specht, D.: Probabilistic Neural Networks for Classification, Mapping, or Associative Memory. IEEE International Conference on Neural Networks (1988) 525-532
6. Lin, W. M., Lin, C. H., Tasy, M. X.: Transformer-Fault Diagnosis by Integrating Field Data and Standard Codes with Training Enhancible Adaptive Probabilistic Network. IEE Proceedings Generation, Transmission and Distribution 152 (2005) 335-341
7. Kahn Jr., C. E., Laur, J. J., Carrera, G. F.: A Bayesian Network for Diagnosis of Primary Bone Tumors. Journal of Digital Imaging 14(2001) 56-57
8. Mast, T. A., Reed, A. T., Yurkovich, S., Ashby, M., Adibhatla, S.: Bayesian Belief Networks for Fault Identification in Aircraft Gas Turbine Engines. IEEE Conference on Control Applications 1(1999) 39-44
9. Romessis, C., Mathioudakis, K.: Bayesian Network Approach for Gas Path Fault Diagnosis. Journal of Engineering for Gas Turbines and Power 128(2006) 64-72
10. Chien, C. F., Chen, S. L., Lin, Y. S.: Using Bayesian Network for Fault Location on Distribution Feeder. IEEE Transactions on Power Delivery 17 (2002) 785-793
11. Giagopulos, D., Salpistis, C., Natsiavas, S.: Effect of Non-Linearities in the Identification and Fault Detection of Gear-Pair Systems. International Journal of Non-Linear Mechanics 41 (2006) 213-230
12. Heng, R. B. W., Nor, M. J. M.: Statistical Analysis of Sound and Vibration Signals for Monitoring Rolling Element Bearing Condition. Applied Acoustics 53(1998) 211-226
13. Berger, J. O.: Statistical Decision Theory and Bayesian Analysis, 2nd edition, New York, Springer-Verlag (1985)
14. Perrot, N., Trystram, G., Le, G. D., Guely, F.: Sensor Fusion for Real Time Quality Evaluation of Biscuit During Baking. Comparison Between Bayesian and Fuzzy Approaches. Journal of Food Engineering 29(1996) 301-315

An Intelligent Hybrid Algorithm for Job-Shop Scheduling Based on Particle Swarm Optimization and Artificial Immune System

Ge Hong-Wei[1], Du Wen-Li[1], Qian Feng[1,*], and Wang Lu[2]

[1] State-Key Laboratory of Chemical Engineering, China East China University of Science and Technology, Shanghai 200237, China
[2] College of Computer Science, Jilin University, Changchun 130012, China
fqian@ecust.edu.cn

Abstract. A computationally effective algorithm of combining PSO with AIS for solving the minimum makespan problem of job shop scheduling is proposed. In the particle swarm system, a novel concept for the distance and velocity of a particle is presented to pave the way for the job-shop scheduling problem. In the artificial immune system, the models of vaccination and receptor editing are designed to improve the immune performance. The proposed algorithm effectively exploits the capabilities of distributed and parallel computing of swarm intelligence approaches. The algorithm is examined using a set of benchmark instances with various sizes and levels of hardness and compared with other approaches reported in some existing literatures. The computational results validate the effectiveness of the proposed approach.

1 Introduction

The job shop scheduling problem (JSSP) is a very important practical problem in both fields of production management and combinatorial optimization. The JSSP has drawn the attention of researchers for the last three decades, but because scheduling problems vary widely according to specific production tasks and most of them are strongly NP-hard problems such that some test problems of moderate size are still unsolved. As a matter of fact, only small size instances of the problems can be solved within a reasonable computational time by exact optimization algorithm such as branch and bound [1] and dynamic programming (DP) [2]. By contrast, approximate and heuristic methods make a tradeoff between solution quality and computational cost. These methods mainly include dispatching priority rules [3], shifting bottleneck approach [4], tabu search [5] and have made considerable achievement. In recent years, much attention has been devoted to meta-heuristics with the emergence of new techniques from the field of artificial intelligence such as genetic algorithm (GA) [6], simulated annealing (SA) [7], artificial immune system (AIS) [8], and so on. Among the meta-heuristic algorithms, GA has been used with increasing frequency to address scheduling problems and may not remain much room for the improvement, however, the use of the particle swarm optimization (PSO) and AIS for the solution of

[*] Corresponding author.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 628–637, 2007.
springerlink.com

scheduling problems has been scarce, especially the PSO. Besides, many research results of the JSSP show that it is difficult to obtain a good-enough solution only by single search scheme. Motivated by these perspectives, we propose a novel hybrid algorithm for the job shop scheduling problem based on particle swarm optimization and artificial immune system in this paper.

2 Representation for JSSP and Initial Schedule Generation

The job shop problem studied in the paper consists of scheduling a set of jobs on a set of machines with the objective to minimize the makespan, i.e., the maximum of completion times needed for processing all jobs. Each machine can handle at most one job at a time. Each job consists of a chain of operations to be processed in a specified sequence, on specified machines, and during an uninterrupted time period.

Before solving the JSSP, we describe a proper representation for the solution of the problem, namely a scheduling. In this paper, an operation-based representation is adopted, which uses an unpartitioned permutation with *m*-repetitions of job numbers for the problems with *n* jobs and *m* machines. A job represents a set of operations that has to be scheduled on *m* machines. In this formulation, each job number occurs *m* times in the permutation. By scanning the permutation from left to right, the *k*-th occurrence of a job number refers to the *k*-th operation in the technological sequence of this job.

A good initial solution is fundamental for the computational performance of the search algorithm. In this paper, a new scheduling initialization algorithm is proposed based on the well-known algorithm of Giffler and Thompson to produce the active schedules. The brief outline of the algorithm is as follows:

Step 1: Let set A contain the first schedulable operation of each job. Let $s_{km} = 0$, for all the operation (k,m) in set A, where s_{km} is the earliest time at which the operation (k,m) can start.

Step 2: Calculate $t(A) = \min(s_{km} + p_{km})$, for all the operation $(k,m) \in A$,where p_{km} is the processing time of operation (k,m).

Step 3: Set up set M, for all operation $(k,m) \in A$, if $s_{km} < t(A)$, then put the machine m into set M.

Step 4: Set up set G, randomly select a machine m^* from set M, for all the operations $(k,m) \in A$, put the operation (k,m) into set G if it is processed on the machine m^*.

Step 5: For all the operations $(k,m^*) \in G$, calculate the earliest completed time of job k if the operation (k,m^*) is currently processed, let k^* denote the job which is completed lastly.

Step 6: Select the operation (k^*,m^*) from set G and append it to the schedule.

Step 7: Delete the operation (k^*,m^*) from set A, then subjoin its immediate successor into set A.

Step 8: Update s_{km} in set A and then return to step 2 until A is empty, namely all the operations are scheduled.

According to the algorithm proposed by Giffler and Thompson, the operation with an early start time and a minimum processing time possesses a high priority to be scheduled. In our algorithm, besides the above rule, we also consider the degree of the effect on the makespan when a schedulable operation is delayed in current schedule set. So we first need to calculate the earliest finishing time of each job in the current schedule set and find the job with the longest finishing time, the effect on the makespan is great when its corresponding current operation is delayed. So the corresponding current operation of the job has a high priority to be scheduled just as the Step 5 stated.

3 PSO-Based Scheduling Algorithm

The particle swarm optimization (PSO), originally developed by Kennedy and Elberhart [9], is a method for optimizing hard numerical functions based on swarm intelligence. A swarm consists of individuals, called particles, which change their positions over time. Each particle represents a potential solution to the problem. In a PSO system, particles fly around in a multi-dimensional search space. During its flight each particle adjusts its position according to its own experience and the experience of its neighboring particles. The effect is that particles move towards the better solution areas, while still having the ability to search a wide area. The performance of each particle is measured according to a pre-defined fitness function.

The conventional particle swarm optimization cannot be applied to the JSSP directly. In this section, we describe the formulations of the proposed PSO algorithm for the JSSP. Firstly, the concept of the difference of the locations is extended. Let us present the similarity measure of two particles. Denote the i th and the j th particles in a D -dimensional space as $X_i = (x_{i1},\dots,x_{id},\dots,x_{iD})$ and $X_j = (x_{j1},\dots,x_{jd},\dots,x_{jD})$, respectively. Define functions

$$s(k) = \begin{cases} 1 & if \quad x_{ik} = x_{jk} \\ 0 & if \quad x_{ik} \neq x_{jk} \end{cases} \quad \text{and} \quad S(X_i, X_j) = \sum_{k=1}^{D} s(k) . \tag{1}$$

where $S(X_i, X_j)$ is called the similarity measure between particle X_i and particle X_j. Let $f(X_i)$ denote the fitness of particle i and assume that $0 \le f(X_i) \le C$, $\forall i \in \{1,2,\cdots n\}$, where n is the population size. The difference of the locations between two particles, namely "distance", is redefined by the following equation:

$$dis(X_i - X_j) = k \cdot [\alpha \cdot | f(X_i) - f(X_j) | / C + \beta \cdot (D - S(X_i, X_j)) / D] . \tag{2}$$

where D is the dimension of a particle, k is a positive integer called acceleration coefficient, and α and β are two positive weights, which can be ascertained by trial and error and satisfy the relation that the sum of α and β is equal to 1. In this paper, α and β is taken as 0.7 and 0.3, respectively. It is obvious that $0 \le | f(X_i) - f(X_j) | / C \le 1$ and $0 \le (D - S(X_i, X_j)) / D \le 1$.

Correspondingly, the concept of the velocity is also extended. The velocity is defined as the times of “adjustment operation”. The brief outline of the adjustment algorithm is as follows.

Let $X=(x_1,\ldots,x_k,\ldots,x_D)$ and $Y=(y_1,\ldots,y_k,\ldots,y_D)$ be two particles in a D-dimensional space respectively.

Step 1: Select an index k of location randomly and set an indicator $m=0$, if $x_k=y_k=s$, where s is the number on location k, go to Step 2, else if $x_k \neq y_k$, namely $x_k=s$ and $y_k \neq s$, go to Step 3.

Step 2: Scan set $X'=\{x_i \mid 1 \le i < k\}$ and set $Y'=\{y_i \mid 1 \le i < k\}$ from left to right, respectively, to find out the times that s appears, and denote them as $X'\_times(s)=t_x$ and $Y'\_times(s)=t_y$, respectively. If $t_x > t_y$, go to Step 4; if $t_x < t_y$, go to Step 5; if $t_x = t_y$, go to Step 6.

Step 3: Select a location index j in the particle Y randomly, which satisfies $y_j=s$ and $y_j \neq x_j$. Swap y_j and y_k, and set the indicator $m=1$, then go to Step 2.

Step 4: Swap s which appears after location k and y_j which satisfies that $y_j \neq x_j$, $y_j \neq s$ and $j<k$ until $t_x=t_y$, then go to Step 7.

Step 5: Swap s which appears in the front of location k and y_j which satisfies that $y_j \neq x_j$, $y_j \neq s$ and $j>k$ until $t_x=t_y$, then go to Step 7.

Step 6: If $m=0$, reselect an index k randomly; if $m=1$, go to Step 7.

Step 7: Terminate.

Using “$\oplus$” to denote the adjustment operation, the proposed PSO algorithm could be performed by the following equations:

$$v_{id}=\text{int}[wv_{id}+c_1r_1dis(p_{id}-x_{id})+c_2r_2dis(p_{gd}-x_{id})], \tag{3}$$

$$x_{id}=x_{id}\oplus v_{id}. \tag{4}$$

where int[·] represents the truncation function and $dis(\cdot)$ can be calculated using Eq. (8). In this paper, the parameters are taken as $w=0.5$, $c_1=1.8$, and $c_2=1.0$.

4 AIS-Based Scheduling Algorithm

The artificial immune system (AIS) is rapidly emerging inspired by theoretical immunology and observed immune functions, principles, and models. The efficient mechanisms of immune system, including the clonal selection, learning ability, memory, robustness and flexibility, make artificial immune systems useful in many application fields. The AIS has appeared to offer powerful and robust information processing capabilities for solving complex problems. In this section, a new AIS-based scheduling algorithm is proposed. The proposed algorithm is built on the principles of clonal selection, affinity maturation and the abilities of learning and memory. The proportional model is adopted for the clonal selection in this paper. It is a direct ratio selection strategy and can be used to select a new population according to the direct ratio with respect to fitness values.

4.1 Vaccination

In this section, a novel vaccination operation is proposed. The regulatory mechanisms of the immune system make modulation of the immune response to antigens towards those responses that are protective against the pathogen. Similarly to the techniques of the vaccine inoculation, we take full advantage of some characteristic information and knowledge in the process of solving problem to distill vaccines, and then vaccinate some antibodies. The process of vaccination is designed as follows. First, the antibody with the highest affinity is recognized, which is called here the best antibody, and then an antibody is selected for vaccination. Produce a vector of length $j\times m$ for the problem of j jobs and m machines which is randomly filled with elements of the set {0, 1}. This vector defines the order in which the genes of the newly produced antibody are drawn from the best antibody and the selected antibody, respectively. After a gene is drawn from one antibody molecule and deleted from the other one, it is appended to the newly produced antibody molecule. This step is repeated until both the best and the selected antibody molecules are empty and the produced antibody contains all the genes involved. An offspring is created using the operator and accepted into next population if the affinity value is higher than the original.

4.2 Mutation

In immune systems, random changes take place in the variable region genes of antibody molecules. Mutations cause structurally different antibody molecules. In the artificial immune system for solving the JSSP, the number of mutational genes in an antibody molecule is inversely proportional to the antibody affinity. The number of mutational genes in an antibody molecule is decided by the following formulation

$$bit = \text{int}[\frac{(f_{\max} - f\ )(\rho - \eta)}{f_{\max} - f_{\min}}] + \eta \tag{5}$$

Where bit is the number of the mutational genes, $\text{int}[\cdot]$ is the truncation function, f is the affinity of the mutational antibody, $f_{\max}$ and $f_{\min}$ are respectively the maximal and the minimal affinity in the population, ρ and η are respectively the upper bound and the lower bound of the mutation bits. The mutation operations are completed by selecting bit positions from the antibody and permuting the genes on these positions randomly. The mutated antibody replaces its original regardless of its fitness value.

4.3 Receptor Editing

Due to the random nature of the mutation processes, a large proportion of mutating genes become non-functional or develop into harmful anti-self specificities. Those cells are eliminated by a programmed death process known as receptor editing. The proposed receptor editing model embodies the two operations of antibody introduction and gene shift. If there is no improvement of the highest affinity degree for a certain number of generations, the two operations are performed. The antibody introduction is that a certain percentage of antibodies are randomly produced by the proposed initialization algorithm in Section 2 and replace the worst antibodies of the whole population. The process of the gene shift is described as follows. Randomly select some antibodies, for every selected antibody, randomly select a gene of the antibody molecule, make it move backwards in turn and once shift one position until it moves λ times, and then λ new antibodies are produced. We replace the original antibody by the antibody with the highest affinity among the produced antibodies if it is better than the original.

4.4 AIS-Based Scheduling Algorithm

The outline of the proposed algorithm based on the AIS can be described as follows:

Step 1: Initialize *pop_size* antibodies as an initial population using the proposed initialization algorithm, where *pop_size* denotes the population size.
Step 2: Select *m* antibodies from the population by the proportional selection model and clone them to a clonal library.
Step 3: Perform the mutation operation for each of the antibodies in the clonal library.
Step 4: Randomly select *s* antibodies from the clonal library to perform the operation of the vaccination.
Step 5: Replace the worst *s* antibodies in the population by the best *s* antibodies from the clonal library.
Step 6: Perform the operation of receptor editing if there is no improvement of the highest affinity degree for a certain number of generations *G*.
Step 7: Stop if the termination condition is satisfied, else repeat Step 2-Step 7.

In this paper, parameters are taken as $pop\_size = 50$, $m = 30$, $s = 10$, $G = 80$.

5 Hybrid Algorithm Based on the Proposed PSO and AIS

Based on the models of the proposed PSO and AIS, a hybrid intelligence algorithm (HIA) is proposed to solve the JSSP. The objective is to search a scheduling such that

$$\min(T(JM)) = \min\{\max[T(1),T(2),\cdots,T(i),\cdots,T(m)]\} \tag{6}$$

where $T(i)$ is the final completion time of machining on the *i*th machine, $T(JM)$ is the final completion time of all the jobs.

The fitness and the affinity are evaluated using the following equation

$$f_i = 100 \times opt / T_i(JM) \tag{7}$$

Where f_i is the fitness value of the *i*th particle, and opt is the theoretical optimal makespan for a given problem and the value f_i is in the interval (0, 100].

The value of the objective function *T*(*JM*) can be obtained by the process of decoding. The makespan is produced by the process that assigns operations to the machines at their earliest possible starting time by the technological order of each job, scanning the permutation from left to right. Then the active decoding is applied, which checks the possible blank time interval before appending an operation at the last position, and fills the first blank interval before the last operation to convert the semi-active schedule to an active one so that the makespan can be shorter.

It is worth mentioning that the fitness and affinity is as pivotal to the guidance of the search and clonal selection because the chance of a solution being chosen is determined solely by its objective function value. So their values are linearly adjusted to avoid early convergence and ensure the variety of the population.

The flow chart of the hybrid algorithm is shown in Figure 1. Furthermore, the proposed algorithm stops if the best obtained solution equals the best known makespan or if for a given number *cnt* of generations there has been no improvement of the best solution found so far, where $cnt = 20000$.

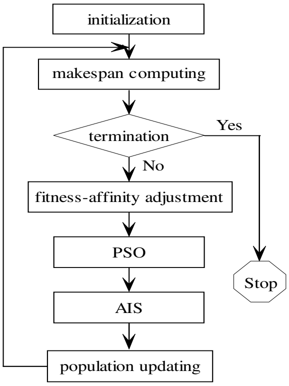


Fig. 1. The flow chart of the hybrid algorithm for the JSSP

6 Numerical Simulation Results and Comparisons

The performance of the proposed hybrid intelligence algorithm (HIA) for the JSSP is examined by using some test problems taken from the OR-Library [10]. Numerical experiments are performed in C++ language on a PC with Pentium IV 1.4 GHz processor and 256MB memory. The numerical results are compared with those reported in some existing literatures using other approaches [8, 11-14], including some heuristic and meta-heuristic algorithms.

Table 1 summarizes the results of the experiments. The contents of the table include the name of each test problem (Instance), the scale of the problem (Size), the value of the best known solution for each problem (BKS), the value of the best solution found by using the proposed algorithm (HIA), the number of running generations (Iterations), the percentage of the deviation with respect to the best known solution (RD%), and the best results reported in other literatures. It is worth mentioning that the makespan listed in the table is the best obtained from five executions by inching the defined fitness function (8), which is changed according to the following equation:

$$f_i = 100 \times (opt \pm \delta) / T_i(JM) \tag{8}$$

where δ is the tuning parameter and its value is taken as 0, 1 and 2 respectively. The proposed algorithm produces quite satisfactory solutions in reasonable amount of iterations by tuning only the parameter δ.

Table 1. Comparisons of the results between HIA and other approaches

Inst-ance	Size	BKS	HIA	Iterations	RD (%)	Goncalves Param.A [11]	Ombuki LSGA [12]	Coello AIS[8]	Aiex[13] GP+PR	Binato[14] GRASP
Ft06	6×6	55	**55**	3	0.00	55	-	-	55	55
Ft10	10×10	930	**930**	8704	0.00	930	-	941	930	938
Ft20	20×5	1165	**1165**	110944	0.00	1165	-	-	1165	1169
La05	10×5	593	**593**	2	0.00	593	-	593	593	593
La10	15×5	958	**958**	4	0.00	958	-	958	958	958
La15	20×5	1207	**1207**	97	0.00	1207	-	-	1207	1207
La16	10×10	945	**945**	2274	0.00	945	959	945	945	946
La17	10×10	784	**784**	716	0.00	784	792	785	784	784
La18	10×10	848	**848**	554	0.00	848	857	848	848	848
La19	10×10	842	**842**	1167	0.00	842	860	848	842	842
La20	10×10	902	**902**	229642	0.00	907	907	907	902	907
La21	15×10	1046	**1046**	143989	0.00	1046	1114	-	1057	1091
La22	15×10	927	**932**	47391	0.54	935	989	-	927	960
La23	15×10	1032	**1032**	15972	0.00	1032	1035	-	1032	1032
La24	15×10	935	**950**	18028	1.60	953	1032	-	954	978
La25	15×10	977	**979**	315935	0.20	986	1047	1022	984	1028
La26	20×10	1218	**1218**	215323	0.00	1218	1307	-	1218	1271
La27	20×10	1235	**1256**	114978	1.70	1256	1350	-	1269	1320
La28	20×10	1216	**1227**	198732	0.90	1232	1312	1277	1225	1293
La29	20×10	1152	**1184**	420597	2.78	1196	1311	1248	1203	1293
La30	20×10	1355	**1355**	119454	0.00	1355	1451	-	1355	1368
La31	30×10	1784	**1784**	1664	0.00	1784	1784	-	1784	1784
La32	30×10	1850	**1850**	6680	0.00	1850	1850	-	1850	1850
La33	30×10	1719	**1719**	142	0.00	1719	1745	-	1719	1719
La34	30×10	1721	**1721**	55065	0.00	1721	1784	-	1721	1753
La35	30×10	1888	**1888**	7979	0.00	1888	1958	1903	1888	1888
La36	15×15	1268	**1281**	56613	1.03	1279	1358	1323	1287	1334
La37	15×15	1397	**1415**	41241	1.72	1408	1517	-	1410	1457
La38	15×15	1196	**1213**	347419	1.42	1219	1362	1274	1218	1267
La39	15×15	1233	**1246**	276302	1.05	1246	1391	1270	1248	1290
La40	15×15	1222	**1240**	225832	1.47	1241	1323	1258	1244	1259

From the table it can be seen that the deviation of the minimum found makespan from the best known solution is only on average 0.46%. The proposed algorithm yields a significant improvement in solution quality with respect to almost all other algorithms. The superior results indicate the successful incorporation of the improved PSO and AIS, which facilitates the escape from local minimum points and increase the possibility of finding a better solution.

To illustrate the simulated results more intuitively, the notorious FT10 problem is specially described as an example. Figure 2 shows a group of convergence curves for different values of the parameter δ . Figure 3 shows the Gantt chart of a best solution.

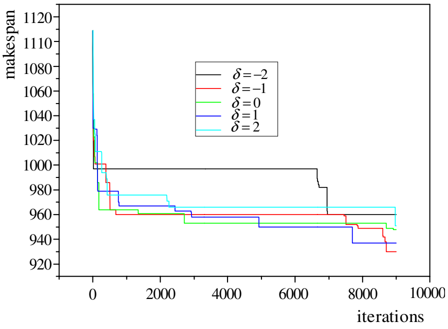


Fig. 2. Convergence curves for different values of the parameter δ

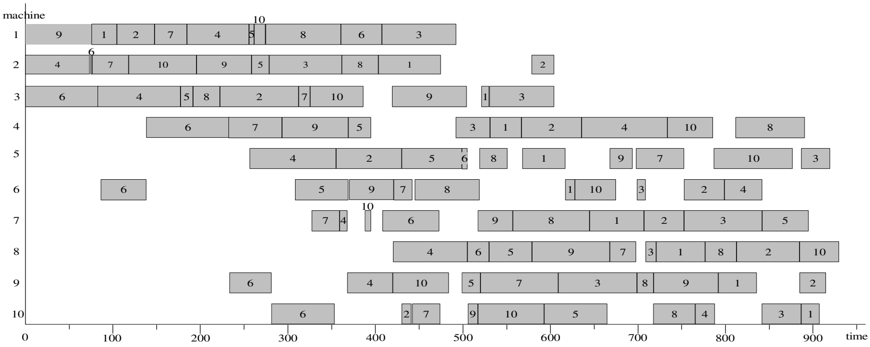


Fig. 3. The Gantt chart of an optimal schedule for FT10

7 Conclusions

A promising hybrid intelligence algorithm combing an improved PSO and AIS is proposed to solve job shop problems. In the PSO system, a novel concept for the distance and velocity of a particle is presented for the JSSP. In the AIS, the models of vaccination and receptor editing are designed to improve the immune performance. Besides, a new initialization algorithm based on G&T algorithm is proposed to produce the initial population in the PSO and AIS. The algorithm is tested on a set of 31 standard instances taken from the OR-Library. Computational results are compared with those obtained using other existing approaches and the proposed approach yields significant improvement in solution quality. The superior results indicate the successful incorporation of the two improved swarm intelligence approaches. The investigation on further study is to extend the proposed algorithm in order that it could be applied to more practical and integrated manufacturing problems.

Acknowledgments

The first three authors are grateful to the support of the national outstanding youth science fund (60625302), the National “973” Plan (2002CB3122000), the National “863” Plan (20060104Z1081), the Natural Science Fund of Shanghai (05ZR14038), and the Key Fundamental Research Program of Shanghai Science and Technology Committee (05DJ14002).

References

1. Heilmann R.: A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. European Journal of Operational Research, (2003), 144(2): 348-365.
2. Lorigeon T.: A dynamic programming algorithm for scheduling jobs in a two-machine open shop with an availability constraint. Journal of the Operational Research Society, (2002), 53(11): 1239-1246.

3. Canbolat Y.B. and Gundogar E.: Fuzzy priority rule for job shop scheduling. Journal of intelligent manufacturing, (2004), 15(4): 527-533.
4. Huang W.Q. and Yin A.H.: An improved shifting bottleneck procedure for the job shop scheduling problem. Computers & Operations Research, (2004), 31(12): 2093-2110.
5. Geyik F. and Cedimoglu I.H.: The strategies and parameters of tabu search for job-shop scheduling. Journal of intelligent manufacturing, (2004), 15(4): 439-448.
6. Fayad C. and Petrovic S.: A fuzzy genetic algorithm for real-world job shop scheduling. Lecture Notes in Computer Science, (2005), 3533: 524-533.
7. Suresh R.K. and Mohanasundaram K.M.: Pareto archived simulated annealing for job shop scheduling with multiple objectives. International Journal of Advanced Manufacturing Technology, (2006), 29(1-2): 184-196.
8. Coello C.A.C., Rivera D.C. and Cortes N.C.: Use of an artificial immune system for job shop scheduling. Lecture Notes in Computer Science, (2003), 2787: 1-10.
9. Kennedy J. and Eberhart R.: Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway, NJ, (1995), 4: 1942–1948.
10. Beasley J.E.: OR-Library: Distributing Test Problems by Electronic Mail. Journal of the Operations Research Society, (1990), 41(11): 1069-1072.
11. Goncalves J.F., Mendes J.J.D.M. and Resende M.G.C.: A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research, (2005), 167(1): 77-95.
12. Ombuki B.M. and Ventresca M.: Local search genetic algorithms for the job shop scheduling problem. Applied Intelligence, (2004), 21(1): 99-109.
13. Aiex R.M., Binato S. and Resende M.G.C.: Parallel GRASP with path-relinking for job shop scheduling. Parallel Computing, (2003), 29(4): 393-430.
14. Binato S., Hery W.J., Loewenstern D.M. and Resende M.G.C.: A GRASP for Job Shop Scheduling. Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, Boston, (2001), 59-80.

Fuzzy Multi-criteria Decision Making Method for Machine Selection

İrfan Ertuğrul[1] and Mustafa Güneş[2]

[1] Business Administration Department, Pamukkale University, 20017, Denizli, Turkey
iertugrul@pamukkale.edu.tr
[2] Industrial Engineering Department, King Abdulaziz University,
PO Box 80204 Jeddah 21589, Saudi Arabia
mgunes@kau.edu.sa

Abstract. The selection of appropriate machines is one of the most crucial decisions for a manufacturing company to develop an efficient production environment. The aim of this study is to propose a fuzzy approach for selecting the best machine. This paper is based on a fuzzy extension of the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) method. In this method, the ratings of various alternatives versus various subjective criteria and the weights of all criteria are assessed in linguistic variables represented by fuzzy numbers. Fuzzy numbers try to resolve the ambiguity of concepts that are associated with human being's judgments. To determine the order of the alternatives, closeness coefficient is defined by calculating the distances to the fuzzy positive ideal solution (FPIS) and fuzzy negative ideal solution (FNIS). By using fuzzy TOPSIS, uncertainty and vagueness from subjective perception and the experiences of decision maker can be effectively represented and reached to a more effective decision.

Keywords: Fuzzy logic, multi-criteria decision making, fuzzy TOPSIS, machine selection.

1 Introduction

A machine selection is an important decision making process for many manufacturing companies. Improperly selected machines can negatively affect the overall performance of a production system. Since the selection of machines is a time consuming and difficult process, requiring advanced knowledge and experience, that it may cause several problems for the engineers and managers [1]. The evaluation data of machine selection problem for various subjective criteria and the weights of the criteria are usually expressed in linguistic terms. Thus, in this paper a fuzzy TOPSIS method is proposed, where the ratings of various alternatives under various subjective criteria and the weights of all criteria are assessed in linguistic terms represented by fuzzy numbers.

In classical TOPSIS, the ratings and the weights of the criteria are known precisely. However, under many conditions, crisp data are inadequate to model real life situations since human judgments are often vague and decision makers cannot estimate their preferences with an exact numerical value [2]. In fuzzy TOPSIS the decision makers use the linguistic variables to asses the importance of the criteria and to evaluate the each alternative with respect to each criterion. These linguistic variables are converted into trapezoidal fuzzy numbers and fuzzy decision matrix is

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 638–648, 2007.
springerlink.com

formed. Then normalized fuzzy decision matrix and weighted normalized fuzzy decision matrix are formed. After FPIS and FNIS are defined, the closeness coefficient of each alternative is calculated. According to these values, decision maker can determine the order of the alternatives and can choose the best one.

The rest of the paper is organized as follows: in the following section, first, basic concepts such as fuzzy sets, fuzzy numbers are tried to be defined; then, in the third section fuzzy TOPSIS method is introduced. In Section 4, a numerical example of machine selection problem is illustrated. Lastly, the Section 5 concludes.

2 Fuzzy Sets and Fuzzy Numbers

In 1965 Lotfi A. Zadeh proposed a new approach to a rigorous, precise theory of approximation and vagueness based on generalization of standard set theory to fuzzy sets [3]. Fuzzy set is an extension of a crisp set. Crisp sets only allow full membership or no membership at all, whereas fuzzy sets allow partial membership. In other words, an element may partially belong to a fuzzy set. Zadeh [4], proposed to use values ranging from 0 to 1 for showing the membership of the objects in a fuzzy set. Complete non-membership is represented by 0, and complete membership as 1. Values between 0 and 1 represent intermediate degrees of membership [5]. The membership degree of the fuzzy set can be described with triangular, trapezoidal, Gaussian, sigmodial functions or different functions can be formed [6]. Variety of membership functions of fuzzy sets decrease the ambiguity in decision making [7].

A fuzzy number is a convex fuzzy set, characterized by a given interval of real numbers, each with a grade of membership between 0 and 1 [8]. It is possible to use different fuzzy numbers according to the situation. Generally in practice triangular and trapezoidal fuzzy numbers are used [9].

2.1 Trapezoidal Fuzzy Numbers

In applications it is often convenient to work with trapezoidal fuzzy numbers because of their computational simplicity, and they are useful in promoting representation and information processing in a fuzzy environment. In this study trapezoidal fuzzy numbers in the fuzzy TOPSIS is adopted.

Trapezoidal fuzzy numbers can be expressed as (n_1 ,n_2 ,n_3 ,n_4). A trapezoidal fuzzy number $\tilde{n}$ is shown in Fig. 1.

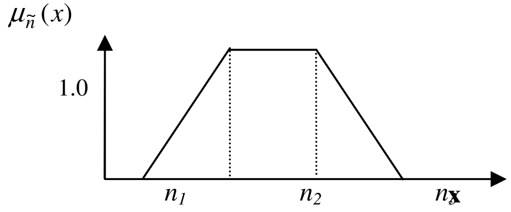


Fig. 1. A Trapezoidal fuzzy number, $\tilde{n}$

Trapezoidal fuzzy numbers are a special class of fuzzy numbers, defined by four real numbers, expressed as (n_1, n_2, n_3, n_4). Their membership functions are described as:

$$\mu_{\tilde{n}}(x) = \begin{cases} 0, & x < n_1 \\ \dfrac{x - n_1}{n_2 - n_1}, & n_1 \leq x \leq n_2 \\ 1, & n_2 \leq x \leq n_3 \\ \dfrac{x - n_4}{n_3 - n_4}, & n_3 \leq x \leq n_4 \\ 0, & x > n_4 \end{cases} \tag{1}$$

There are various operations on trapezoidal fuzzy numbers. But here, three important operations used in this study are illustrated. If we define, two positive trapezoidal fuzzy numbers $A = (m_1, m_2, m_3, m_4)$ and $B = (n_1, n_2, n_3, n_4)$ then

$$A \oplus B = (m_1 + n_1, m_2 + n_2, m_3 + n_3, m_4 + n_4) \tag{2}$$

$$A \otimes B = (m_1.n_1, m_2 n_2, m_3 n_3, m_4 n_4) \tag{3}$$

$$A \otimes k = (m_1 k, m_2 k, m_3 k, m_4 k) \tag{4}$$

(k is a positive real number)

The distance between two trapezoidal fuzzy numbers can be calculated by using Euclidean distance as [10]:

$$d_v(\tilde{m}, \tilde{n}) = \sqrt{\frac{(m_1 - n_1)^2 + 2(m_2 - n_2)^2 + 2(m_3 - n_3)^2 + (m_4 - n_4)}{6}} \tag{5}$$

3 Fuzzy TOPSIS Method

In this paper, the extension of TOPSIS method is considered which was originally proposed by Chen [11]. But differently from Chen's approach, in this study the distance between two fuzzy numbers is calculated by Euclidean distance.

Table 1. Linguistic variables for importance weight of each criterion

Linguistic Variables	Trapezoidal Fuzzy Numbers
Very Low (VL)	(0, 0, 0.1, 0.2)
Low (L)	(0.1, 0.2, 0.2, 0.3)
Medium Low (ML)	(0.2, 0.3, 0.4, 0.5)
Medium (M)	(0.4, 0.5 ,0.5, 0.6)
Medium High (MH)	(0.5, 0.6, 0.7, 0.8)
High (H)	(0.7, 0.8, 0.8, 0.9)
Very High (VH)	(0.8, 0.9, 1,1)

Table 2. Linguistic variables for ratings

Linguistic Variables	Trapezoidal Fuzzy Numbers
Very Poor (VP)	(0, 0, 1, 2)
Poor (P)	(1, 2, 2, 3)
Medium Poor (MP)	(2, 3, 4, 5)
Fair (F)	(4, 5, 5, 6)
Medium Good (MG)	(5, 6, 7, 8)
Good (G)	(7, 8, 8, 9)
Very Good (VG)	(8, 9, 10, 10)

In this study the importance weights of various criteria and ratings of qualitative criteria are considered as linguistic variables. The decision makers use the linguistic variables shown in Table 1 and Table 2 to evaluate the importance of the criteria and the ratings of alternatives with respect to criteria.

The fuzzy TOPSIS method can be described by the help of following sets [12]:

- a set of K decision-makers called $E = \{D_1, D_2, ..., D_K\}$;
- a set of m possible alternatives called; $A = \{A_1, A_2, ..., A_m\}$;
- a set of n criteria , $C = \{C_1, C_2, ..., C_n\}$, with which alternative performances are measured;
- a set of performance ratings of $A_i = \{i = 1,2,.....,m\}$ with respect to criteria $C_j = \{j = 1,2,..........,n\}$ called $X = \{x_{ij}, i = 1,2,.........m, j = 1,2,.......,n\}$.

In a decision committee that has K decision makers; fuzzy rating of each decision maker $D_k = (k = 1,2,.......,K)$ can be represented as trapezoidal fuzzy number $\tilde{R}_k = (k = 1,2,.....K)$ with membership function $\mu_{\tilde{R}_k}(x)$.

Let the fuzzy ratings of all decision-makers be trapezoidal fuzzy numbers $\tilde{R}_k = (a_k, b_k, c_k, d_k)$, $k = 1,2,......,K.$ Then the aggregated fuzzy rating can be defined as

$\tilde{R} = (a, b, c, d), \qquad k = 1,2,....,K$ where;

$$a = \min_k \{a_k\}, \qquad b = \frac{1}{K}\sum_{k=1}^{K} b_k , \qquad c = \frac{1}{K}\sum_{k=1}^{K} c_k , \quad d = \max_k \{d_k\} \tag{6}$$

Let the fuzzy rating and importance weight of the kth decision maker be $\tilde{x}_{ijk} = (a_{ijk}, b_{ijk}, c_{ijk}, d_{ijk})$ and $\tilde{w}_{ijk} = (w_{jk1}, w_{jk2}, w_{jk3}, w_{jk4})$, $i = 1,2,..........m$, $j = 1,2,..........n$ respectively. Then the aggregated fuzzy ratings $(\tilde{x}_{ij})$ of alternatives with respect to each criterion can be calculated as

$$(\tilde{x}_{ij}) = (a_{ij}, b_{ij}, c_{ij}, d_{ij})$$

$$a_{ij} = \min_k \{a_{ijk}\}, \qquad b_{ij} = \frac{1}{K}\sum_{k=1}^{K} b_{ijk} , \quad c_{ij} = \frac{1}{K}\sum_{k=1}^{K} c_{ijk} , \quad d_{ij} = \max_k \{d_{ijk}\} \tag{7}$$

The aggregated fuzzy weights $(\tilde{w}_{ij})$ of each criterion can be calculated as:

$$(\tilde{w}_j) = (w_{j1}, w_{j2}, w_{j3}, w_{j4}), \tag{8}$$

where

$$w_{j1} = \min_k \{w_{jk1}\}, \quad w_{j2} = \frac{1}{K}\sum_{k=1}^{K} w_{jk2} , \quad w_{j3} = \frac{1}{K}\sum_{k=1}^{K} w_{jk3} , \quad w_{j4} = \max_k \{w_{jk4}\}$$

Then machine selection problem can be expressed in matrix format as follows:

$$\tilde{D}=\begin{bmatrix}\tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1n}\\ \tilde{x}_{21} & \tilde{x}_{22} & \cdots & \tilde{x}_{2n}\\ \vdots & \vdots & \cdots & \vdots\\ \tilde{x}_{m1} & \tilde{x}_{m2} & \cdots & \tilde{x}_{mn}\end{bmatrix},$$

$$\tilde{W}=[\tilde{w}_1, \tilde{w}_2, \cdots \tilde{w}_n]$$

where $\tilde{x}_{ij}=(a_{ji}, b_{ij}, c_{ij}, d_{ij})$ and $\tilde{w}_j=(w_{j1}, w_{j2}, w_{j3}, w_{j4})$; $i=1,2,....m,\quad j=1,2,....n$ can be approximated by positive trapezoidal fuzzy numbers.

To avoid the complicated normalization formula used in classical TOPSIS, the linear scale transformation can be used to transform the various criteria scales into a comparable scale. Therefore, it is possible to obtain the normalized fuzzy decision matrix denoted by [2]:

$$\tilde{R}=\left[\tilde{r}_{ij}\right]_{mxn} \tag{9}$$

where B and C are the set of benefit criteria and cost criteria respectively:

$$\tilde{r}_{ij}=\left(\frac{a_{ij}}{d_j^*}, \frac{b_{ij}}{d_j^*}, \frac{c_{ij}}{d_j^*}, \frac{d_{ij}}{d_j^*}\right), \quad j\in B;$$

$$\tilde{r}_{ij}=\left(\frac{a_j^-}{d_{ij}}, \frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{a_{ij}}\right), \quad j\in C;$$

$$c_j^*=\max_i d_{ij}, \quad \text{if } j\in B;$$

$$a_j^-=\min_i a_{ij}, \quad \text{if } j\in C.$$

The normalization method mentioned above is designed to preserve the property in which the elements $r_{ij}, \forall i, j$ are normalized trapezoidal fuzzy numbers.

Considering the different importance of each criterion, the weighted normalized fuzzy-decision matrix is formed as:

$$\tilde{V}=\left[\tilde{v}_{ij}\right]_{mxn} \quad i=1,2,.....,m \quad j=1,2,......,n \tag{10}$$

where, $\tilde{v}_{ij}=\tilde{r}_{ij}(.)\tilde{w}_j$.

According to the weighted normalized fuzzy-decision matrix, normalized positive trapezoidal fuzzy numbers can also approximate the elements $\tilde{v}_{ij}, \forall i, j$. Then, the fuzzy positive ideal solution (FPIS, A^*) and fuzzy negative ideal solution (FNIS, A^-) can be defined as:

$$A^*=\left(\tilde{v}_1^*, \tilde{v}_2^*, ...\tilde{v}_n^*\right), \tag{11}$$

$$A^-=\left(\tilde{v}_1^-, \tilde{v}_2^-, ...\tilde{v}_n^-\right), \tag{12}$$

where

$$\tilde{v}_j^*=\max_i\left\{v_{ij3}\right\} \text{ and } \tilde{v}_j^-=\min_i\left\{v_{ij1}\right\} \quad i=1,2,..........m, \ j=1,2,..........n.$$

The distance of each alternative from A^* and A^- can be calculated as:

$$d_i^* = \sum_{j=1}^{n} d_v(\tilde{v}_{ij}, \tilde{v}_j^*) \quad i = 1,2,..........m \tag{13}$$

$$d_i^- = \sum_{j=1}^{n} d_v(\tilde{v}_{ij}, \tilde{v}_j^*), \qquad i = 1,2,..........m \tag{14}$$

where $d_v(.,.)$ is the distance measurement between two fuzzy numbers.

A closeness coefficient (CC_i) is defined to determine the order of all possible alternatives. The closeness coefficient represents the distances to the fuzzy positive ideal solution (A^*) and fuzzy negative ideal solution (A^-) simultaneously. The closeness coefficient of each alternative is calculated as:

$$CC_i = \frac{d_i^-}{d_i^* + d_i^-}, \qquad i = 1,2,..........m \tag{15}$$

It is clear that $CC_i = 1$ if $A_i = A^*$ and $CC_i = 0$ if $A_i = A^-$. The order of all alternatives can be determined according to the descending order of CC_i. [12]

4 Numerical Example

Suppose that a textile firm desires to select best printing machine from three alternatives (A_1, A_2, A_3). After the information is taken from three decision makers (D_1, D_2, D_3) who are familiar with the machine properties, the hierarchy for the selection of the best machine is formed as in Fig. 2.

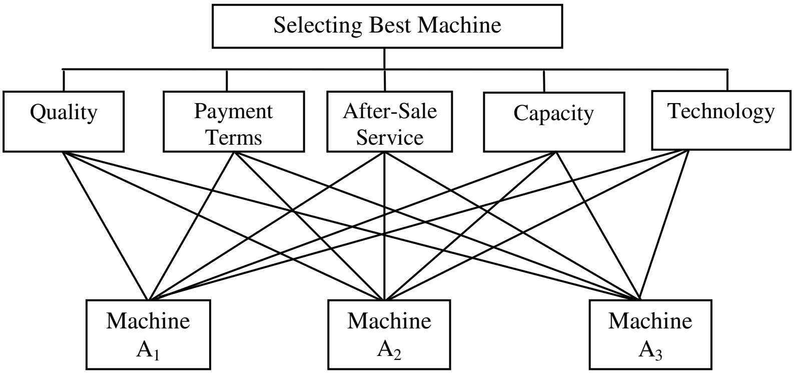


Fig. 2. Hierarchical Structure of Best Machine Selection Process

There are five criteria in the hierarchy. These are quality, payment terms, after-sale service, capacity and technology.

Decision makers use the linguistic variables in Table 1 to evaluate the importance of criteria. The importance weights of the criteria determined by these three decision makers are shown in Table 3.

Table 3. Importance weight of criteria from three decision-makers

Criteria	Decision-Makers		
	D_1	D_2	D_3
C_1	VH	VH	VH
C_2	MH	MH	H
C_3	H	H	H
C_4	MH	H	MH
C_5	VH	H	VH

Three decision makers use the linguistic variables shown in Table 2 to evaluate the ratings of candidates with respect to each criterion. The ratings of three candidates under five criteria are shown in Table 4.

Table 4. Ratings of the three candidates by decision-makers under five criteria

Criteria	Alternatives	Decision-Makers		
		D_1	D_2	D_3
C_1	A_1	G	G	MG
	A_2	VG	G	VG
	A_3	MG	G	G
C_2	A_1	F	F	MG
	A_2	MG	MG	MG
	A_3	VG	G	VG
C_3	A_1	G	G	MG
	A_2	G	G	G
	A_3	MG	F	F
C_4	A_1	MG	MG	MG
	A_2	G	G	VG
	A_3	VG	VG	VG
C_5	A_1	G	G	VG
	A_2	VG	G	VG
	A_3	MG	F	MP

Then linguistic variables shown in Table 3 and 4 are converted into trapezoidal fuzzy numbers to form fuzzy decision matrix as shown in Table 5.

The normalized fuzzy decision matrix is formed as in Table 6. Then weighted normalized fuzzy decision matrix is formed as in Table 7.

Table 5. Fuzzy decision matrix and fuzzy weights of three candidates

	A_1	A_2	A_3	Weight
C_1	(5, 7.3, 7.7, 9)	(7, 8.7, 9.3, 10)	(5, 7.3, 7.7, 9)	(0.8, 0.9,1,1)
C_2	(4, 5.3, 5.7, 8)	(5, 6, 7, 8)	(7, 8.7, 9.3, 10)	(0.5, 0.67, 0.73, 0.9)
C_3	(5, 7.3, 7.7, 9)	(7, 8, 8, 9)	(4, 5.3, 5.7, 8)	(0.7, 0.8, 0.8, 0.9)
C_4	(5, 6, 7, 8)	(7, 8.3, 8.7, 10)	(8, 9, 10, 10)	(0.5,0.67,0.73,0.9)
C_5	(7, 8.3, 8.7, 10)	(7, 8.7, 9.3, 10)	(2, 4.7, 5.3, 8)	(0.7, 0.87, 0.93,1)

Table 6. Normalized fuzzy decision matrix

	A_1	A_2	A_3
C_1	(0.5, 0.73, 0.77, 0.9)	(0.7, 0.87, 0.93, 1)	(0.5, 0.73, 0.77, 0.9)
C_2	(0.4, 0.53, 0.57, 0.8)	(0.5, 0.6, 0.7, 0.8)	(0.7, 0.87, 0.93, 1)
C_3	(0.5, 0.73, 0.77, 0.9)	(0.7, 0.8, 0.8, 0.9)	(0.4, 0.53, 0.57, 0.8)
C_4	(0.5, 0.6, 0.7, 0.8)	(0.7, 0.83, 0.87, 1)	(0.8, 0.9, 1, 1)
C_5	(0.7, 0.83, 0.87, 1)	(0.7, 0.87, 0.93, 1)	(0.2, 0.47, 0.53, 0.8)

Table 7. Weighted normalized fuzzy decision matrix

	A_1	A_2	A_3
C_1	(0.4, 0.66, 0.77,0.9)	(0.56, 0.78, 0.93, 1)	(0.4, 0.66, 0.77, 0.9)
C_2	(0.2, 0.36, 0.42, 0.72)	(0.25, 0.4, 0.51, 0.72)	(0.35, 0.58, 0.68, 0.9)
C_3	(0.35, 0.58, 0.62, 0.81)	(0.49, 0.64, 0.64, 0.81)	(0.28, 0.42, 0.46, 0.72)
C_4	(0.25, 0.4, 0.51, 0.72)	(0.35, 0.56, 0.64, 0.9)	(0.4, 0.6, 0.73, 0.9)
C_5	(0.49, 0.72, 0.81, 1)	(0.49, 0.76, 0.86, 1)	(0.14, 0.41, 0.49, 0.8)

After forming weighted normalized fuzzy decision matrix fuzzy positive ideal solution (FPIS) and fuzzy negative ideal solution (FNIS) are determined as;

$$A^* = [(1,1,1,1), (0.9,0.9,0.9,0.9), (0.81,0.81,0.81,0.81), (0.9,0.9,0.9,0.9), (1,1,1,1)]$$

$$A^- = [(0.4,0.4,0.4,0.4),(0.2,0.2,0.2,0.2),(0.28,0.28,0.28,0.28),(0.25,0.25,0.25,0.25)$$
$$(0.14,0.14,0.14,0.14)]$$

Then the distance of each alternative from FPIS and FNIS with respect to each criterion are calculated by using Euclidean distance as;

$$d(A_1, A^*) = \sqrt{\frac{(1-0.4)^2 + 2(1-0.66)^2 + 2(1-0.77)^2 + (1-0.9)^2}{6}} = 0.34$$

$$d(A_1, A^-) = \sqrt{\frac{(0.4-0.4)^2 + 2(0.4-0.66)^2 + 2(0.4-0.77)^2 + (0.4-0.9)^2}{6}} = 0.33$$

Here only the calculation of the distance of the first alternative to FPIS and FNIS for the first criterion is shown as the calculations are similar in all steps. The results of all alternatives' distances from FPIS and FNIS are shown in Table 8 and 9.

Table 8. Distances between $A_i\,(i = 1,2,3)$ and A^* with respect to each criterion

	$d(A_1, A^*)$	$d(A_2, A^*)$	$d(A_3, A^*)$
C_1	0.34	0.22	0.34
C_2	0.51	0.46	0.32
C_3	0.25	0.19	0.37
C_4	0.46	0.34	0.28
C_5	0.29	0.26	0.58

Table 9. Distances between $A_i\,(i = 1,2,3)$ and A^- with respect to each criterion

	$d(A_1, A^-)$	$d(A_2, A^-)$	$d(A_3, A^-)$
C_1	0.33	0.45	0.33
C_2	0.26	0.30	0.46
C_3	0.34	0.37	0.22
C_4	0.26	0.39	0.44
C_5	0.64	0.67	0.37

d_i^* and d_i^- of three alternatives are shown in Table 10. Then closeness coefficient of three alternatives are calculated as

$$CC_1 = \frac{1.83}{1.85+1.83} = 0.50 \qquad CC_2 = \frac{1.47}{2.19+1.47} = 0.60$$

$$CC_3 = \frac{1.82}{1.89+1.82} = 0.49$$

Table 10. Computations of d_i^*, d_i^- and CC_i

	A_1	A_2	A_3
d_i^*	1.85	1.47	1.89
d_i^-	1.83	2.19	1.82
$d_i^*+d_i^-$	3.68	3.66	3.72
CC_i	0.50	0.60	0.49

According to the closeness coefficient of three alternatives it can be said that the order of three alternatives is $A_2 > A_1 > A_3$. The firm will choose the Alternative 2 as its closeness coefficient has the highest value. In other words, the second alternative is closer to the FPIS and farther from the FNIS.

5 Conclusions

Decision-making process is getting harder in today's complex environment. Decision-makers face up to the uncertainty and vagueness from subjective perceptions and experiences in the decision-making process. Multi-criteria decision systems need experts in different areas. Fuzzy decision making theory can be used in many decision making areas like that.

The aim of this study is to propose fuzzy TOPSIS approach for selecting the best machine for a firm. Quality, payment term, after-sale service, capacity and technology factors were evaluated to obtain the preference degree associated with each machine alternative for selecting the most appropriate one. By the help of the fuzzy approach, the ambiguities involved in the assessment data could be effectively represented and processed to make a more effective decision.

As a result of the fuzzy TOPSIS method Alternative 2 is the best machine as its closeness coefficient is the highest. There are other methods that are used in comparing machines, such as AHP, ELECTRE. These methods have been recently used in a fuzzy environment. Further research may be the application of these methods to the machine selection problem.

In this study, the distance between two fuzzy numbers is calculated by using Euclidean distance. In future studies, other methods like vertex and Minkowski, Hamming, Chebyshev distance can be used in calculating the distance between two fuzzy numbers. And the results can be compared.

References

[1] Arslan, Ç., Çatay, B., Budak, E.: A Decision Support System For Machine Tool Selection. Journal of Manufacturing Technology Management, **15** (1) (2004) 101-109.

[2] Saghafian, S., Hejazi, A.R.: Multi-criteria Group Decision Making Using a Modified Fuzzy TOPSIS Procedure. Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and Conference Intelligent Agents, Web Technologies and Internet Commerce (2005)

[3] Ertuğrul, İ.: Bulanık Mantık ve Bir Üretim Planlamasında Uygulama Örneği. Master thesis, Pamukkale University (1996)

[4] Zadeh, L.A.: Fuzzy Sets. Information and Control **8** (1965) 338-353.

[5] Ertuğrul, İ., Karakaşoğlu, N.: The Fuzzy Analytic Hierarchy Process for Supplier Selection and an Application in a Textile Company. Proceedings of 5th International Symposium on Intelligent Manufacturing Systems (2006) 195-207.

[6] Başlıgil, H.: The Fuzzy Analytic Hierarchy Process for Software Selection Problems. Journal of Engineering and Natural Sciences. **2** (2005) 24-33.

[7] Güneş, M., Yiğitbaşı, O.N.: Türk Vergi Sisteminde Bulanık Mantık Uygulamaları. 5. Ulusal Ekonometri ve İstatistik Sempozyumu, Çukurova (2001). (www.ceterisparibus.net/kongre/cukurova_5htm)

[8] Deng, H., Multicriteria Analysis with Fuzzy Pairwise Comparison. International Journal of Approximate Reasoning. **21** (1999) 215-231.

[9] Baykal, N., Beyan, T.: Bulanık Mantık İlke ve Temelleri. Bıçaklar Kitabevi İstanbul (2004)

[10] Li, D.F: Compromise Ratio Method for Fuzzy Multi-Attribute Group Decision Making. Applied Soft Computing (2006) Article in press.

[11] Chen, C.T.: Extensions of the TOPSIS for Group Decision-Making under Fuzzy Environment. Fuzzy Sets and Systems **114** (2000) 1-9.

[12] Chen, C.T., Lin, C.T. Huang, S.F.: A Fuzzy Approach for Supplier Evaluation and Selection in Supply Chain Management", International Journal of Production Economics **102** (2006) 289–301.

Fuzzy Goal Programming and an Application of Production Process

İrfan Ertuğrul[1] and Mustafa Güneş[2]

[1] Business Administration Department, Pamukkale University, 20017, Denizli, Turkey
iertugrul@pamukkale.edu.tr
[2] Industrial Engineering Department, King Abdulaziz University, PO Box 80204 Jeddah 21589, Saudi Arabia
mgunes@kau.edu.sa

Abstract. Nowadays, information and advanced technology are commonly used. It is certain that the societies which benefit from information and technology will be more successful in this world where global competition is intensive. So managers must decide fast and right in the global competition. The way to decide fast and right is to benefit from the scientific methods that increase options and reduce uncertainty. Recently it has seen that decisions are mostly being made, in one sense in fuzzy. So the importance of fuzzy goal programming, one of operation research models including fuzzy, has been increasing day by day. The aim of this article is to examine fuzzy goal programming model which is one of the models providing the best decision-making under fuzzy. The article, in the direction of this aim, consists of five parts. In the first part, there is a general introduction about the subject. In the second part, it's mentioned about the fuzzy sets and the basic concepts of the membership function. In the third part fuzzy goal programming that has common usage area is explained. In the fourth part, fuzzy goal programming has applied to a firm with profit and sale goals. In the last part, conclusions and findings have been interpreted.

Keywords: Fuzzy set theory, multi criteria decision making, fuzzy goal programming.

1 Introduction

Fuzzy set theory is proposed for developing too simplified models and by this way analyzing complicated systems of the real world [1]. Moreover, it helps to decision maker not only in evaluating alternatives under the given constraints but also in developing new alternatives [2]. Fuzzy set theory has provided the real world to be expressed by mathematics. In this manner by going beyond the boundaries that classic mathematics created, it has provided uncertainty to take place in decision making processes. By the widespread use of fuzzy set theory nearly in every field of science and technology, even ordinary people has found themselves in their daily life surrounded with industrial products that had been produced with this methodology and began to use electronical devices whose names begin with "fuzzy". The studies that are such widespread in practice has extended field of effect of the classical quantitative methods' studies by the new approaches about the decision making in industrial systems. Since Zadeh has proposed the fuzzy set theory, it has been used in fields of

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 649–659, 2007.
springerlink.com

quantitative methods, management sciences, control theory, systems with artificial intelligence and human behaviours [1].

2 Fuzzy Sets and Membership Function

A fuzzy set is a set that has flexible boundaries. Fuzzy set theory is based on generalization of standard set theory and it accepts any value in the [0,1] interval for set membership. In fuzzy sets membership degree of an object is defined with a number between 0 and 1. Here, complete non-membership is represented by 0, and complete membership by 1. And values between these two numbers show the membership degree of the object in the set or the partial membership [3].

Although special algorithms are developed for membership functions in fuzzy set theory, in a lot of applications membership functions that can be expressed parametrically are used for their ease of computation. Determining the appropriate membership functions for the applications has a significant place in fuzzy set theory. Because, membership functions form the basis of fuzzy set theory [4]. Variety of membership functions of fuzzy sets decrease the ambiguity in decisions of managers. Fuzzy set theory can be applied to a lot of field of quantitative methods like linear programming, goal programming, multi-objective decision making, dynamic programming, queuing models, transportation models, game theory and network analysis [5].

3 Fuzzy Goal Programming

3.1 Decision Making in a Fuzzy Environment

Traditional decision making problem consists of six components. These components are respectively, decision maker, objective, decision criteria, alternatives, events and outcome. Here goal component can be defined as a maximization or minimization operation. Benefit, profit, income and cost functions form decision criteria. A universal set can be accepted as alternatives set. Restrictive conditions determine the circumstances. From this point of view, goals that are determined by the decision maker by taking the current condition or restrictive situations into consideration are the basis of decision making problems. Decision making in a fuzzy environment can be defined by the components mentioned above. Here, it is accepted as decision maker and alternative sets do not involve fuzziness. Goal and decision criteria components can involve fuzziness.

Decision maker can determine the aspiration level that he wants to reach for the goal function as fuzzy. Also parameter values of the function that indicate the decision criterion can be defined with fuzzy numbers. Goal and decision criterion components that complete each other can be considered as a fuzzy goal. On the other hand, parameter values of the constraints and /or right side constants can be fuzzy. In bigger and equal, equal, smaller and equal relations some tolerances can be allowed. So, cases in fuzzy environment can be taken as fuzzy constraints. It is unavoidable that the decisions which are given by fuzzy goals and /or fuzzy constraints will be fuzzy. A fuzzy decision is defined as a set that is formed by compromising given goal and constraints. Fuzzy decision set which is a subset of fuzzy goal and fuzzy constraints, indicates the degree of satisfaction of fuzzy constraint and fuzzy goal at the same time [6].

3.2 Goal Programming Model

Goal programming is one of the first business sciences that consider multi-objective decision making problems. This approach is firstly proposed by Charnes, Cooper and Ferguson in 1955. By taking different approaches and algorithms as a basis it has become to an extensive subject of study [7].

Goal programming model is a type of multiple goal programming. In multiple goal programming models the aim is determining a solution vector that satisfies mutually contradictory goals at the same time according to constraint sets. But in goal programming a solution that satisfies decision maker is tried to be determined. For this reason it can be said that goal programming model is based on satisfaction idea more than optimization idea.

Goal programming model can be examined in two sections as constraint set and objective function. All functions in a linear programming model form only the constraint set of goal programming model. In goal programming model, decision maker should determine aspiration value of objective functions. As a result of this, objective functions that have aspiration value are added to the constraint set as equality. This operation requires definition of deviation value for each goal function. Deviation values used in measuring the distance of objective functions from aspiration level. In goal programming model, deviations from the determined aspiration level for goals are minimized [8].

Goal programming model can be considered as two types according to goals' priority. First of them is the goal programming model that includes the same preference scale. Here the relative importance of goals is equal to each other and all goals are tried to be satisfied at the same time. The second one is the goal programming model with preference priority that includes the different preference features. Here, a hierarchic model is formed by the decision maker and the goals should be arranged from the most important one to less important one. This arrangement process can be made by qualitatively and also can be made quantitatively by using the weight concept. Goal programming models where all the goals are in the same preference priority and weighted goal programming models can be solved by simplex method [9].

3.3 Fuzzy Goal Programming Model

In goal programming model, objective functions, aspiration levels of them and constraints can be expressed as deterministically. To determine aspiration value of goals, the sequence of goals with preference priority and weights of goals is a difficult task [10].

Aspiration values, the preference priority of goals and relative weights are usually determined on the basis of decision maker's subjective judgment. The subjectivity in goal programming model can be considered as fuzzy set theory. When the fuzzy set theory is applied to goal programming model, aspiration values of goals and preference priorities can be described with uncertain expressions. Fuzzy set theory provides to define aspiration levels like "approximately equal to", "quite smaller than" for the objectives of decision makers that are based on subjective judgments. These kinds of definitions are considered as membership functions in fuzzy sets. By this way, it is emphasized that goal programming model is based on satisfaction idea rather than optimization idea. According to the priority structure of goals, goal programming model can be considered as two ways. The first one is fuzzy goal programming model. In this model, a solution is determined that satisfy all goals at the same time. The second one is fuzzy goal programming model with preference priority. In this model it is tried to find a solution that takes the priority preference of decision makers into consideration.

With the assumption of aspiration levels that are determined for goals are fuzzy; a generalized fuzzy goal programming model is expressed as [8]:

$$\left.\begin{array}{l} (Ax)_i = b_i \,;\ i= 1,2,\ldots,m_1 \\ (Ax)_i \overset{\sim}{\geq} b_i \,;\ i= m_1+1,\ \ldots,m_2 \\ (Ax)_i \geq b_i \,;\ i= m_2+1,\ \ldots,m_3 \end{array}\right\} \quad \text{Fuzzy Goals}$$

$$\left.\begin{array}{l} (Ax)_l\{=,\leq,\geq\}\ b_l \,;\ l=1,2,\ldots,p \\ x_j \geq 0 \,;\ j=1,2,\ldots,n \end{array}\right\} \quad \text{Non-fuzzy Constraints}$$

Here $\eqsim$, ≥ symbols are respectively the fuzzy versions of $=, \leq, \geq$ symbols. In this model, the aspiration value that is determined by decision maker for the i *th* goal is shown by $b_{i.}$.

The main difference between traditional set theory and fuzzy set theory is membership functions. While a traditional set is described with only one membership function, a fuzzy set can be described with infinite number of membership function. The membership functions of fuzzy sets can be classified as discontinuous and continuous, parametric and non-parametric, symmetric and asymmetric. Fuzzy goals are described as triangular, isosceles, partial linear, concave partial linear, semi-concave partial linear, s-shaped partial linear and convex partial linear membership functions in literature. These membership functions are usually formed by interviewing with decision maker. Membership function of a fuzzy goal can be formed by intuition on the basis of concepts' meaning in application. And, as membership functions form the basis of fuzzy set theory, after membership functions are determined it said that there is nothing fuzzy in the fuzzy set theory.

In many of the solution approaches that are developed for fuzzy goal programming, fuzzy goals are described by Zimmermann type membership function. Zimmermann type membership functions for fuzzy goals can be expressed as:

$$\begin{array}{l} (Ax)_i \eqsim b_i \;\Rightarrow\; \mu_i(x) = \\ \\ (i= 1,2,\ldots,m_1) \end{array} \begin{cases} 0 & ;\ \text{if } (Ax)_i \leq b_i-d_i \\ 1- \dfrac{b_i-(Ax)_i}{d_i} & ;\ \text{if } b_i-d_i \leq (Ax)_i \leq b_i \\ 1- \dfrac{(Ax)_i-b_i}{d_i} & ;\ \text{if } b_i \leq (Ax)_i \leq b_i+d_i \\ 0 & ;\ \text{if } (Ax)_i \geq b_i+d_i \end{cases} \qquad (2)$$

$$(Ax)_i \quad b_i \Rightarrow \mu_i(x) = \begin{cases} 0 & ; \text{if } (Ax)_i \geq b_i + d_i \\ 1- \dfrac{b_i - (Ax)_i}{d_i} & ; \text{if } b_i \leq (Ax)_i \leq b_i + d_i \\ 1 & ;\text{if} (Ax)_i \leq b_i \end{cases} \qquad (i = m_1+1, \ldots, m_2) \tag{3}$$

$$(Ax)_i \geq b_i \Rightarrow \mu_i(x) = \begin{cases} 0 & ; \text{if } (Ax)_i \leq b_i - d_i \\ 1- \dfrac{(Ax)_i - b_i}{d_i} & ; \text{if } b_i - d_i \leq (Ax)_i \leq b_i \\ 1 & ; \text{if } (Ax)_i \geq b_i \end{cases} \qquad (i = m_1+1, \ldots, m_3) \tag{4}$$

Here, aspiration value for i*th* fuzzy goal that is determined by decision maker is shown by b_i and accepted tolerance amount for deviation from this aspiration value is shown by d_i.

Goal programming model with fuzzy aspiration value was firstly proposed by Narasimhan. Narasimhan's approach is defined as determining the solution vector "x" from the constraints set below.

$$\begin{aligned} (Ax)_i \quad b_i \quad & i=1,2,...,m_1 \\ x_j \geq 0 \quad & j=1,2,...,n \end{aligned} \tag{5}$$

Narasimhan accepted fuzzy goals as fuzzy equalities and described them with triangular membership functions. Narasimhan has been inspired by the Zimmermann's solution approach for fuzzy linear programming and tried to determine the solution of fuzzy goal programming model on the basis of fuzzy decision set concept. This approach aims to determine the member with highest membership degree of fuzzy decision set. In one respect it aims to increase the membership degrees of fuzzy goals. For this, the solution of the problem that is given below is required [8]:

$$\mu_{\underset{\sim}{D}}(x^M) = \max_{x \geq 0} (\min[\ \mu_i(x)]) \tag{6}$$

For solving this problem, to put the membership functions that describe fuzzy goals in equation 6 is required. But in this situation, we will face with difficulty. To define i*th* membership function with two linear function causes this difficulty. If membership functions are considered as the part that the membership degree is increasing from 0 to 1 and decreasing from 1 to 0, it will be possible to deal with the difficulty. By this way, the problem of determining the member with the highest membership degree is transformed into two sub-problems. In another words, it is tried to determine that x^M vector belongs to which interval from $[b_i-d_i, b_i]$ and $[b_i, b_i+d_i]$. From this point sub-goals are formed for i *th* fuzzy goal as [11]:

First problem:

$$\max_{x\geq 0}\ \{\min[1-\frac{bi-(Ax)i}{di}]\}$$

constraints

$$b_i\text{-}d_i \leq (Ax)_i \leq b_i$$

$$i=1,2,\ldots,m_l$$

Second problem:

$$\max_{x\geq 0}\ \{\min[1-\frac{(Ax)i-bi}{di}]\}$$

constraints

$$b_i \leq (Ax)_i \leq b_i+d_i$$

$$i=1,2,\ldots,m_l \qquad (7)$$

If λ variable that shows the degree of reaching goals is defined, these problems can be expressed as linear programming model.

First problem:

$$\max \lambda$$

constraints

$$1\text{-}\frac{b_i-(Ax)_i}{d_i} \geq \lambda$$

$$b_i\text{-}d_i \leq (Ax)_i \leq b_i$$

$$i=1,2,\ldots,m_l$$

Second problem:

$$\max \lambda$$

constraints

$$1\text{-}\frac{(Ax)_i-b_i}{d_i} \geq \lambda$$

$$b_i \leq (Ax)_i \leq b_i+d_i$$

$$i=1,2,\ldots,m_l \qquad (8)$$

Here, while x^M vector satisfies $b_i\text{-}d_i \leq (Ax)_i \leq b_i$ inequality for any fuzzy goal, it can satisfy $b_i \leq (Ax)_i \leq b_i+d_i$ inequality for another fuzzy goal. For this reason, the sub-problems are combined as [8]:

$$\max \lambda$$

constraints

$$\left.\begin{array}{l} 1\text{-}\dfrac{b_i-(Ax)_i}{d_i} \geq \lambda \\ b_i\text{-}d_i \leq (Ax)_i \leq b_i \end{array}\right\} \quad \text{for some i's}$$

$$\left.\begin{array}{l} 1\text{-}\dfrac{(Ax)_i-b_i}{d_i} \geq \lambda \\ b_i \leq (Ax)_i \leq b_i+d_i \\ \lambda \in [0,1] \end{array}\right\} \quad \text{for the other i's}$$

$$x \geq 0 \qquad (9)$$

In Narasimhan approach, 2^{m1} units of sub-problems are formed for goal programming model with m_l units of fuzzy goal. The solution of a sub-problem that gives the highest λ value is accepted as the solution of fuzzy goal programming model in Narasimhan approach.

4 Application Sample

4.1 Aim of the Application

The aim of the application is to determine the weekly production plan and the profit of the firm and to find optimal solution by using fuzzy goal programming algorithm.

4.2 Application

A manufacturing firm decides to decrease the production amount of a production line that works without profit. Management divides the remaining capacity to three products. Table 1 gives the capacity of machines that restrict the amounts of output and the machine hours required for the production of a product. According to the information that are taken from the sales department, sale potential of product 1 and product 2 are bigger than the production amount and the sale potential for product 3 is approximately 20 units for a week. The unit profit of product 1 is 30 TL, the unit profit of product 2 is 12 TL and the unit profit of product 3 is 15 TL. The firm wants to earn approximately 1800 TL profit per week.

Table 1. Machine hours required for production of products and the capacity of machines

	Required unit time (hour)			
Machine Type	**Product 1**	**Product 2**	**Product 3**	**Weekly machine time (hour)**
Rubbing	9	3	5	550
Polishing	5	4	0	350
Finishing	3	0	2	150

4.3 Application of the Model with WinQSB Program

By using these information, yearly production plan of the firm and the profit amount will be tried to determine. By showing the amount of product 1 that will be produced by x_1 variable, the amount of product 2 by x_2 and product 3 by x_3 we can define this problem as fuzzy goal programming model.

$30x_1+12x_2+15x_3 \underset{\sim}{=} 18000$ (profit goal)
$x_3 \underset{\sim}{=} 40$ (sale goal)
$9x_1+3x_2+5x_3 \leq 500$ (rubbing constraint)
$5x_1+4x_2 \leq 350$ (polishing constraint)
$3x_1+2x_3 \leq 150$ (finishing constraint)
$x_1, x_2, x_3 \geq 0$

We can suppose that tolerance amounts for profit and sale goals are determined by manager of the firm as respectively 2000 TL and 20 unit product 3. Under these circumstances, we can define membership functions for fuzzy goals as:

$$\mu_{profit}(x) = \begin{cases} 0 & ;\text{ if } 30x_1+12x_2+15x_3 \leq 1600 \\ 1-\dfrac{1800-(30x_1+12x_2+15X_3)}{200} & ;\text{ if } 1600 \leq 30x_1+12x_2+15x_3 \leq 1800 \\ 1-\dfrac{(30x_1+12x_2+15x\ )-1800}{200} & ;\text{ if } 1800 \leq 30x_1+12x_2+15x_3 \leq 2000 \\ 0 & ;\text{ if } 30x_1+12x_2+15x_3 \geq 20000 \end{cases}$$

$$\mu_{\text{sale}}(x) = \begin{cases} 0 & ;\text{ if } x_3 \leq 20 \\ 1-\dfrac{40-x_3}{20} & ;\text{ if } 20 \leq x_3 \leq 40 \\ 1-\dfrac{x_3-20}{20} & ;\text{ if } 40 \leq x_3 \leq 60 \\ 0 & ;\text{ if } x_3 \geq 60 \end{cases}$$

To determine the member with the highest membership degree of fuzzy decision set, we must take $[b_i-d_i,\ b_i]$ and $[b_i,\ b_i+d_i]$ intervals into consideration that fuzzy goals are defined. If we look over the membership functions above, it is seen that profit goal is between [1600,1800] and [1800,2000] intervals, sale goal is between [20,40] and [40,60] intervals. To find out in which intervals X^M vector takes place, four ($2^{m_1}=2^2=4$) linear programming problem must be solved. Because there are two fuzzy goals like profit and sale in the model. The solution of linear programming models and the solutions that are found with WinQSB program are given below:

The model that is formed for $(30x_1+12x_2+15x_3) \in [1600,1800]$ and $x_3 \in [20,40]$ intervals:

max λ
constraints
$9x_1+3x_2+5x_3 \leq 500$
$5x_1+4x_2 \leq 350$
$3x_1+2x_3 \leq 150$
$1-\dfrac{1800-(30x_1+12x_2+15x_3)}{200} \geq \lambda$
$1600 \leq 30x_1+12x_2+15x_3 \leq 1800$
$1-\dfrac{40-x_3}{20} \geq \lambda$
$20 \leq x_3 \leq 40$
$\lambda \in [0,1]$
$x_1, x_2, x_3 \geq 0$

max λ
constraints
$9x_1+3x_2+5x_3 \leq 500$
$5x_1+4x_2 \leq 350$
$3x_1+2x_3 \leq 150$
$30x_1+12x_2+15x_3-200\lambda \geq 1600$
$30x_1+12x_2+15x_3 \geq 1600$
$30x_1+12x_2+15x_3 \leq 1800$
$x_3-20\lambda \geq 20$
$x_3 \geq 20$
$x_3 \leq 40$
$\lambda \leq 1$
$x_1, x_2, x_3, \lambda \geq 0$

Optimal Solution
$X_1=7{,}1429$
$X_2=78{,}571$
4
$X_3=40$

The model that is formed for $(30x_1+12x_2+15x_3) \in [1600,1800]$ and $x_3 \in [40,60]$ intervals:

max λ

constraints

$9x_1+3x_2+5x_3 \leq 500$

$5x_1+4x_2 \leq 350$

$3x_1+2x_3 \leq 150$

$1-\frac{1800-(30x_1+12x_2+15x_3)}{200} \geq \lambda$

$1600 \leq 30x_1+12x_2+15x_3 \leq 1800$

$1-\frac{x_3-40}{20} \geq \lambda$

$40 \leq x_3 \leq 60$

$\lambda \in [0,1]$

$x_1, x_2, x_3 \geq 0$

max λ

constraints

$9x_1+3x_2+5x_3 \leq 500$

$5x_1+4x_2 \leq 350$

$3x_1+2x_3 \leq 150$

$30x_1+12x_2+15x_3-200\lambda \geq 1600$

$30x_1+12x_2+15x_3 \geq 1600$

$30x_1+12x_2+15x_3 \leq 1800$

$x_3+20\lambda \geq 60$

$x_3 \geq 40$

$x_3 \leq 60$

$\lambda \leq 1$

$x_1, x_2, x_3, \lambda \geq 0$

Optimal Solution

$X_1=3,3333$

$X_2=83,3333$

$X_3=44$

The model that is formed for $30x_1+12x_2+15x_3) \in [1800,2000]$ and $x_3 \in [20,40]$ intervals:

max λ

constraints

$9x_1+3x_2+5x_3 \leq 500$

$5x_1+4x_2 \leq 350$

$3x_1+2x_3 \leq 150$

$1-\frac{(30x_1+12x_2+15x_3)-1800}{200} \geq \lambda$

$1800 \leq 30x_1+12x_2+15x_3 \leq 2000$

$1-\frac{40-x_3}{20} \geq \lambda$

$20 \leq x_3 \leq 40$

$\lambda \in [0,1]$

$x_1, x_2, x_3 \geq 0$

max λ

constraints

$9x_1+3x_2+5x_3 \leq 500$

$5x_1+4x_2 \leq 350$

$3x_1+2x_3 \leq 150$

$30x_1+12x_2+15x_3+200\lambda \leq 2000$

$30x_1+12x_2+15x_3 \geq 1800$

$30x_1+12x_2+15x_3 \leq 2000$

$x_3-20\lambda \geq 20$

$x_3 \geq 20$

$x_3 \leq 40$

$\lambda \leq 1$

$x_1, x_2, x_3, \lambda \geq 0$

There is no available solution

The model that is formed for $(30x_1+12x_2+15x_3) \in [1800,2000]$ and $x_3 \in [40,60]$ intervals:

$$\left.\begin{array}{l} \max \lambda \\ \text{constraints} \\ 9x_1+3x_2+5x_3 \le 500 \\ 5x_1+4x_2 \le 350 \\ 3x_1+2x_3 \le 150 \\ 1-\frac{(30x_1+12x_2+15x_3)-1800}{200} \ge \lambda \\ 1800 \le 30x_1+12x_2+15x_3 \le 2000 \\ \\ 1-\frac{x_3-40}{20} \ge \lambda \\ 40 \le x_3 \le 60 \\ \hfill x_3 \le 60 \\ \lambda \in [0,1] \\ x_1, x_2, x_3 \ge 0 \end{array}\right\} \left.\begin{array}{l} \max \lambda \\ \text{constraints} \\ 9x_1+3x_2+5x_3 \le 500 \\ 5x_1+4x_2 \le 350 \\ 3x_1+2x_3 \le 150 \\ 30x_1+12x_2+15x_3+200\lambda \le 2000 \\ 30x_1+12x_2+15x_3 \ge 1800 \\ 30x_1+12x_2+15x_3 \le 2000 \\ \\ x_3+20\lambda \le 60 \\ x_3 \ge 40 \\ \\ \lambda \le 1 \\ x_1, x_2, x_3, \lambda \ge 0 \end{array}\right\} \begin{array}{c} \text{There is} \\ \text{no avail-} \\ \text{able solu-} \\ \text{tion} \end{array}$$

In Narasimhan approach, the solution of a sub-problem that gives the highest λ value is accepted as the solution of fuzzy goal programming model. For this reason, the optimal solution of fuzzy goal programming is found in the level of λ =0.8. In other words the fuzzy goals that are determined by decision maker are reached in the level of 0.8. While λ is equal to 0.8, $x_{1,}$ x_2 and x_3 are found respectively as 3,3333, 83,3333 and 44. As a result, profit that is earned in one week is 1780 TL.

5 Conclusions

The most important factor that affects the effectivity of the solution of the fuzzy linear programming is parameters that are used for showing the fuzziness. The type of fuzzy geometry that parameters form is the most sensitive point of the decision making process. Because the success of solution depends on the success of the model in reflecting the system. This makes determining the parameters that form the model extremely important.

While forming the mathematical models of decision making problems of real world, two main characteristics should be taken into consideration: First one is the goals, and the second one is fuzziness in the definition of the problem. At the end of 1970s, these two characteristics were expressed mathematically together by using fuzzy set theory in multi-objective decision making [12].

This study consists of some theoretical information about goal programming, fuzzy set theory and fuzzy goal programming and lastly application of fuzzy goal programming to a linear programming example.

The source of the fuzziness in the application sample is fuzzy goal values of objectives. The fuzzy goal programming model has solved with the tolerance intervals that are given to the profit and sale goal and the production amount of the firm is determined. In according to the solution, the firm earns 1780 TL profit by producing 3 units of product 1, 83 units of product 2 and 44 units of product 3.

The solution of application sample that is modeled by fuzzy goal programming proves that fuzzy goal programming has more flexible and appropriate mathematical frame than classic goal programming. As a result, it is possible to say fuzzy goal programming technique gives more meaningful results than classic technique in applications.

References

[1] Paksoy, T., Atak M.: Etkileşimli Bulanık Çok Amaçlı Doğrusal Programlama ile Bütünleşik Üretim Planlama, Gazi Üniversitesi Fen Bilimleri Enstitüsü Dergisi, **15** (2) (2003), 457-466.

[2] Hammerbacher, I.M., Yager, R.R.: The Personalization of Security Selection: An Application of Fuzzy Set Theory, Fuzzy Sets and Systems **5**, 1-9 (1981). North Holland, Publishing Company

[3] Ertuğrul, İ.: Bulanık Mantık ve Bir Üretim Planlamasında Uygulama Örneği, Y.L. Tezi (1996).

[4] Özkan, M. M., Bulanık Hedef Programlama, Ekin Kitabevi, Bursa (2001).

[5] Güneş, M., Yiğitbaşı, O.N.: Türk Vergi Sisteminde Bulanık Mantık Uygulamaları. 5. Ulusal Ekonometri ve İstatistik Sempozyumu, Çukurova (2001). (www.ceterisparibus.net/kongre/cukurova_5htm)

[6] Ertuğrul, İ.: Bulanık Hedef Programlama ve Bir Tekstil Firmasında Uygulama Örneği, Osman Gazi Üniversitesi, Sosyal Bilimler Dergisi, Aralık 2005, Cilt: 6, Sayı: 2, s. 45 -75, Eskişehir.

[7] Render, B., Stair R.M.: Quantitative Analysis for Management, Fourth Edition, (1991).

[8] Özkan, M.M.: Çok Amaçlı Doğrusal Programlama ve Bir Tekstil İşletmesinde Uygulama Denemesi, Yüksek Lisans Tezi, Uludağ Üniversitesi, Bursa (1994).

[9] Winston W. L.: Operation Research, Applications and Algorithms, Third Edition, Duxbury Press, California (1994).

[10] Rubin P.A., Narasimhan R.: Fuzzy Goal Programming with Nested Priorities, Fuzzy Sets and Systems **14**(1984), 115-129, North Holland

[11] Narasimhan, R.: Goal Programming in a Fuzzy Environment, Decision Sciences 11 (1980).

[12] Arıkan, F.: Bulanık Hedef Programlamanın Çok Amaçlı Proje Şebekesi Problemine Uygulanması, Y.L. Tezi (1996).

The Usage of Fuzzy Quality Control Charts to Evaluate Product Quality and an Application

Irfan Ertuğrul[1] and Mustafa Güneş[2]

[1] University of Pamukkale, Fac., of Economic and Adm. Sciences,
Department of Bussines, Denizli- Turkey
iertuğrul@pamukkale.edu.tr
[2] University of Dokuz Eylül, Fac., of Economic and Adm. Sciences, Department
of Econometrics, Buca-İzmir, Turkey
Mustafa.gunes@deu.edu.tr

Abstract. The quality improvement is of the great importance to strength a competitive position in our markets today. Though improving the quality, shrinkages and so production costs decrease and the customers obtain the appropriate products and services to use. Models are needed for transferring information from one place to another quickly, decreasing and even for eliminating it in the complex subjects. These vagueness is explained by the fuzzy set concept which is useful for making optimal decision under uncertainty and which is accepted as inference based on a specific logic. Control charts have an efficient usage field to keep the process under control. Control charts are accepted as graphical analysis method which determines the products whether to remain in the acceptable limits or not and as a graphical analysis method which gives a signal in the case of product to be out of these limits. In this study by revealing basic idea and principles behind the control charts usage and the improvement; they are combined with fuzzy quality control charts and an application about their usage is mentioned. As a result of the application, it's possible to say that building fuzzy control charts have a more flexible and a more appropriate mathematical description concept and have more reasonable results than the traditional quality chart techniques.

Keywords: Statistical Quality Control, Control Charts, Fuzzy Logic, Fuzzy Control Charts.

1 Quality Control and Techniques

Quality control is the measurement and evaluation for realized quality performance. In other words, it is to compare planned quality goals with realized situation and to determine necessary corrective actions to minimize or remove deviations by determining deviations from goals [1].

Statistical process control is a feedback circle that determines whether correspond or not to required properties by measuring products taken as output from a process and provides input to decision support systems. This regulation, not only controls products, but also provides to control and improve the process consistently.

It is impossible that all products produced in a firm are not same in point of attributes and measurement values. This situation, it means that quality level shows

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 660–673, 2007.
springerlink.com

“variety” product by product. An important point is not exceed from the certain tolerance level of this variation, not to affect the functionality of product and to be at acceptable quality level to satisfy customers’ needs and expectations. So improving quality not only decreases cost but also produces more consistent products which will in turn lead to greater customer satisfaction [2].

“Statistical quality control techniques” are useful to search quality levels of produced products and to determine quality variations. With this aim, instead of controlling all products come from production process, samples that can represent process adequately at certain time intervals are taken, and estimations or inference are made about process by evaluating the results of them [3]. The primary tool of SPC is the statistical control chart named by Shewart [4].

Before production process tolerance limits are determined according to certain measures by determining quality attributes of product that will be produced. It is accepted as normal that quality attributes like size, shape, durability, performance, change in certain limits determined before. Quality control charts are prepared schemes for the aim of measuring the change of certain product group to quality limits that are determined before. After information obtained with quality control charts, if the quality attributes of a certain product are out of quality limits, the causes of this situation are searched and taken corrective actions to control the production process [5]. The power of control charts lies in their ability to detect process shifts and to identify abnormal conditions in a production process [6].

The variation in the quality attributes is understood whether it is random or it is resulted from determined results with control chart. Ideally in the process there must be only the chance in other words common cause factors. Because they show possible minimum change. If there are no variation caused from determined factors, process is “in control statistically”. Thus, observation methods determined with chance rules are in control limits. In this situation, there are common causes and process must be continued with present situation.

In the statistical quality control the most used control charts are explained in the name of “control charts for measurable and immeasurable attributes. The most used control charts among them are P charts, c charts, $\overline{x}-\sigma$ control charts and $\overline{x}-R$ control charts. The method that will be followed to plot all these charts are;

- Quality attribute that will be observed is determined.
- According to certain sample method, measurement values are noted by taking enough sample.
- Control chart type is determined.
- Control limits are determined.
- Limits are plotted after determining sufficiency of determined limits.
- Points out of control limits are determined and the reasons of these points are searched [7].

Control charts will be used in this study are charts for the measurable variables. So, control charts for the attributes will be mentioned as a concept but will not be explained in detail.

The base of control charts for the measurable variables like length, weight, density, etc are attributes that can be expressed with numbers and measured by the help of tools. Charts used for these attributes are mean ($\overline{\mathrm{X}}$), standard deviation (σ) and distribution range (R) charts. Control of process average or mean quality level is usually with the control charts for means, or the $\overline{\mathrm{X}}$ chart. Process variability or dispersion can be controlled with either a control chart for the standard deviations, called the S chart, or a control chart for the range, called an R chart [8]. These charts can be applied mean with standard deviation ($\overline{\mathrm{X}} - \sigma$) or mean with distribution range ($\overline{\mathrm{X}} - \mathrm{R}$). So in the point of both mean and variety whether the process is in control or not can be searched. In practice, the control charts prepared for variation range(R) is more preferable than the control chart prepared for standard deviation. Because distribution range (R) is easier to express in the point of both calculating, explaining and interpreting.

2 Fuzzy Logic

Almost all of events that human being met in the world are complex. This complexity generally is from vagueness, strict idea or not making decision. In a lot of social, economics and technical subjects there is always vagueness because of not being matured of human ideas accurately. Computers improved by human can't operate these kinds of vagueness and quantitative information is necessary for their operations. To understood the real event accurately is not possible completely because of inadequateness of human information. For this reason, there is thinking approximately such these event in their thinking system and their idea or ability to make operation with data and information consists of very inadequateness and vagueness. Generally incomplete and unstrict information sources like complexity and vagueness that arise from in different forms are called " fuzzy sources" [9].

Fuzzy logic approach gives machine to ability of operating special data of human and studying by using their experience and intuition. While having this ability symbolic expressions are used instead of numerical expressions. So transferring these symbolic expressions to machines is based on the mathematical base. This mathematical base is Fuzzy Logic Sets Theory and Fuzzy Logic [10].

Fuzzy logic is used in two meaning. In narrow meaning, fuzzy logic is the state of generalized traditional two valued logic. In wide meaning, it expresses all theories and technologies that use fuzzy sets. A fuzzy logic system with imprecise information exhibits a human-like thinking process that is good at dealing with qualitative, inexact, uncertain and complicated processes [11].

In general, fuzzy logic based on fuzzy sets helps modeling vague and uncertain data that occur usually in real world by providing to realize operations that are identical to human thinking. In traditional logic, one proposition is "right" or "wrong". But the events in real world should be determined the degree to which is good or wrong for example, if the temperature of water at 100 ^{o}C degree is expressed as "hot", so for water at 95 ^{0}C, 80 ^{0}C the expression of "not hot" is not wrong or right in this meaning. For this reason fuzzy set concept is suggested by using the values between right(1) or wrong(0) values. Fuzzy sets theory realizes graduated data modeling by using linguistic structure such as few, little, frequent, medium, low, many, a lot. So it provides to obtain results that are more realistic and close to nature in modeling events. Because the rules introduce the behavior of fuzzy system, fuzzy sets learn themselves [12].

The first of two situations that fuzzy logic is the most valid is the pay attention on view and the judgement of people when there is very complexity of the observed event and adequate information about it and the second one is the cases that are necessary to human thinking conception and decision making from fuzzy logic even if all kinds of problem met are complex it doesn't mean that they can solve. But at least it means that human thinking can be more understandable because of having linguistic inferences deal with observed event.

Excluded middle principle and contradiction principle, one of the important characteristic differ fuzzy logic from the other systems and the most important one for the other fuzzy systems and also is basic rule is not valid for fuzzy logic. In fuzzy logic, it can't be said a proposition is not be both right or wrong at the same time. This case is resulted from the multivalued accuracy and in this framework, the meaning imposed to "and" conjuction. Fuzziness is resulted from the vagueness between one proposition and its complement.

For analyzing the event with fuzzy logic, firstly it is necessary to decide before, that inferences are in the certain tolerance limits. High accuracy causes not only high costs but also the complexity of solving problem at the same time.

Before beginning the problem solving, the decision about the best method for the solution must be decided by considering quantitative and linguistic data that can be collected. Fuzzy logic is also efficient for operating linguistic data. Fuzzy operation can be gotten by introducing this type of information to computer. For this, the most valid methodology fuzzy set, logic and systems. The base of fuzzy system is used in decision making process for inputs built linguistic variables from membership function. These variables match with each other by linguistic IF-THEN rules' preconditions. The result of each rule is determined by obtaining numerical value with defuzzified method from the membership values of input. The information generally obtained from expert operators is used for the rule list of fuzzy logic system and the design of membership function. Membership functions are used being triangle, trapezoid or curve. According to the property of the system controlled, suitable function can be used except them.

The principle of fuzzy sets, logic and system benefit to find solution totally by operating linguistic information that the expert people will give. Each linguistic information is equal to a fuzzy set. Membership degree functions at fuzzy sets can be decided by doing subjective suggestions. So, fuzzy sets help to improve the dialog between people.

The variability in membership functions at fuzzy sets decrease the vagueness of managers' decision making. In decision making fuzzy sets theory can be applied to OR techniques such as linear programming, nonlinear programming, goal programming, multi-criteria decision making, dynamic programming, waiting line models, transportation models, game theory and network analysis.

There are many benefits in application of fuzzy logic. Fuzzy logic can be understood easily and its theories are easy. Flexible but not rigid data, vogue and possibility cases, complex and nonlinear functions can model based on experts' idea and they increase the ability of expert systems. Also, fuzzy logic is natural and it provides the modeling possibility with speaking language [13].

3 Numerical Example

Data used in this paper took from a textile firm which is operating in Denizli. These data shows that the raw yarn is operated and towel errors in kg. For intermediate product towel each 15 days with error reasons.

Fuzzy system built in this paper includes five input variables and one output variable. Each input variable has two membership functions and also output variable has five membership functions. Inputs and output are connected by 32 rules. These rules are given in Table 1. Five input variables, one output variable and their linguistic variables are as follows:

Five input variables:

- Torn (0-50 kg.)
  a) small
  b) large
- Machine Breakdown (0-50 kg.)
  a) low
  b) high
- Pattern Error (0-60 kg.)
  a) few
  b) many
- Dimension Difference (0-300 kg.)
  a) few
  b) many
- Dirt/Stain (0-24 kg.)
  a) small
  b) large

One output variable:

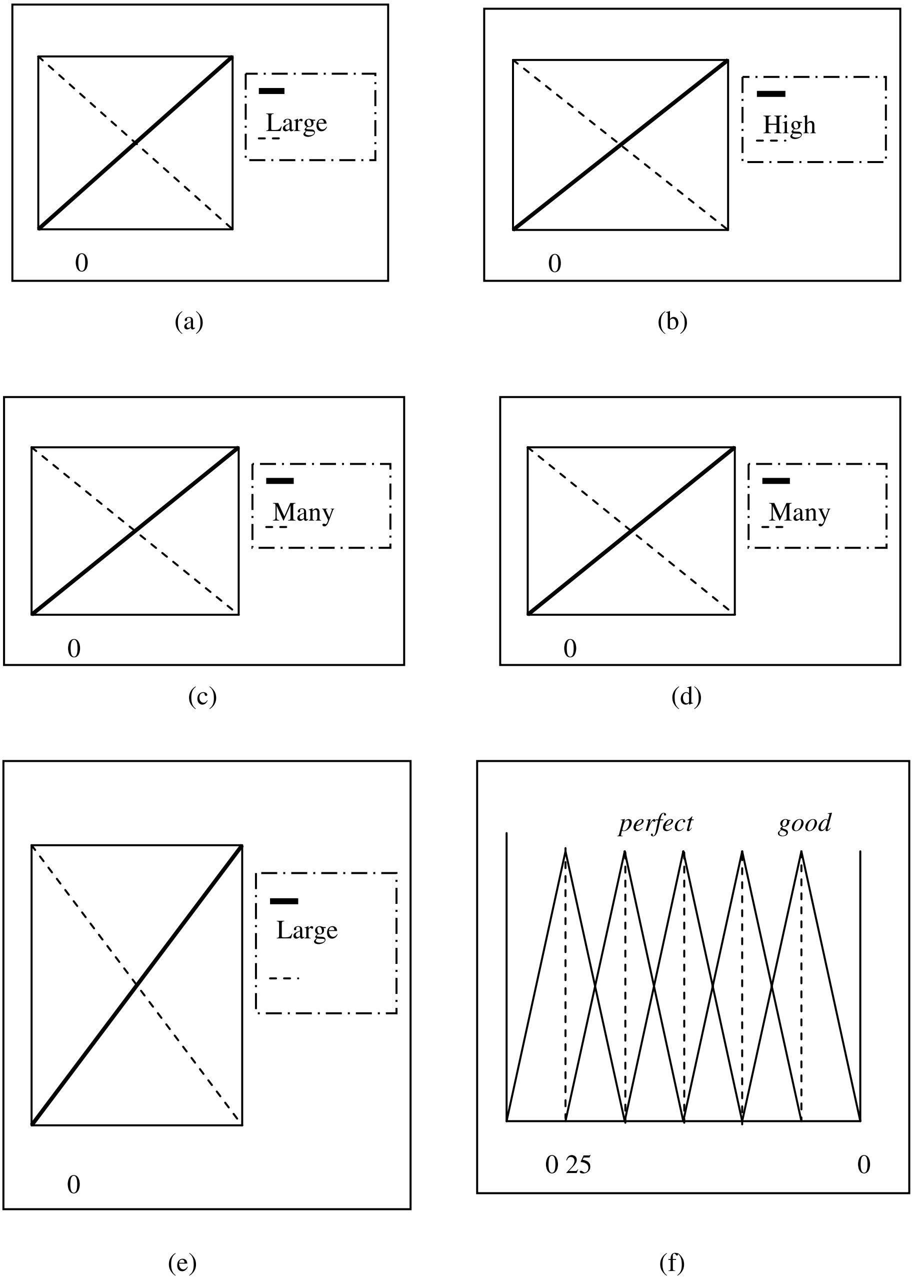
- Intermediate Towel
  a) Perfect
  b) Good
  c) Ok
  d) Bad
  e) Terrible

Membership functions of linguistic variables of each input and output is represented in Figure 1.

Entire calculation for 12 months is not shown but instead of this fact, calculation with the data of the first 15 days which is valid for the month January is shown. The calculation structure of other months' data is same. For the first 15 days of used data is represented monthly in the following Table 2.

Table 1. Rules for Fuzzy System Built (for 5 inputs and 1 output)

Rule Number	Torn	Machine Breakdown	Pattern Error	Dimension Difference	Dirt/Stain	Intermediate Towel
1	small	low	few	few	small	good
2	small	low	few	few	large	terrible
3	small	low	few	many	small	ok
4	small	low	few	many	large	terrible
5	small	low	many	few	small	good
6	small	low	many	few	large	bad
7	small	low	many	many	small	good
8	small	low	many	many	large	terrible
9	small	high	few	few	small	good
10	small	high	few	few	large	bad
11	small	high	few	many	small	good
12	small	high	few	many	large	terrible
13	small	high	many	few	small	perfect
14	small	high	many	few	large	bad
15	small	high	many	many	small	good
16	small	high	many	many	large	bad
17	large	low	few	few	small	good
18	large	low	few	few	large	bad
19	large	low	few	many	small	good
20	large	low	few	many	large	terrible
21	large	low	many	few	small	perfect
22	large	low	many	few	large	bad
23	large	low	many	many	small	good
24	large	low	many	many	large	bad
25	large	high	few	few	small	perfect
26	large	high	few	few	large	bad
27	large	high	few	many	small	good
28	large	high	few	many	large	bad
29	large	high	many	few	small	perfect
30	large	high	many	few	large	ok
31	large	high	many	many	small	perfect
32	large	high	many	many	large	bad



(a) (b) (c) (d) (e) (f)

Fig. 1. Linguistic Variables for Inputs and Output [a) Torn b) Machine Breakdown c) Pattern Error d) Dimension Difference e) Dirt/Stain f) Intermediate Towel]

Table 2. Data Used for the First 15 Days for Each Months

	Torn (0-50)	***Machine Breakdown (0-50)***	***Pattern Error (0-60)***	***Dimension Difference (0-300)***	***Dirt /Stain (0-24)***	***Total***	***Average***
January	6,1	3,5	2	280	13	304,6	60,92
February	7,2	7,8	8	113	8	144	28,8
March	3,5	2,5	3,5	30	7	46,5	9,3
April	10,7	8	5	60	8	91,7	18,34
May	15,3	9,9	9	40	15	89,2	17,84
June	3,6	2,6	1,5	27	15	49,7	9,94
July	4,3	5,5	0	65	4	78,8	15,76
August	2,5	3,5	4,6	20	6	36,6	7,32
September	3,5	6,2	5	21	3	38,7	7,74
October	12,3	19	9	32	5	77,3	15,46
November	13,2	16	10	19	2	60,2	12,04
December	10,5	14	8	32	9	73,5	14,7

Firstly, fuzzy values for input variables of January is computed with the support of membership functions. For example, torn quantity is given 6.1 kg. for January. There is two membership function of "torn" variable which are naming small and large. According to the fuzzy logic, membership degree in small is 0,878 and membership degree in large is 0.122. For January, membership degrees of Machine Breakdown, Pattern Error, Dimension Difference, Dirt/Stain are specified by following the same way in Table 3.

When these data are transferred to rule table, the results illustrated in Table 4 are obtained.

Table 3. Membership Degrees Computed for January

Torn	6,1	Small:	0,878
		Large:	0,122
Machine Breakdown	3,5	Low:	0,93
		High:	0,07
Pattern Error	2	Few:	0,97
		Many:	0,03
Dimension Difference	280	Few:	0,07
		Many:	0,93
Dirt/Stain	13	Small:	0,46
		Large:	0,54

Table 4. Rules Organized for January

Rule Number	Torn	Machine Breakdown	Pattern Error	Dimension Difference	Dirt/Stain	Intermediate Towel	Intermediate Towel
1	0,878	0,93	0,97	0,07	0,46	0,07	good
2	0,878	0,93	0,97	0,07	0,54	0,07	terrible
3	0,878	0,93	0,97	0,93	0,46	0,46	ok
4	0,878	0,93	0,97	0,93	0,54	0,54	terrible
5	0,878	0,93	0,03	0,07	0,46	0,03	good
6	0,878	0,93	0,03	0,07	0,54	0,03	bad
7	0,878	0,93	0,03	0,93	0,46	0,03	good
8	0,878	0,93	0,03	0,93	0,54	0,03	terrible
9	0,878	0,07	0,97	0,07	0,46	0,07	good
10	0,878	0,07	0,97	0,07	0,54	0,07	bad
11	0,878	0,07	0,97	0,93	0,46	0,07	good
12	0,878	0,07	0,97	0,93	0,54	0,07	terrible
13	0,878	0,07	0,03	0,07	0,46	0,03	perfect
14	0,878	0,07	0,03	0,07	0,54	0,03	bad
15	0,878	0,07	0,03	0,93	0,46	0,03	good
16	0,878	0,07	0,03	0,93	0,54	0,03	bad
17	0,122	0,93	0,97	0,07	0,46	0,07	good
18	0,122	0,93	0,97	0,07	0,54	0,07	bad
19	0,122	0,93	0,97	0,93	0,46	0,122	good
20	0,122	0,93	0,97	0,93	0,54	0,122	terrible
21	0,122	0,93	0,03	0,07	0,46	0,03	perfect
22	0,122	0,93	0,03	0,07	0,54	0,03	bad
23	0,122	0,93	0,03	0,93	0,46	0,03	good
24	0,122	0,93	0,03	0,93	0,54	0,03	bad
25	0,122	0,07	0,97	0,07	0,46	0,07	perfect
26	0,122	0,07	0,97	0,07	0,54	0,07	bad
27	0,122	0,07	0,97	0,93	0,46	0,07	good
28	0,122	0,07	0,97	0,93	0,54	0,07	bad
29	0,122	0,07	0,03	0,07	0,46	0,03	perfect
30	0,122	0,07	0,03	0,07	0,54	0,03	ok
31	0,122	0,07	0,03	0,93	0,46	0,03	perfect
32	0,122	0,07	0,03	0,93	0,54	0,03	bad

Min-max principle is applied while specifying value of output. For example, it's known that intermediate towel will be *good* at the first rule. The problem is about the calculation of membership degrees for linguistic variable "good". While specifying membership degrees for linguistic variable "good", the minimum value (0.07) of membership degrees (0.878, 0.93, 0.97, 0.07, 0.46) at the first rule is taken. All rules has processed in the same way. In Table-4, values of last column reveals that consequent perfect appears 5 times, consequent good appears 10 times, consequent ok appears 2 times, consequent bad appears 10 times and consequent terrible appears 5 times. While specifying membership degrees for linguistic variable "perfect", the maximum value (0.07) of membership degrees (0.03, 0.03, 0.07, 0.03, 0.03) is taken. If we apply the same method for computation of membership degrees of other linguistic variables, following results are obtained for January.

Table 5. Membership Degrees of Output's Linguistic Variables for January

Perfect	0,07
Good	0,12
Ok	0,46
Bad	0,07
Terrible	0,54

Representation of these data is illustrated in Figure 2.

The centroid of the shaded area in Figure 2 is defuzzified value. The centroid method is the most common method in the defuzzification. Following formula is used to compute defuzzified value x^*.

$$x^* = \frac{\int \mu(x) x dx}{\int \mu(x) dx}$$

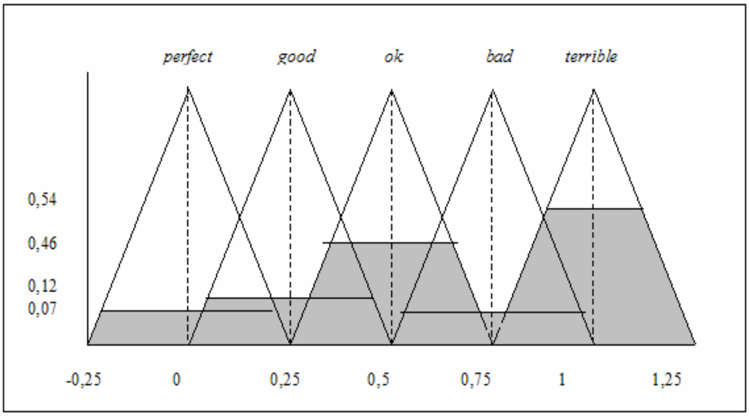


Fig. 2. Representation of Situation for first 15 Days of January with the support of Fuzzy Logic

Table 6. The Values of First 15 Days of Each Months after Defuzzified Operation

January	***February***	***March***	***April***	***May***	***June***
0,685	0,504	0,402	0,435	0,56	0,611
July	***August***	***September***	***October***	***November***	***December***
0,36	0,346	0,213	0,31	0,149	0,373

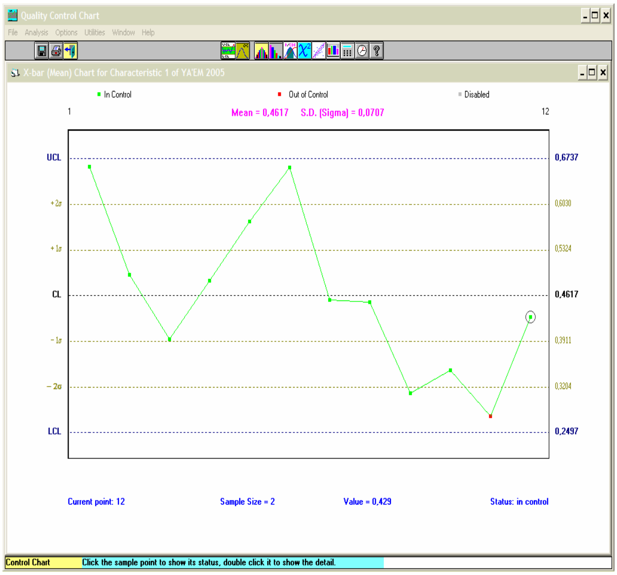
In this case, the centroid is equal to 0,685 for January. Table 6 shows defuzzified value of other months.

In the next step, defuzzified values given above for 12 months will be used for computation of the values of center line, upper and lower control limits to plot fuzzy control charts. Formulas which will be used for plotting fuzzy control charts are the formulas used in traditional quality control charts. Following values are calculated with the aim of these formulas.

Table 7. The Values will be Used in Plotting Fuzzy Quality Control Charts

Lower Control Limit	0,2497
Center Line	0,4647
Upper Control Limit	0,6737

Following quality control chart is developed with the support of these data.



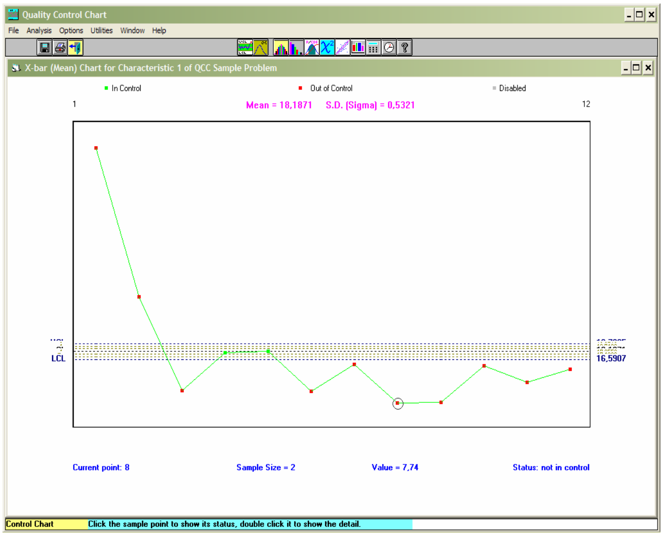
Mean = 0,4617 S.D. (Sigma) = 0,0707

Current point: 12 Sample Size = 2 Value = 0,429 Status: in control

Click the sample point to show its status, double click it to show the detail.

Fig. 3. Fuzzy Quality Control Charts Plotted with the Program of WinQSB

We meet the following control charts when we plot the traditional control chart by using the data given at the beginning of the paper.



Mean = 18,1871 S.D. (Sigma) = 0,5321

Current point: 8 Sample Size = 2 Value = 7,74 Status: not in control

Click the sample point to show its status, double click it to show the detail.

Fig. 4. Traditional Quality Control Charts Plotted with the Program of WinQSB

Table 8. Comments of Fuzzy Quality Control Charts with Linguistic Variables

	Defuzzified Values	**Intermediate Towel**
January	0,661	bad
February	0,494	ok
March	0,394	ok
April	0,4845	ok
May	0,577	ok
June	0,66	bad
July	0,4555	ok
August	0,452	ok
September	0,311	good
October	0,3465	good
November	0,276	good
December	0,429	ok

When we looked at both of two control charts, it's seen that center line, upper and lower control limit have different values. The main reason of this situation is the difference of values used in plotting charts. Values used in traditional control charts are crisp, but in plotting fuzzy control charts are defuzzified values. The audit is provided by translating crisp values to linguistic variables such as such as few, many big, low, high, etc that are frequently used in daily life.

When we evaluate the fuzzy quality control charts, it's seen that production quality is in control for all months. It's noticed that some months' values is about control limits. If we desire to express the result of fuzzy quality chart with linguistic values, we can evaluate first and 6th months as bad, 9th, 10th and 11th months as good, and other months as normal. This situation can be seen in Table 8. When we look at control chart, it is useful to review the months, evaluated as "bad" or "normal" even if their production quality is in control.

On the other hand, it's seen that only 4th and 5th months are in control and others are out of control for traditional control chart.

4 Conclusions

The most important difference of fuzzy logic from the other logic systems is that it allows the using linguistic variables. Linguistic variables provide the concepts which can't be expressed clearly to be qualified approximately. So, linguistic variables become efficient tools that require using of fuzzy sets to express linguistic expression mathematically.

In this study, it has been tried to use approximate thinking instead of thinking based on crisp values. Existing information is transferred to linguistic expressions such as perfect, good, normal, bad, terrible. Fuzzy inference operation is made by rules defined by linguistic variables. For this purpose, these data taken from a textile firm located in Denizli that the raw yarn is operated and towel errors in kg. for intermediate product towel each 15 days with error reasons are analyzed. The fuzzy system built is based on 32 rules with 5 input and 1 output variables. Fuzzy values are calculated monthly for each input variables with the support of membership functions. While calculating output and defuzzied values, min-max principle and centroid method is applied relatively. As we are evaluating the fuzzy quality control charts from the values after defuzzication operation, it has been seen that production quality is under control for all months, but the values reach the control limits in some months, only two months are under control in traditional control charts, however the other months are out of control. As a result of this search, it's possible to say that building fuzzy control charts have more flexible and more appropriate mathematical description frame than control chart approach and give more meaning results than traditional quality chart applications.

References

1. http://www.endustri.8k.com/mmm.html
2. Hefin Rowland, Li Ren Wang, "An approach of fuzzy logic evaluation and control in SPC", 2000, Quality and Reliability Engineering International, 16, p. 91-98.
3. http://www.yazilimvadisi.com/?pg=urun_qpro

4. Reay-Chen Wang and Chung-Ho Chen, “Economic Statistical *np*-control Chart Designs Based on Fuzzy Optimization”, International Journal of Quality & Reliability Management, Vol. 12, No. 1, 1995, pp. 82-92
5. Tekin, Mahmut, Üretim Yönetimi, Arı Ofset Yayıncılık, s.106, Konya, 1999.
6. Wang and Raz,”On the Construction of Control Charts Using Linguistic Variable”, International Journal of Production Research, 1990, Vol: 28, no:3, 477-487.
7. İrfan Ertuğrul, Toplam Kalite Kontrol ve Teknikleri, s.214, Ekin Kitabevi, Bursa, 2004.
8. Montgomery, Douglas C., Introduction to Statistical Quality Control, 1991, Second Edition, p.201.
9. Zekai Şen, Mühendislikte Bulanık (Fuzzy) Mantık ile Modelleme Prensipleri, Su vakfı yayınları,s.7 İstanbul, 2004.
10. Çetin Elmas, Bulanık Mantık Denetleyiciler, Seçkin Yayıncılık, s. 25, Ankara, 2003.
11. Timon Chih-Ting Du , Philip M. Wolfe, “Implementation of Fuzzy Logic Systems and Neural Networks in Industry”, Computers in Industry(32), 1997, p. 261-272.
12. Vasıf V.Nabiyev, Yapay Zeka, Seçkin Yayıncılık, s.640, Ankara, 2003.
13. Nazife Baykal, Timur Beyan, Bulanık Mantık İlke ve Temelleri, Bıçaklar Kitabevi, 2004, s.39-41

Part IX

Intelligent Agents

An Intelligent Belief-Desire-Intention Agent for Digital Game-Based Learning

Christian Anthony L. Go and Won-Hyung Lee

Digital Contents Protection and Computer Game Lab, Graduate School
of Advanced Imaging Science, Multimedia and Film
Chung Ang University, Seoul, 156-756 Korea
chipgo@gmail.com, whlee@cau.ac.kr

Abstract. Interactive digital media, specifically video games, have emerged as a dominant new economic, cultural and, recently, educational force. The prevalence of video games in society today has actually re-wired our brains and has made traditional learning less effective. Today's digital learning methods for young learners fail to engage audiences accustomed to interactive media. And as gaming environments become more complex, games may be more useful in providing alternative paths to hard to teach concepts. This paper introduces an intelligent Belief Desire Intention (BDI) agent architecture for a Non-Player Character that encourages and stimulates situated learning in an online Role-Playing Game.

1 Introduction

Exposure and growing up with technology, especially from an early age, makes using it second nature. Due to the prevalence of video games in society today, our brains have actually re-wired themselves and have made traditional learning less effective. [1] According to the neurology and psychology research in cited in [1], the brain rewires and reorganizes itself according to cultural stimuli. Though traditional educators view educational gaming as mainly useful for drill and practice, as gaming environments increase in complexity, games may be more useful in conveying concepts which are hard to reach using conventional methods.

1.1 MMORPG

The explosive growth of the Internet, rapid advances in computer and network technologies, the widely deployed access and mobile connectivity to Internet infrastructure and services, motivated the designers of computer games to create new, interactive, and distributed multiplayer games, or even to add online multiplayer capabilities to already existing ones. Leveraging these technologies and experiencing phenomenal growth is the Massively Multiplayer Online Game (MMOG) genre, majority of which is dominated by the MMORPG sub genre.

MMORPG stands for Massively Multiplayer Online Role-Playing Game. In an MMORPG, thousands of players exist in the same game world at the same time. MMORPGs provide thousands of hours of game play, with a nearly infinite variety of goals to achieve across a vast world covering miles of land, sea and air.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 677–685, 2007.
springerlink.com

This paper will introduce a BDI agent architecture for a Non-Player Character that acts autonomously and realistically, while, at the same time, gives players a situated learning experience. Its ability to realistically model human behavior by achieving a good balance between reactive and deliberative behavior, while at the same time corresponding to the way people explain their actions and reasoning, have made the BDI Architecture a very popular choice for Agent AI implementations.

2 Related Research

The US military is the world's largest spender on, and user of, digital game-based learning. The military uses games to train soldiers, pilots, sailors, tank drivers and hovercraft operators to master their expensive and sensitive equipment. And because modern warfare increasingly takes place on airplane, tank or battleship computer screens without the operator ever seeing the enemy except as a symbol or avatar, simulations are surprisingly close to actual combat. [3]

The US military also uses games to teach midlevel officers how to employ joint force military doctrine in situations such as battles. Games are also used to teach the art of strategy to senior officers. An example is Joint Force Employment (JFE), a multi-media CD-ROM designed exclusively for today's U.S. Military to convey the concept of "Joint Warfare is Team Warfare". JFE represents a true multi-media environment for joint doctrine education. Players adopt the role of Joint Force Command and tackle ten realistic scenarios to hone their knowledge of doctrine. [4]

Norling and Sonenberg [7] used the BDI architecture to create interactive game characters and successfully demonstrated that a BDI-agent based programming language could clearly and succinctly capture individual differences. Models of expert players of Quake2 were developed. Knowledge was elicited from the players using Applied Cognitive Task Analysis (ACTA) and agent programming was implemented using the JACK Intelligent Agents programming language.

Masuch and Rueger's approach to the Impara game engine was, a very novel and effective one. They developed a game engine in the Squeak/Tweak environment that is used for a commercial game as well as for teaching game design at university. [6]

3 Social Interactions in an MMORPG

MMORPGs are played heavily, with the average time spend in-game being 20 hours a week, and often with friends and family. [23] The virtual worlds players inhabit are persistent social and material worlds, where players are mostly free to do what they will, to kill, steal, slay monsters, etc. The MMORPG virtual world is largely known for its combination of escapist fantasy coupled with emergent social realism. In MMORPGs people save money for homes or equipment, create basket indices of the trading market, build relationships of status and solidarity and even worry about crime and betrayal.

This in-game complexity is born when developers are able to create systems that are so dynamic that a massive number of players can use in-game features to create systemic complexity by interacting according to different frameworks for social structure, politics and economics. Interaction is mediated by the virtual avatars of the individuals who inhabit them. The anonymity afforded by this provides a safe haven beyond the reach of the real world that allows individuals to engage with others socially without repercussions and obligations. [2]

4 Digital Game-Based Learning

4.1 Beyond E-Learning

E-Learning is an all-encompassing term generally used to refer to computer-enhanced learning, although it is often extended to include the use of mobile technologies. It may include the use of web-based teaching materials and hypermedia in general, multimedia cd-roms or websites, discussion boards, collaborative software, e-mail, bligs and wikis, with possibly a combination of different methods being used.

This paradigm is now in turmoil. Critics argue that the proponents of e-learning have lost sight of the grand vision of “learning anytime, anywhere” and have instead replicated the social organization of traditional classroom-based education. [8] As contents are digitized, lectures and notes posted on the Internet, and registration systems are made available online, e-learning has become an evolution in education, not a revolution. The basic organizing elements of traditional education – *knowledge* as discreet and abstract facts, *learning* as the acquisition of content, and therefore *instruction* as the organization, distribution and management of that content – have remained the same. [9]

4.2 Game-Based Learning

An alternative vision for e-learning, commonly called digital game-based learning is emerging. [10] With roots in entertainment, this is a wholly different form of digitally mediated experience, and is emerging as a very powerful form of learning. In recent years, several research projects, organizations, centers, grants, books and studies have emerged, exploring new visions for game-based technologies in learning. [11] Games provide a familiar method of information delivery via a computer or the Internet.

Game players routinely spend hundreds, if not thousands of hours mastering complex skills in digital worlds that are time-consuming, challenging and difficult to master in a highly competitive, rapidly iterating, Darwinian environment, resulting in game mechanics that are highly refined, embodying a wealth of design knowledge. [8] As an industry, games have spent billions of dollars enhancing user interfaces, controls, mechanics and modes of social interaction.

4.3 Situated Learning

As it normally occurs, Learning is a function of the activity, context and culture in which it occurs, it is situated. This is quite opposite to most classroom learning activities involving knowledge, which is abstract and out of context. The critical component of situated learning is Social Interaction -- learners become involved in a "community of practice" which embodies certain beliefs and behaviors to be acquired. As the beginner or newcomer moves from the periphery of this community to its center, they become more active and engaged within the culture and hence assume the role of expert or old-timer. Situated learning is usually unintentional rather than deliberate. These ideas are what Lave & Wenger call the process of "legitimate peripheral participation." [14][15]

It is interesting to note that learning within game environments corresponds with very well with emerging cognitive science research on how people think and learn. [17] Cognitive scientists coming from different traditions have now come to adopt what is called a “situated view of learning”. This proposes that thinking is not a matter of abstract, symbolic representations, but rather rooted in direct experience and concrete contexts. Numerous compelling illustrations of learning based on situated learning are emerging. The most compelling among these are the learning that is naturally occurring in games like Rise of nations, Civilization III, Lineage and Lineage 2. [18]

5 Belief Desire Intention NPC (BDI NPC)

Non-player characters populate the fictional world of the game, and can fill any role not occupied by a player character (PC). Non-player characters might be allies, bystanders or competitors to the PCs.

5.1 Belief Desire Intention (BDI)

The BDI framework describes the way people *think* that they think. This model corresponds closely with how people explain their reasoning and goals. The framework is not involved with an objective view of how the brain actually works. This model represents an abstraction of human deliberation based on a theory of rational actions in the human thinking process. [25]

The BDI framework is the most popular and influential among agent applications. [26]. BDI agents posses the following qualities:

1. Beliefs. These represent the agent’s general knowledge about its environment and itself. An agent can store this information using a database, a set of variables or some other data structure.
2. Desires. These describe the agent’s goal, a system state that the agent wants to achieve. These desires represent the objectives that the agent aims to accomplish or, more generally, the priorities associated with them.
3. Intentions. These are the courses of action that that agent has chosen to achieve that goal, the deliberative state of the agent.
4. Plan. A recipe for action for achieving a certain goal. This is used to guide the decision-making process of agents so they need not search the entire space of possible solutions.

Given its beliefs about the world, an agent will intend to do what will achieve its desires (goals), all in accordance to a plan. An intention is formed when an agent commits to a plan in order to achieve a goal. The steps in the plan may be singular, atomic actions or they can also be subgoals. Since the agent does not need to commit to a particular plan or subgoal until the last possible moment, the result is an agent with a good balance between deliberative (planning) and reactive behavior. [27] [28]

5.2 Non Player Character (NPC)

Non Player Characters must be able to engage players in interesting and entertaining reactions and social interactions. NPC Agents must also be able to engage other NPCs in similarly interesting exchanges. It is imperative for a realistic NPC to be able to make the game player believe that he/she is interacting with an intelligent, sentient character with its own goals, beliefs, desires and agenda.

5.3 A BDI NPC

In order to create believable NPC, the following conditions must be satisfied by our agent architecture:

1. Autonomy
2. Social Ability
3. Reactivity
4. Pro-activeness
5. Persistence

A modular architecture is needed to address these requirements. Figure 4 illustrates the main components of out BDI Agent Architecture.

Behavior. This is a typical static agent system. It is responsible for general agent behavior in the world- traveling, leaving clues, verifying information, attacking, trading etc. Given data from the knowledge store, it will update its behavior and in doing so will also pass and update the information of the knowledge store.

Social. The social system is similar to the behavior module. However, we separate this module to add more realism to NPC's social interactions with players. This is a static system but with a wider repertoire of social skills. Social behavior such as talking, making allies and the like are handled here. The social system gets information and update the knowledge store. It also maintains a separate social memory of encounters with people. At the start of a social interaction, this memory is accessed and at the end of an encounter, it is updated.

Goal Scheduler. This module is responsible for scheduling of goals. These goals have to be predetermined by the game designer. The goal scheduler updates its schedule based on data from the knowledge store and likewise updates the knowledge store with regards to its progress in goals.

Decision Maker. The decision making module is the selection mechanism that mediates between each system and decides what action to take based on the information it receives from them.

Information is collected from the game by agent sensors. These are transformed into a knowledge store to which each of the selected modules has access. This knowledge store represents the NPCs beliefs about its environment.

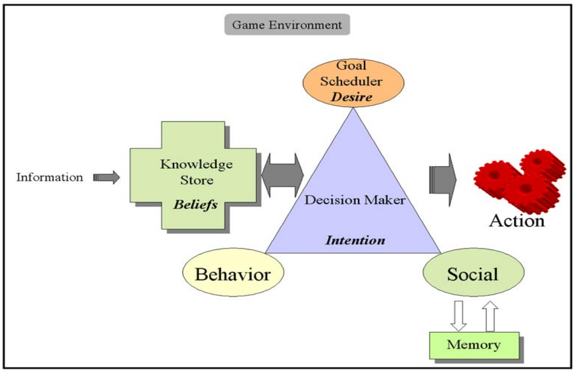
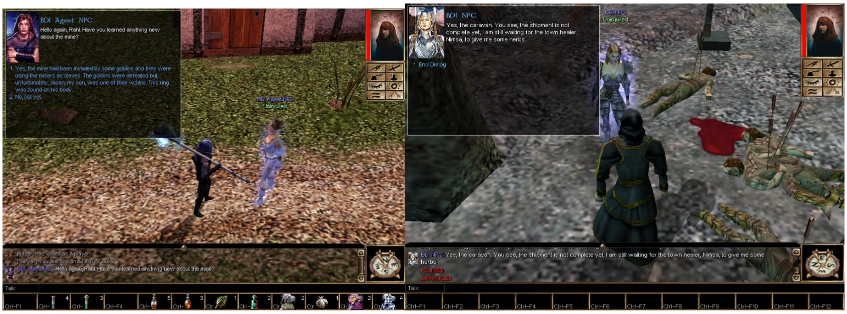


Fig. 1. The NPC BDI Architecture. Note the Separate Social Interaction Module and its dedicated Memory Module.

6 Implementation

By reason of the proprietary nature of most MMORPGs, an alternative, possessing all the elements of an MMORPG yet allowed scripting and code modifications, had to be found. Given these limitations, experiments were conducted on the Aurora Game Engine of BioWare- the game engine used for the massively popular Neverwinter Nights RPG franchise, which has been expanded into an MMORPG. [30]

A prototype NPC using our BDI architecture was implemented. The NPC goal was to impart simple yet novel knowledge to the player through its flexible multi-branched dialogue and quest milestones and items. Information would be verified by the NPC by asking the player a series of follow-up questions at a later time, all in the context of completing a mission. This information was divided into two (2) parts, abstract and contextual information. Abstract information was given as a set of ten (10) trivial facts which were part of the NPC dialogue but unrelated to the mission. Contextual information was provided as a total of ten (10) pieces of information, items to be obtained, and player milestones to achieve. Players were given 20 minutes to complete their mission.



BDI Agent NPC

Hello again, Rahl. Have you learned anything new about the mine?

1. Yes, the mine had been invaded by some goblins and they were using the misers as slaves. The goblins were defeated but, unfortunately, Jacen, my son, was one of their victims. This ring was found on his body.
2. No, not yet.

BDI Agent NPC
Uninjured

Player. has joined as a player..
[Server] You are now in a No PVP area.
BDI Agent NPC: Hello again, Rahl. Have you learned anything new about the mine?

BDI NPC

Yes, the caravan. You see, the shipment is not complete yet, I am still waiting for the town healer, Nimsa, to give me some herbs.

1. End Dialog

BDI NPC
Uninjured

BDI NPC: Yes, the caravan. You see, the shipment is not complete yet, I am still waiting for the town healer, Nimsa, to give me some herbs.
PAUSED
UNPAUSED

Fig. 2. Screenshots of a player Interaction with our BDI NPC (the woman to the right and top)

Fifteen (15) volunteers were employed, all experienced computer game players, all of whom were familiar with the NeverWinter Nights RPG franchise and with navigating and interacting with an NPC. During different intervals after the mission, volunteers were given a short quiz regarding the information they had learned from the game. This was conducted at 2, 24 and 48 hours post-experiment.

6.1 Outcome

On the average, the test subjects correctly recalled 44% of the information for abstract information. Contextual information recall, however, was much higher, with volunteers correctly recalling 71.33% of the information.

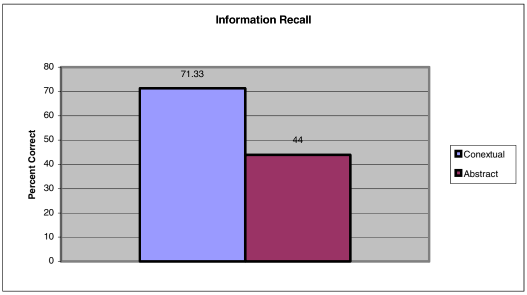


Fig. 3. Abstract and Contextual Information Recall

It is interesting to note while the information recalled does slowly degrade over time due to its being only relevant in the game world, the rate of decay for contextual information was considerably less than abstract information. Contextual information recall degraded to 22% from 72% while abstract information fell from 39% to 5% overall after 48 hours.

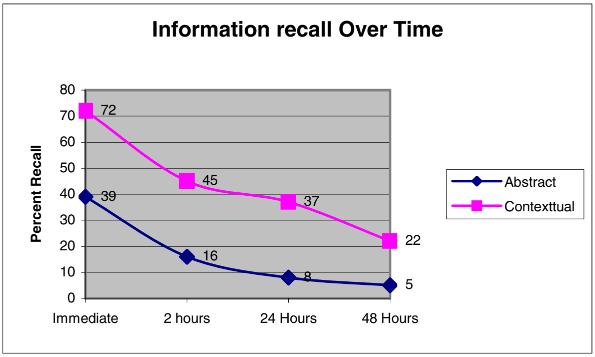


Fig. 4. Percentage of Information recall over time

7 Conclusion

Our preliminary experiments have proven that a BDI NPC can facilitate effective, situated digital game-based in an online game, specifically a Massively Multiplayer Role-Playing Game. Game players can similarly learn abstract facts, along with contextual data, albeit to a lesser degree. We have observed that the degradation of information recall over time can be attributed to two factors: the amount of time spent learning (exposure), and, the lack of further reinforcement. These factors notwithstanding, we have demonstrated that an effective game BDI NPC can give game players the social interaction needed for effective situated learning, by effectively mimicking a real player to a degree. A limitation of our current model however, is the high system overhead its implementation incurs. To avoid unnecessary overhead, it is recommended that the game designer make NPCs that are central to a game persistent BDI agents, while utilizing traditional NPC techniques to implement non-critical game characters. Our research offers a promising glimpse into numerous digital game-based learning opportunities for the immensely-large yet still relatively untapped MMORPG market.

7.1 Future Work

Additional experiments are currently underway. Immediate work in progress includes refinement of the social module of the BDI NPC to add more granularity to player interactions. Because the present prototype introduces significant system overhead, streamlining the BDI NPC Agent is critical. Upon completion of the final version of the model, experiments involving a significantly larger sample shall commence. This sample will include subjects with different skill levels. Future avenues of research will include implementing the agent on the newly-released NeverWinter Nights 2 and, game company willing, may also include actual implementation on a popular online MMORPG such as Lineage and Lineage II.

Acknowledgements. This research was supported by Chung Ang University and Seoul Future Contents Convergence (SFCC) Cluster established by the Seoul R&BD Program.

References

1. Prensky, Marc, “Digital Game-Based Learning”, McGraw-Hill (2001)
2. Steinkuehler, C., and Williams, D, “Where everybody knows your (screen) name: Online games as "third places." Journal of Computer-Mediated Communication, 11(4), article 1 (2006)
3. Prensky, Marc, True Believers: Digital Game-Based Learning in the Military, Digital Game-Based Learning, McGraw-Hill, 2001
4. Joint Force Employment http://www.dtic.mil/doctrine/jfe/
5. Tsushima k, Progress Toward the Science of game and amusement, Proceedings of the 2005 International Conference on Active Media Technology, AMT 2005 (2005)

6. Masuch, M. and Rueger, M., “Challenges in Collaborative Game Design Developing Learning Environments for Creating Games”, Third International Conference on Creating, Connecting and Collaborating through Computing, C5 2005 (2005)
7. Norling, E. and Sonenberg, L., ”Creating Interactive Characters with BDI Agents”, Proceedings of the Australian Workshop on Interactive Entertainment IE2004 (2004)
8. Squire, Kurt, Game-Based Learning, Report to the Masie Consortium (2005)
9. Ednar, A.K., Cunningham D., Duffy, T.M., Perry, D.J.. Theory into practice: How do we link? In Instructional Technology: Past, Present and Future, Englewood, Co: Libraries Unlimited Inc. (1995)
10. Aldrich, C., Simulations and the future of Learning, Pfeiffer, New York (2004)
11. Games-to-Teach team at MIT. Design Principles of next-generation digital gaming for education, Educational Technology (2003)
12. Entertainment Software Association. Top Ten industry Facts, http://www.theesa.com/pressroom.html (2004)
13. Kirk, J. and Belovics, R. An Intro to Online Training Games, http://www.learningcircuits.org/2004/Apr2004/kirk.htm
14. Lave, J., Cognition in Practice: Mind, mathematics, and culture in everyday life. Cambridge, UK: Cambridge University Press (1988)
15. Lave, J., & Wenger, E., Situated Learning: Legitimate Peripheral Participation. Cambridge, UK: Cambridge University Press (1990)
16. Brown, J.S., Collins, A. & Duguid, S., Situated cognition and the culture of learning. Educational Researcher, 18(1), 32-42 (1989)
17. Gee, J.P., Learning by Design: Games as Learning Machines. http://www.gamasutra.com/gdc2004/features/20040324/gee_01.shtml (2004, March)
18. Gee, J.P., “What videogames have to teach us about learning and literacy”. New York: Palgrave Macmillan (2003)
19. Eladhari, M., “Trends in MMOG Development”, http://www.game-research.com/art_trends_in_mmog.asp
20. Putnam, R. D., Bowling Alone: The Collapse and Revival of American Community. New York: Simon & Schuster (2000)
21. Go, C.A. and Lee, W.H., “Virtual Terrain Evaluation Using Pedestrian Agents”, Proceedings of the Second International Conference on Design Computing and Cognition (2006)
22. Woodcock, B., An Analysis of MMOG Subscription Growth-version 18.0. Retrieved July 17, 2006 from http://www.mmogchart.com/ (2006)
23. Yee, N., The Daedalus Project. Retrieved July 17, 2006 from http://www.nickyee.com/daedalus/archives/001468.php (2006)
24. Eladhari, M., “Trends in MMOG Development”, http://www.game-research.com/art_trends_in_mmog.asp
25. Bratman, M.E., “Intentions Plans and Practical Reason”, Harvard University Press: Cambridge, MA, (1987)
26. Rao, A.S. and Georgeff, M.P., “Modeling rational Agents within a BDI architecture”, Proceedings of Knowledge Representation and Reasoning, pp. 473-484, (1991)
27. Rao, A.S., “A Unified View of Plans as Recipes”, Contemporary Action Theory, Kulver Academic Publishers, The Netherlands (1997)
28. Norling, E. and Sonenberg, L., ”Creating Interactive Characters with BDI Agents”, Proceedings of the Australian Workshop on Interactive Entertainment IE2004, 2004
29. Coco, D., “Creating Intelligent Creatures”, Computer Graphics World, (July 1997)
30. Neverwinter Nights, produced by BioWare and published by Infogrames (now Atari)

Improved Scheme for Telematics Fault Tolerance with Agents

Woonsuk Suh[1], Seunghwa Lee[2], and Eunseok Lee[2]

[1] National Information Society Agency
NIA Bldg, 77, Mugyo-dong Jung-ku Seoul, 100-775, Korea
sws@nia.or.kr

[2] School of Information and Communication Engineering, Sungkyunkwan University
300 Chunchun Jangahn Suwon, 440-746, Korea
{shlee, eslee}@ece.skku.ac.kr

Abstract. The CORBA based middlewares are appropriate for Telematics system which is an important and practical infrastructure for ubiquitous convergence. And fault tolerance is a first priority for real time characteristics of transport information provided by Telematics systems. There are many methods to enhance the fault tolerance of CORBA based systems. Among these methods, this paper proposes the one of enabling seamless information services where systems based on the CORBA suffer from object faults which occur processing real time transport information. Namely, this paper observes a method to deal efficiently with object faults occurring in 3 tier architecture environments. It is possible to replicate objects as a method to enhance fault tolerance in preparation against object faults. This paper presents a method of enhancing Telematics fault tolerance and maintaining service continuity by continuing to operate systems until faults of the Fault Tolerant (FT)-CORBA protocol itself is recovered. The proposed agent enhances Telematics performance by adjusting the minimum number of object replicas for real time transport information services during system operation. The evaluation is performed by comparing costs for system fault tolerance which consist of communication traffic and S/W complexities.

1 Introduction

The Telematics is a state-of-the-art transport information system, which maximizes utilization efficiency of transport facilities, enhances transport convenience and safety, and implements environment friendly transport. The system achieves this through economizing energy from applying up-to-date technologies to components of transport systems, such as road, vehicles, goods and then collecting, managing, and providing real time transport information [19]. The key component of Telematics is transport information systems which have characteristics as follows. First, these systems run on communication networks nationwide [5]. Second, travelers can receive and transmit real time information, while driving at high speed. Third, the cycle of transport information update must not be greater than 5 minutes [18]. As a result, it is characterized as a stable service environment.

The Telematics operates on heterogeneous S/W and H/W platforms to satisfy the characteristics, functions, and performance requirements described earlier, and

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 686–695, 2007.
springerlink.com

accordingly, can be built on the Common Object Request Broker Architecture (CORBA). In the US, the DOT notified 'National Transportation Communications for Intelligent Transport System (ITS) Protocol (NTCIP) Application Profile for CORBA' [17] for efficient ITS implementation, which is infrastructure for Telematics, in May 2001, and the Land Transport Authority, Singapore performed the 'traffic.smart' project, which is based on the CORBA [3]. However, the FT-CORBA is required to satisfy real time services of transport information systems.

Accordingly, this paper proposes an agent based method to enhance fault tolerance of the Telematics using the merits of the FT-CORBA in cases high system stability is required, such as airport information systems and land transport information systems. In chapter 2, the FT-CORBA related research is discussed. The proposed architecture introduces agents to complement the FT-CORBA in chapter 3. The performance of the proposed architecture is evaluated quantitatively in chapter 4. This research is summarized and future research directions are presented in chapter 5.

2 Related Work

The Object Management Group (OMG) established a standard which enhances fault tolerance by creating replicas of objects in information systems based on the CORBA. The standard addresses three incompatible fault tolerant solutions to the CORBA – integration, service, and interception strategies [4][6][7][11][12]. The WARM_PASSIVE replication style which maintains a constant replica group size is considered appropriate in Telematics in terms of service requirements and computing resource utilization. However, a method is required to maintain a constant replica group size efficiently.

The current FT-CORBA standard has some limitations [8][15]. First, it assumes that all faults arise from object or host crashes. Accordingly, the FT-CORBA cannot tolerate faults of replication infrastructure including the Replication Manager (RM). The detection, diagnosis, and response to failures other than crash failures are exceedingly challenging problems to address in an application-independent manner [2]. Second, the FT-CORBA is designed to prevent single points of failure within a distributed object computing system: It assumes a single-failure model, where no nested failures can occur while the system is recovering from a previous failure condition. Additionally, the standard does not support legacy ORBs, vendor dependences, deterministic behavior, network partitioning faults, commission faults, and design or programming faults.

This paper focuses on improving first two of the above factors. An additional method is required to satisfy user requirements for timely services and enhance system stability and performance in the case of the FT-CORBA based Telematics.

3 The Proposed System

3.1 Organization and Protocols

As illustrated in Figure 1, for the FT-CORBA implementation using the WARM_PASSIVE scheme 9 steps are required before the client makes a request and the client ORB ensures that the request is transmitted to the primary replica ('client-to-object' communication) [8].

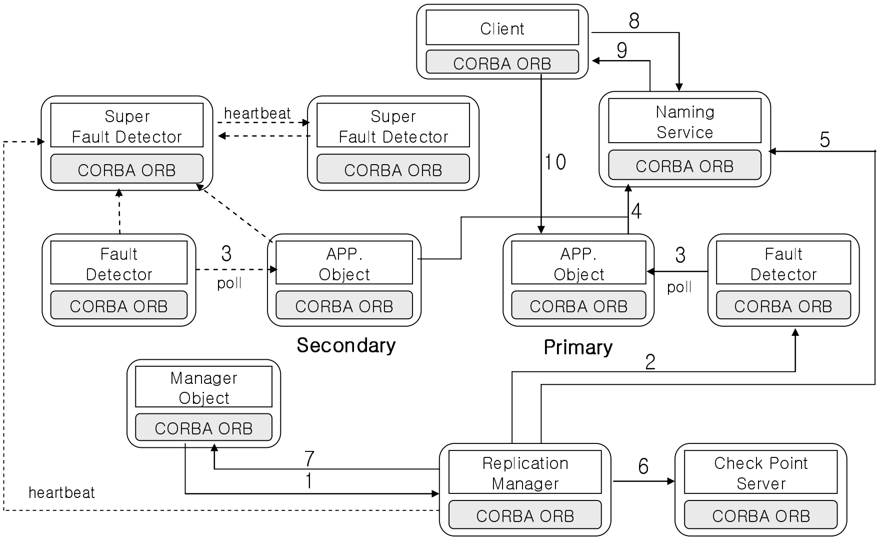


Fig. 1. The FT-CORBA Protocols Using the WARM_PASSIVE Scheme

First, service delay is improved and the possibility of faults of objects is lowered through efficient network management with Mobile Agents (MAs) in the Voyager, an agent operation environment [13]. Second, an architecture is proposed to apply an agent to enhance fault tolerance (FT-Agent), Object Remote Procedure Call (ORPC) and backup objects of the Distributed Component Object Model (DCOM) [16] to the FT-CORBA which provides fault tolerance through object replication. The MAs reduce traffic on networks when objects and clients communicate with each other via ORPC. The above points are illustrated in Figure 2, where the FT-Agent plays the main role of supporting the RM of the FT-CORBA.

3.2 RM and FT-Agent

In Figure 2, notations '1 ~ 4' correspond to the case where the number of replicas of objects is smaller than the preset minimum number (3 in this paper) of normal replicas and notations '① ~ ⑥' correspond to the case where the number of replicas of objects is greater than or equal to the preset minimum number of normal replicas following the recovery mechanism of the FT-CORBA. The FT-Agent monitors the number of replicas of each object (NO, initially 5 in this paper) maintained by the RM. The steps of the protocol are summarized as follows.

A. NO < the preset minimum number of normal replicas

1. The RM shares the NO with the FT-Agent : the FT-Agent knows the preset minimum number of normal replicas and if the NO is smaller than this value,
2. The FT-Agent notifies clients of this status.
3. They request services to the object concerned of the DCOM via the ORPC (using the bridge of the CORBA 2.3 standard [10]).
4. The object processes the requested services and returns the result.

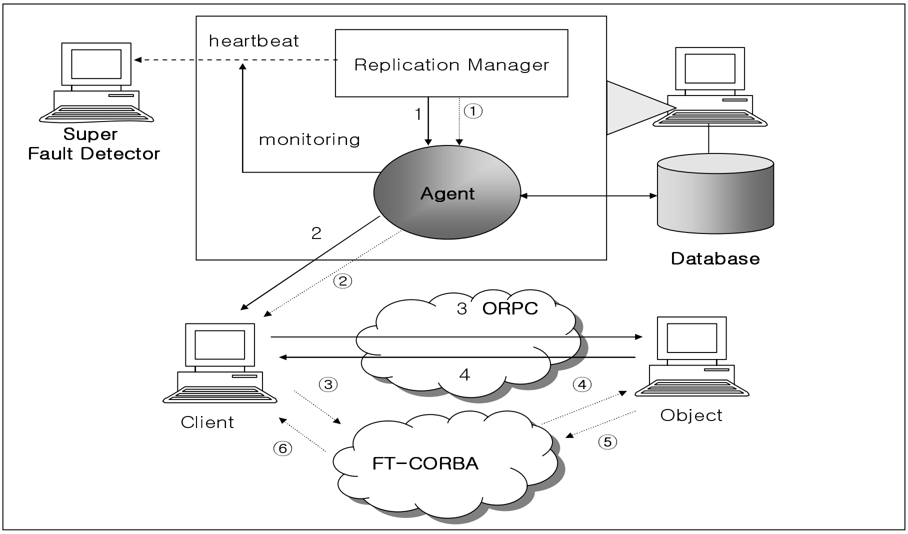


Fig. 2. The Protocol Initiated when FT-CORBA has Problems

B. NO >= the preset minimum number of normal replicas

① The RM shares the NO with the FT-Agent

② If the NO becomes greater than or equal to the preset minimum number of normal replicas, the FT Agent notifies clients of this status

③ ~ ⑥ The clients request and receive services via the recovered FT-CORBA protocol

Although the first steps of the above protocols of 'A' and 'B' are the same information sharing between the RM and the FT-Agent, they are described separately. The first steps of 'A' and 'B' occur when replication of objects occurs. If the heartbeat stops, object services are provided through the protocol of '2 ~ 4' of 'A' and if it resumes, they are provided by the FT-CORBA.

3.3 Organization of the FT-Agent

The FT-Agent is proposed as an agent with a control unit comprising of a Cooperative Planning Layer (CPL), Local Planning Layer (LPL), and Behavior-based Layer (BBL) [1]. The function of the control unit is described in abstractive codes as follows [14].

```
function Agent-FOR-FT-CORBA {percept = number of replicas of each ob-
ject, time[(a),(b)], heartbeat of RM, availabilities and locations of
MAs, network traffic of link LL}/* refer to Table 1 for time
[(a),(b)]*/
returns {action = (announce faults to clients, move to the DCOM mode,
and set the DCOM flag to 1) or (announce the nearest MA to move to
the link LL and the MA balances the network traffic)}
static: state ; Table 1, Table 2, Table 3
```

```
rules :
  if [{the rate of cases where ((b)-(a)) < 5 min.} > 0.5 for the 1
   hour for each object which has a fault]
  then (subtract 1 from the minimum number of replicas of each object
       and maintain FT-CORBA)
  else(maintain the preset or adjusted minimum number of replicas of
     each object)
  if (network traffic of link LL is heavy)
  then(locate the nearest and available MA)
① the FT-Agent percepts the number of replicas of each object (NO), (a),
   and (b) in a cycle of 5 minutes the RM provides in real time;/* Table 1*/
   if ( NO < the preset or adjusted minimum number)
   then (set a time widow from the moment to 1 hour before and calculate α);
   /* refer to Table 1 for α */
   percepts existence of heartbeat of the RM;
   updates (Table 1) and (Table 3); /* LPL */
   percepts availability and locations of MAs and 'network traffic of
   link LL' by set time intervals/*communication between MAs and FT-
   Agent*/ and updates Table 2 and Table 3; /* CPL */
② if (network traffic of link LL is heavy) /* Determined by a preset
    evaluation value and 'p' of Table 2 */
    then (locate the nearest and available MA);/* CPL */
    if [{the rate of cases where ((b)-(a)) < 5 min.} > 0.5 for the 1
    hour for each object which has a fault]
    then(subtract 1 from the minimum number of replicas of each object
         and maintain FT-CORBA)  /* set the DCOM flag to 0 */
    else(maintain the preset or adjusted minimum number of replicas of
        each object); /* set the DCOM flag to 0 *//* LPL */
③ if {(number of replicas of each object < the preset or adjusted
       minimum number) or (no heartbeat of RM)}
    then(action = announce faults to clients, move to the DCOM
   mode, and set the DCOM flag to 1);  /* LPL and BBL */
    if {(network traffic of link LL is heavy) and (the nearest MA is available)}
    then (action = announce the nearest MA to move to the link LL and
    the MA balances the network traffic);/* CPL, LPL, and BBL */
④ change MAs' locations and network traffic status;/*update Table 2*//* CPL */
⑤ Performs the action of ③ and moves to ①; /*return action*//*BBL*/
```

Then, the FT-Agent maintains three internal state tables (DBs) of Table 1 ~ 3.

Table 1. Internal State DB for Fault Management (Local Planning Layer: LPL)

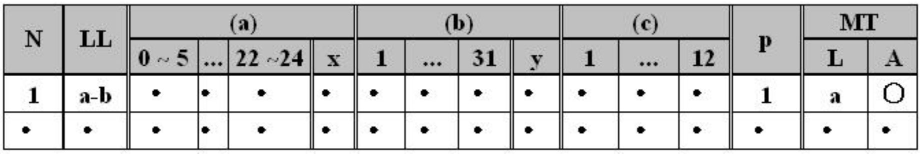
N	ON	NO	PDN	(a)	(b)	α	FD	FH
1	A	5	3	07:00:09	07:05:09	0.5	0	1
				…	…			
•	•	•	•	•	•	•	•	•

N: Object Number; **ON**: Object Name; **NO**: Number of replicas of each object; **PDN**: Preset /During-operation $NO_{minimum}$; **(a)**: Time when 'NO' gets smaller than an existing PDN; **(b)**: Time when 'NO' gets greater than or equal to an existing PDN; **α** : Rate recovered to greater than or equal to $NO_{minimum}$ within 5 min. for 1 hour; **FD**: Flag of DCOM state; **FH**: Flag of heartbeat.

In this paper, it is assumed that the number of all objects = n, NO = 5, and the preset minimum number of replicas of each object ($NO_{minimum}$) = 3. As the database of the planning layer of local systems the FT-Agent runs on, Table 1 is related to occurrences and recoveries of faults and is also used in calculating the data to control the Reactive Layer of the FT-Agent.

The FT-Agent adjusts $NO_{minimum}$ flexibly by managing Table 1. Namely, the FT-Agent maintains the preset $NO_{minimum}$ variably and calculates the rate recovered to greater than or equal to the $NO_{minimum}$ within 5 minutes (update cycle of transport information) [18] for 1 hour of a time window. In the case **α** is greater than the evaluation value (0.5 in this paper), the FT-Agent subtracts 1 from the preset $NO_{minimum}$ of related objects during Telematics operation as far as a resultant $NO_{minimum}$ is greater than or equal to 2, and otherwise maintains $NO_{minimum}$ of related objects as an existing value. The FT-Agent notifies the RM of $NO_{minimum}$ if it has been changed and the RM manages object replication with it.

Table 2. DB for Network Management (Cooperative Planning Layer: CPL)

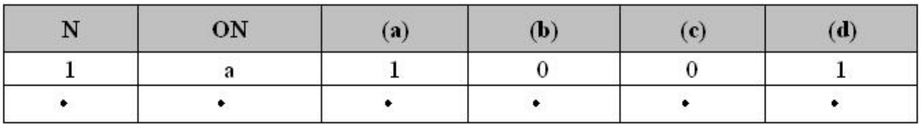


N	LL	(a)				(b)				(c)			p	MT	
		0 ~ 5	...	22 ~24	x	1	...	31	y	1	...	12		L	A
1	a-b	•	•	•	•	•	•	•	•	•	•	•	1	a	○
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

N: Object Number; **LL**: Link Location; **(a)**: Av. Network Traffic Transition(daily)(unit : hour); **x**: Av. daily traffic; **(b)**: Av. Network Traffic Transition(monthly)(unit : day); **y**: Av. monthly traffic; **(c)**: Av. Network Traffic Transition(yearly)(unit : month); **p**: Priority for MA movement; **MT**: MAs tracing; **L**: Location; **A**: Availability.

Table 2 is used for agent based network management. Namely, the FT-Agent reduces network traffic between clients and objects and lowers the possibility of fault occurrences by using MAs to prevent service delay and faults of objects through efficient network management. For these purposes, the FT-Agent distributes MAs optimally by monitoring locations of MAs, directing them to move to links having heavy traffic, and managing their history.

Table 3. State DB to Decide Transfer to ORPC Mode (Bahavior Based Layer: BBL)



N	ON	(a)	(b)	(c)	(d)
1	a	1	0	0	1
•	•	•	•	•	•

N: Object Number; **ON**: Object Name; **(a)**: number of replicas of each object < the preset or adjusted minimum number; **(b)**: no heartbeat of RM; **(c)**: network traffic of link LL is heavy; **(d)**: the nearest MA is available.

Finally, Table 3 contains the data delivered and induced from the CPL and LPL. This is used as input data to enable the FT-Agent Reactive Layer to complement the FT-CORBA by running the ORPC protocol.

4 Performance Evaluation

The index of formula (1) is introduced to compare the performances before and after application of the proposal in this paper.

$$\text{Cost for Fault Tolerance for Entire Telematics (CFT)} = \text{① Communication Traffic (B)} \times \text{② S/W Complexity (C)} \quad (1)$$

The numbers of both objects and clients are set to ‘n’, respectively because their roles change according to services. ‘B’ and ‘C’ have to be small to enhance the fault tolerances of the entire Telematics. The greater the values of ‘B’ and ‘C’, the greater the probability that faults occur with the performance decline of systems. These values are proportional to the number of clients. Namely, ‘B’ and ‘C’ are independent variables with regard to each other, and must be considered simultaneously. Therefore, it is possible that the CFT can be represented as multiplication of ‘B’ and ‘C’ [9].

Accordingly, performances can be evaluated by comparing the CFTs before and after applying the proposed method in this paper. ① and ② of formula (1) are observed in detail.

First, ‘B’ is calculated. The ‘B’ is a value proportional to the number of clients requesting real time services (θ: proportional constant). If it is assumed the number of clients requesting real time services and the service processing time of an object are ‘n’ and ‘t’, respectively and if the 10 steps including the ‘client-to-object’ communication of the FT-CORBA are considered, the communication traffic (B) of the FT-CORBA is as formula (2) in the case no object failures occur.

$$B = 10 \times n\theta + n\theta \times t \quad (2)$$

(10 : number of FT-CORBA steps, θ : proportional constant, t : the service processing time of an object)

In the case of fault occurrences in the FT-CORBA itself, communication traffic (B) is generated as presented in formula (3).

① Traffic between RM and FT-Agent: nθ(they are on the same computer)
② Traffic between the FT-Agent and Clients: nθ (required for clients with objects containing faults)
③ Traffic between Clients and Objects: nθ + nθ×t (required for clients with objects containing faults)
∴ Communication Traffic (B) = 3nθ + nθ×t (t: the service processing time of an object) (3)

The traffic of ‘nθ’ is generated to share status information of objects (n) in ① of formula (3). The FT-Agent announces the relevant objects faults to clients (n) and the maximum ‘nθ’ of traffic is generated in ②. The traffic ‘nθ’ is generated when the maximum ‘n’ of clients request services to an object and then traffic ‘nθ×t’ is generated in ③, for communications occur for the time of ‘t’ when an object respond to service requests.

Second, ‘C’ is calculated. The maximum value of the ‘C’ must be calculated, including the worst case for the comparison of efficiency. This value is proportional to the number of objects. If the S/W complexity ‘C’ of the FT-CORBA per object is assumed as ‘c’, then $C_{FT\text{-}CORBA}$ = cn. This is calculated as $C_{proposed\ architecture}$ = 2cn because the S/W complexity doubles where the DCOM objects and ORPC are implemented.

It is assumed that the processed information is all real-time in the evaluated Telematics and that fault tolerance is provided twice, by adding a backup protocol to the FT-CORBA to enhance service continuity. The Efficiency Index (EI) to evaluate performance improvement is as presented in formula (4), for the $CFT_{FT\text{-}CORBA}$ of the backup part is as ‘$CFT_{FT\text{-}CORBA}$ = (10nθ + nθ×t) × cn' and the $CFT_{proposed\ architecture}$ is as '$CFT_{proposed\ architecture}$ = (3nθ + nθ×t) × 2cn'.

$$\text{EI} = \frac{CFT_{proposed\ architecture}}{CFT_{FT-CORBA}} = \lim_{t\to 0}\frac{(3n\theta + n\theta t)\times 2cn}{(10n\theta + n\theta t)\times cn} = 0.6 \tag{4}$$

This is calculated as EI = 0.6, for ‘t’ is the service processing time of an object and can be approximated to 0, in order to compare efficiency between the FT-CORBA and the proposed architecture. Accordingly, higher efficiency of 40% can be achieved by implementing the FT-Agent and ORPC of DCOM in addition to the FT-CORBA than ‘2 × FT-CORBA’ to complement the limitations of the FT-CORBA. Efficiency improvement of 20% is achieved if it is normalized to compare the proposed architecture with the FT-CORBA itself.

$$\text{EI} = \frac{CFT_{(whole\ proposed\ architecture)}}{CFT_{(2\times FT-CORBA)}} = \lim_{t\to 0}\frac{\{(10n\theta + n\theta t)\times cn\}\times 0.7 + \{(3n\theta + n\theta t)\times 2cn\}\times 0.3}{\{(10n\theta + n\theta t)\times cn\}\times 0.7 + \{(10n\theta + n\theta t)\times cn\}\times 0.3} = 0.88 \tag{5}$$

The EI is as formula (5) where the 30% of clients request real time services and the remaining 70% do not. Accordingly, the degree of improvement is smaller than when all clients request real time services, and improvement of 12% occurs. The first and latter halves of addition to the numerator reveal the implementation costs of the FT-CORBA and the proposed architecture, respectively. The denominator reveals that the FT-CORBA protocol is implemented twice, as described earlier.

The computer that the FT-Agent and the RM operate on requires performance of formula (6) because look-ups of Table 1 in Section 3.3, one of the DBs of the FT-Agent occur for the time window of an hour only where the number of replicas of each object reduces from NO = 3 to NO = 2.

$$\sum_{i=1}^{m}\left(\frac{2}{5}\times t_{(mean-lookup)i}\right) < 5\text{minutes} \tag{6}$$

($t_{(mean\text{-}lookup)\ i}$: Av. lookup time by objects for the time window of an hour;
m : maximum number of objects to provide real time services simultaneously)

5 Conclusion

The characteristics of the Telematics and the corresponding necessity of the FT-CORBA are presented in this paper. This paper observes the limitations of the current FT-CORBA for the Telematics, presenting a method to improve them with agents. From the result of performance evaluation, the improvements are as follows.

First, the proposed architecture reduces service delays and the possibility of fault occurrences by reducing network traffic in terms of the entire network using MAs when objects, which provide real time services and are requested frequently by clients, respond to their requests.

Second, the preset minimum value of the number of replicas is adjusted dynamically, unlike that in the FT-CORBA, and therefore the operational efficiency of the CORBA based Telematics is enhanced.

The proposed architecture improves the first two limitations of the current FT-CORBA in the Telematics, but does not improve the other six limitations as described earlier. Therefore, further research is required in order to improve the limitations of the FT-CORBA based Telematics.

References

1. K. Fischer, J. P. Muller, and M. Pischel, "Unifying Control in a Layered Agent Architecture" Technical Memo TM-94-05, DFKI GmbH, pp.3, Saarbrucken, Jan.1995
2. A. Gokhale, B. Natarajan, D. C. Schmidt, and J. Cross, "Towards Real-time Fault-Tolerant CORBA Middleware", Cluster Computing: the Journal on Networks, Software, and Applications Special Issue on Dependable Distributed Systems, Vol.7, No.4, pp.15, Oct.2004.
3. C. C. Guan, and S. L. Li, "Architecture of traffic.smart," the 8th World Congress on Telematics, pp.2-5, Oct.2001
4. Isis Distributed Systems Inc. and Iona Technologies Limited. Orbix+Isis Programmer's Guide, 1995.
5. Korea Research Institute for Human Settlements, “National ITS Architecture”, pp. 359- 378, Dec.1999.
6. S. Maeis, “Adding group communication and fault tolerance to CORBA”, In Proceedings of the 1995 USENIX Conference on Object-Oriented Technologies, pp.135-146, Monterey, CA, 1995.
7. P. Narasimhan, "Transparent Fault Tolerance for CORBA", Ph. D. thesis, Department of Electrical and Computer Engineering, University of California, Santa Barbara, pp.10-12, Dec.1999.
8. B. Natarajan, A. Gokhale, and S. Yajnik, "DOORS: Towards High-performance Fault Tolerant CORBA", the Proceedings of the 2nd Distributed Applications and Objects (DOA) conference, pp.1-2., Antwerp, Belgium, Sept.21, 2000
9. J. L. Neto, A. D. Santos, C. A.A. Kaestner, and A. A. Freitas, "Document Clustering and Text Summarization," In Proceedings, 4th Int. Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), pp.1-4, 2000
10. Object Management Group, "CORBA Revision 2.3", Jun.1999
11. Object Management Group, "CORBA Version 3.0.3", OMG Document formal/04-03-21 edition, Mar.2004

12. Object Management Group, "The Common Object Services specification", OMG Technical Committee Document formal/98-07-05, Jul.1998.
13. Recursion Software, Inc., "Voyager", http://www.recursionsw.com/voyager.htm, 2006
14. S. Russel, and P. Norvig, "Artificial Intelligence", pp.40-45, Prentice-Hall, Inc., 1995
15. D. C. Schmidt, "Applying Patterns to Improve the Performance of Fault Tolerant CORBA", the 7th International Conference on High Performance Computing, ACM/IEEE, pp.1-4. Bangalore, India, pp.17-20, Dec.2000
16. Z. Tari, O. Bukhres, "Fundamentals of Distributed Object Systems", pp.20-25, John Wiley & Sons, Inc., 2001
17. U.S. Department of Transportation, “NTCIP 2305 (Draft) - National Transportation Communications for ITS Protocol (NTCIP) - Application Profile for Common Object Request Broker Architecture (CORBA),” Intelligent Transportation Systems Standards Fact Sheet, Aug.2001.
18. Vehicle Information and Communication System Center, "VICS units reach 3.2 million", the 8th World Congress on Telematics, pp.1-2, Oct.2001
19. Wikimedia Foundation, Inc., http://en.wikipedia.org/wiki/Telematics, Dec.2006

Multi-agent Based Integration of Production and Distribution Planning Using Genetic Algorithm in the Supply Chain Management

Byung Joo Park, Hyung Rim Choi*, and Moo Hong Kang

Department of Management Information Systems, Dong-A University
840 Hadan-dong, Saha-gu, Busan 604-714, Korea
{a967500, hrchoi, mongy}@dau.ac.kr

Abstract. Companies are pursuing a new approach through which overall functions within a supply chain, ranging from material purchase to production, distribution, and sales are designed, planned, and managed in an integrated way. The core functions among those functions are production planning and distribution planning. In this paper, we propose a multi-agent system and genetic algorithm that make it possible to integrate the production and distribution planning on a real-time basis in supply chain management.

Keywords: Supply chain management, multi-agent system, genetic algorithm.

1 Introduction

SCM (supply chain management) planning and operation includes the mutually related partial problems of the following – purchase, production, scheduling, distribution, inventory, network design, location and assignment, and logistics. And these problems should be dealt with integrated manner on the whole aspect. Active researches on the integration and coordination of these partial problems are being made. The core partial problems in the SCM are production planning and distribution planning. In order to satisfy the customers' demand, production planning has to deal with the problem of which manufacturer to produce, when to produce, and how much to produce for the orders placed by the customers. The distribution planning deals with the problem of deciding a product delivery channel from manufacturer to distributor or customer. As these problems are mutually related, they should be dealt with simultaneously in an integrated manner. Many researches to deal with the integrated problems of production and distribution planning have been made in the SCM. MIP (mixed integer programming) and a simulation technique were often used for the integration of production/distribution planning [2, 4]. Erenguc et al. [3] discussed about the matters that must be considered for optimization of the production/distribution, when supplier, factory, and distribution shops are separated. They combined an analytical model and a simulation model to integrate all the stages in the supply chain. Some researcher applied GA (genetic algorithm) for integrated planning in SCM.

* Corresponding author.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 696–706, 2007.
springerlink.com

Dellaert [1] presented a method to represent the problem with chromosome, when a GA is applied to the multi-level production problem.

Recently, many enterprises are moving toward an open architecture for integrating their activities with their suppliers, customers and other partners within the supply chain. Agent-based technology provides an effective approach in such environments [7]. Multi-agent systems have been proven suitable to represent domains such as supply chain networks which involve interactions among manufacturing organization, their customers, suppliers, etc. with different individual goals and propriety information [9]. Zhang et al. [9] presents an agent-based manufacturing system that enable manufacturing organizations to dynamically and cost-effectively integrate, optimize, configure, simulate, restructure and control not only their own manufacturing systems but also their supply networks, in a coordinated manner to cope with the dynamic changes occurring in a global market. Turoski [8] developed agent-based techniques for coordinating activities of e-commerce and internet-based supply chain system for mass customization market. Li and Fong [5] proposed agent-based architectures to facilitate the formation and organization of virtual enterprise for order management. SCM is large-scale and multi-stage problems. Also, its various kinds of internal or external factors (e.g. machine failure, production delay, and order change) can, at any time, dynamically bring a change to the existing plan or situation. In this study, we propose a multi-agent system and genetic algorithm that make it possible to integrate production and distribution planning on a real-time basis in SCM.

2 Multi-agent System for Integration of Production and Distribution Planning in SCM

2.1 Design of a Multi-agent System

SCM has to deal with the dynamic environments that include diverse changes in a supply chain. The integrated planning under an SCM environment has to take into consideration all the suppliers, manufacturers, and distributors simultaneously, as well as their environmental changes, so that it can change and coordinate its plan on a real-time basis. If a manufacturer cannot produce a specific product at a certain time due to its machine failure, this information has to be reflected immediately in the integrated planning. In order to obtain the information on current situation and capacity of the suppliers, manufacturers, and distributors, and to make a new plan in response to the new environmental changes, this study used a multi-agent system.

On the basis of integrated planning agent of production and distribution using genetic algorithm, the multi-agent system suggested in this study enables the suppliers, manufacturer and distributor to make an optimal transportation plan, production plan, supply plan, and inventory management and to help to make the optimal supply chain plan through sharing information on inventory and demand on a real-time basis. Furthermore, this system enables speedy and efficient re-planning in response to the dynamic environmental changes taking place in the SCM, consequently making it possible to modify its operational plan within the supply chain.

The multi-agent system has the following agents: The distributor agent collects the order information from the customers by period, and also manages the information on the changes of orders on a real-time basis. The manufacturer agent manages the information on production capacity, production possibility of an item, observance of due date, and environmental changes such as production delay caused by machine failure. The supplier agent manages the information on the supplier's supply capacity, supply possibility of an item, and environmental changes related to observance of due date. Besides, there are three more agents: the supplier management agent, manufacturer management agent, and distributor management agent. These three agents communicate with the above-mentioned agents, while providing to them the information received from the mediator. Also, there is a mediator, the central agent in the multi-agent system, which coordinates all the other agents and controls the message exchange. The multi-agent system still involves the integrated SCM planning agent, which has a GA solver performing the integrated planning of production and distribution. While making the mediator play the central role, this multi-agent system enables each agent to exchange messages and perform their own function. The structure of the multi-agent system is shown in Fig. 1.

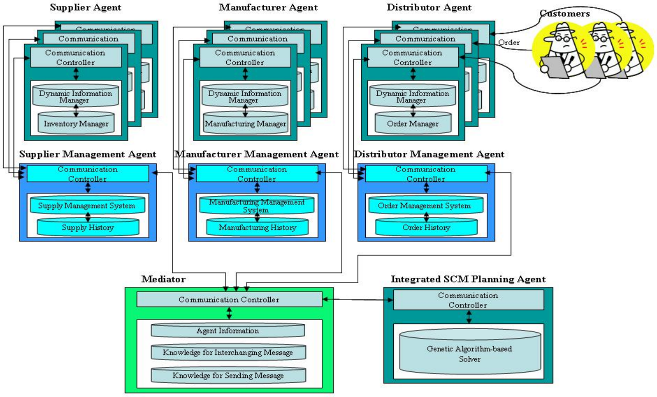


Fig. 1. The architecture of the multi-agent system

2.2 Function of Agents in the Multi-agent System

(1) Mediator
Mediator performs the role of coordinating and controlling the message exchange of each agent. In order to support the efficient performance of all the agents and to help their message exchange, a central agent who can coordinate and control them is vitally necessary. The mediator has information on each agent and knowledge base on the control and message exchange of each agent.

(2) Supplier Agent (SA)
SA enables the supplier to communicate with other agents within the multi-agent system. In order to speedily provide information on the suppliers' capacity by period, the SA has the information on the inventory and the supply capacity of each supplier by

period. Also, the SA receives the information on the environmental changes suddenly happened to the supplier through user interface, and sends it to the supplier management agent.

(3) Manufacturer Agent (MA)
MA enables the manufacturer to communicate with other agents within the multi-agent system. In order to speedily provide the information on the production capacity of each manufacturer by period, the MA has the information on the current work situation, waiting work, and production capacity by period. Also, as it has the information on the possible outsourcing companies by item and by period, it helps make judgment on the production capacity and production possibility of the manufacturers. The MA receives the information on the environmental changes suddenly happened to the manufacturers through user interface, and sends it to the manufacturer management agent.

(4) Distributor Agent (DA)
DA enables the distributor to communicate with other agents within the multi-agent system. The DA receives the orders by period from customers, and sends the information to the distributor management agent. In order to speedily provide the information on the distribution capacity of the distributors by period, it has information on the inventory by period and by item. Also, it receives through user interface the information on the environmental changes including an order cancellation that suddenly happened to the customers, and sends them to the distributor management agent.

(5) Supplier Management Agent (SMA)
SMA performs the role of exchanging a message with the SA. The SMA collects the information on the environmental changes happened to the suppliers, and sends them to the mediator so that it may be reflected in the integrated planning. Also, it receives from the mediator the information on the supply quantity and supply schedule, and then send it to the SA. It keeps record of the suppliers' supply history and their observance of supply schedule. Based on this record, it assesses the reliability of the suppliers. The suppliers who are below a certain level of reliability are to be excluded from the integrated planning.

(6) Manufacturer Management Agent (MMA)
MMA performs the role of exchanging a message with the MA. The MMA collects on a real-time basis the information on the environmental changes happened to the manufacturers, and sends them to the mediator so that it may be reflected in the integrated planning. Also, it receives from the mediator the information on production schedule and production quantity by manufacturers, and sends it to the MA. It keeps record of the manufacturers' production history and their observance of due date. Based on this record, it assesses the reliability of the manufacturers. And the manufacturers who are below a certain level of reliability are to be excluded from the integrated planning.

(7) Distributor Management Agent (DMA)
DMA performs the role of exchanging a message with the DA. The DMA collects on a real-time basis the information on the environmental changes including the change

of orders from the customer and sends it to the mediator so that it may be reflected in the integrated planning. Also, it receives from the mediator the information on the time of placing orders and the quantity of orders by distributors and sends it to the DA. It keeps account of the information on the customers' orders so that it may be used for analysis of customer's purchase pattern.

(8) Integrated SCM Planning Agent (ISPA)
Based on the information on the customer's demand collected by the DMA, the ISPA analyzes the distributor's time of placing orders and order quantity, manufacturer's production time and production quantity, and supplier's supply time and supply quantity, making a transportation plan between the supplier and manufacturer and between the manufacturer and distributor, and finally making it possible to reschedule in response to the dynamic environmental changes. The ISPA is a core agent with the genetic algorithm solver that makes possible the integrated planning for SCM. The performance of the genetic algorithm for integrated planning has been proved through tests in the section 4.

3 Design of a Genetic Algorithm for the Integration of Production and Distribution Planning

GA are the most well-known and robust methods. GA deals with coding of the problem using chromosomes as representative elements, instead of decision making valuables handled in a definite method. To represent the integration problem of production/distribution handled in this study with one chromosome, the chromosome has to represent all the components in the process of supply/production/distribution, and then the pattern of supply/production/distribution should be able to be decided.

3.1 Representation of a Chromosome

The most important stage in designing a GA is to represent solution with chromosome. A chromosome should be able to reflect the features of a problem and represent them properly, and produce a more suitable solution for the objective function, through an evolutionary process, by a genetic operator. This study uses the chromosome representation suitable for each supplier, manufacturer, and distributor for the integrated planning of production and distribution. Fig. 2 demonstrates the example of a represented chromosome for the integrated planning. For example, if there are two distributors, two manufacturers, two suppliers, and two products ordered by each distributor, respectively, and periods are divided into five, the representation of the chromosome is exhibited as in Fig. 2. In the chromosome composed of four rows in Fig. 1, the first row is the representation of the chromosome to decide a distributor's order quantity at each period. The second row is the representation to decide a manufacturer to produce the ordered products at each period. It has the random numbers used to determine a manufacturer. The random number is produced within the number of maximum manufacturer to be able to produce the product. The number of manufacturers is decided by the production possibility information of manufacturer obtained from the multi-agent system. The third row is to decide production output of a manufacturer and the fourth row is to select a supplier to supply a raw material. Also, the information of a supplier is obtained from the multi-agent system.

1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1
1	1	2	1	2	2	1	2	2	1	2	1	1	2	1	2	2	1	1	1

1	1	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0
1	2	1	2	2	1	1	2	1	2	2	2	1	1	2	1	2	2	1	1

Fig. 2. Representation of a Chromosome

When demand information of each distributor for each product in each period was collected, the chromosome represented in Fig. 2 is interpreted as demonstrated in Fig. 3 so as to select the quantity of order and period to order, a manufacturer to produce and output quantity, and a supplier to supply raw materials. To decide the quantity of order and period to order from a distributor, a binary representation method was used [1]. If the value of the gene represented in each period is 1, the quantity of order equivalent to the demand quantity in the concerned period is ordered, and if not, the order is not ordered. For instance, the first gene in first row of the chromosome has the value of 1, this means that distributor 1 has to order for product 1 in period 1 equivalent to demand quantity. For the gene value of 0, it is interpreted that order is simultaneously placed in the previous period, whose gene value is 1. In Fig. 2, the value of the third gene in the first row is 0. This means that an order is not placed in period 3, but is placed in period 2 added with demand quantity in period 3. Therefore, the quantity of 230 is ordered in period 2. The distributor holds 100 units of product 1 in period 2. Accordingly, additional inventory cost is generated. After the quantity of order and period to order are decided. Manufacturer producing the order quantity has to be decided through chromosome in the second row of manufacturer part in Fig. 2. Manufacturers can be decided through the genes represented through the manufacturer's number. For example, the first gene value is 1 in the second row. This means that manufacturer 1 produces ordered quantity 120 for product 1 which has been ordered by distributor 1. Each manufacturer's production period and output is decided through the chromosome in the third row. This chromosome uses the same binary representation and interpretation methods which were used to decide the quantity of an order and period to order. For an instance, the fourth gene in the third row is 1. This means producing order quantity of 180 in period 4. Meanwhile, the gene value is 0 in the period 5, and this means producing order quantity of 330 in period 4 added with quantity of 150 from period 5. In the third row, the period 1 and 2 of product 2 of manufacturer 1 were no output that manufacturer has to produce, but the representation of the chromosome is 1. In this case, output is interpreted as 0, when there is nothing to actually produce. However, when production is not conducted in the remaining periods, the output is decided by adding all the output in those periods. While reading the chromosome for the period for allocation of output, in the period that is represented as 1, the production output needed in the period and the following periods represented as 0 is produced. In the chromosome representation in Fig. 2, in the case that only period 1 and 2 are represented as 1 among five periods, it is interpreted that output of 170 is produced in period 2, which is needed in the following periods. In period 1, it is interpreted that no production is made, because there is no required output. The chromosome in the fourth row is to select a supplier. This also indicates the number of a supplier in the same way used to select a manufacturer above. The first gene 1 in the fourth row means that manufacturer 1 receives raw materials required for production of 120 from supplier 1.

	Distributor 1										Distributor 2									
	Product 1					Product 2					Product 1					Product 2				
Period	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Demand Quantity	120	130	100	100	80	60	50	50	80	50	200	180	180	200	150	30	50	40	40	40
Chromosome	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1
Quantity of Order	120	230	0	180	0	110	0	50	80	50	200	560	0	0	150	80	0	80	0	40
Chromosome	1	1	2	1	2	2	1	2	2	1	2	1	1	2	1	2	2	1	1	1
	Manufacturer 1										Manufacturer 2									
	Product 1					Product 2					Product 1					Product 2				
Production Quantity	120	230 560	0	180	150	0	0	80	0	50 40	200	0	0	0	0	110 80	0	50	80	0
Chromosome	1	1	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0
Output	120	790	0	330	0	0	170	0	0	0	200	0	0	0	0	240	0	0	80	0
Chromosome	1	2	1	2	2	1	1	2	1	2	2	2	1	1	2	1	2	2	1	1

Fig. 3. Interpretation of Chromosome Representation

In the representation of a chromosome to decide a distributor's period to order in first row, 1 is represented in the first period for each product so that an order can always be placed. The reason is because no inventory was assumed in the previous period. In the decision regarding production periods and outputs of a manufacturer in third row, 1 is represented in the first period for each product so that production can always be conducted at the period. When no inventory was assumed in the previous period, it is to make a solution that can be executed.

3.2 Genetic Operator

This study applied the crossover which can inherit parents' chromosome order and type properly. Once parent 1 and parent 2 are selected, this crossover randomly generates number 1 and 2, which indicate parents, as many as the length of chromosome genes in order to inherit genes from one parent of the two parents. The length of the chromosome corresponds to the longer chromosome between the result of (number of distributors × number of products × number of period) and the result of (number of manufacturers × number of products × number of period). And then, a new chromosome is generated by inheriting all the genes within the column from the parent who corresponds to the random number. This process is conducted again by changing parent 1 and parent 2. Between two children generated in the above mentioned manner, one, which has better value of objective function, is sent to the next generation.

This study uses a mutation method in which genes of all the products in relation to a distributor and a manufacturer in each period are mutually exchanged with the genes of another distributor and another manufacturer. When multi-distributors and manufacturers exist, genes are mutually exchanged by randomly selecting distributors and manufacturers.

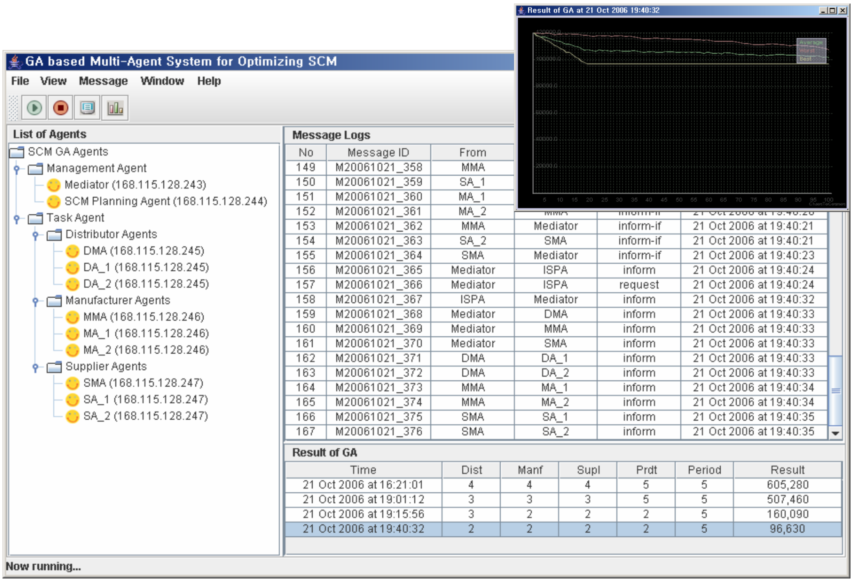
3.3 Selection Method and Objective Function

As a selection method, a seed selection is used [6]. The seed selection has introduced preservation method of good individuals in evolutionary process. A seed, which corresponds to a father, is randomly selected among excellent individual group that are within top rank in a group. The mother individual is randomly selected from the entire group. These are used as parents, and they are returned to their own group, so that they can be selected again. The next generation is newly formed by using a selection method from the current generation and by using genetic operators. After that, bad individuals are replaced by good ones in equal number by applying elitism.

Integrated planning using minimum cost in SCM means an efficient operation for distributors, manufacturers, and suppliers. The value of objective function is obtained through reading genes row by row from left to right. The total cost of the following is calculated: cost of order, inventory cost, transport cost from a manufacturer to a distributor according to the selection of the manufacturer, production preparation cost and inventory cost according to output of a manufacturer in each period, and the price of raw materials supplied by a supplier and transport cost from a supplier to a manufacturer, according to the selection of a supplier. The object is to select an integrated plan with minimum total cost.

4 Prototype Implementation

The development of the multi-agent system has focused on the efficient performance of the message exchange among agents, and its structure is based on the AP (Agent Platform) structure of the FIPA that was adopted as an international standard. If an agent within the structure wants to communicate with the other agent within the same platform or another platform, it has to be registered at least in one of the platforms. AP provides a service for cooperation among agents belonging to the platforms. AP has the following components: ACC (Agent Communication Channel), ANS (Agent Name Server), DF (Directory Facilitator), and AMS (Agent Management System). The ACC supports the message transmission among agents within the AP and the communication between the APs. The ANS provides the information on the name and address of each agent for transmission. The DF provides each agent with the information on the capacity and services of all the agents within the AP. AMS performs the registration, removal, temporary suspension and recovery of the agents within the AP. Based on the AP structure, this paper has made use of a mediator in place of AMS, and DF performs the role of ontology for understanding exchanging message among the agents and has the form of a simple database. Fig. 4 shows a snapshot of the multi-agent system for integrated planning in SCM.



List of Agents

- SCM GA Agents
  - Management Agent
    - Mediator (168.115.128.243)
    - SCM Planning Agent (168.115.128.244)
  - Task Agent
    - Distributor Agents
      - DMA (168.115.128.245)
      - DA_1 (168.115.128.245)
      - DA_2 (168.115.128.245)
    - Manufacturer Agents
      - MMA (168.115.128.246)
      - MA_1 (168.115.128.246)
      - MA_2 (168.115.128.246)
    - Supplier Agents
      - SMA (168.115.128.247)
      - SA_1 (168.115.128.247)
      - SA_2 (168.115.128.247)

Message Logs

No	Message ID	From			
149	M20061021_358	MMA			
150	M20061021_359	SA_1			
151	M20061021_360	MA_1			
152	M20061021_361	MA_2	MMA	inform-if	21 Oct 2006 at 19:40:20
153	M20061021_362	MMA	Mediator	inform-if	21 Oct 2006 at 19:40:21
154	M20061021_363	SA_2	SMA	inform-if	21 Oct 2006 at 19:40:21
155	M20061021_364	SMA	Mediator	inform-if	21 Oct 2006 at 19:40:23
156	M20061021_365	Mediator	ISPA	inform	21 Oct 2006 at 19:40:24
157	M20061021_366	Mediator	ISPA	request	21 Oct 2006 at 19:40:24
158	M20061021_367	ISPA	Mediator	inform	21 Oct 2006 at 19:40:32
159	M20061021_368	Mediator	DMA	inform	21 Oct 2006 at 19:40:33
160	M20061021_369	Mediator	MMA	inform	21 Oct 2006 at 19:40:33
161	M20061021_370	Mediator	SMA	inform	21 Oct 2006 at 19:40:33
162	M20061021_371	DMA	DA_1	inform	21 Oct 2006 at 19:40:33
163	M20061021_372	DMA	DA_2	inform	21 Oct 2006 at 19:40:33
164	M20061021_373	MMA	MA_1	inform	21 Oct 2006 at 19:40:34
165	M20061021_374	MMA	MA_2	inform	21 Oct 2006 at 19:40:34
166	M20061021_375	SMA	SA_1	inform	21 Oct 2006 at 19:40:35
167	M20061021_376	SMA	SA_2	inform	21 Oct 2006 at 19:40:35

Result of GA

Time	Dist	Manf	Supl	Prdt	Period	Result
21 Oct 2006 at 16:21:01	4	4	4	5	5	605,280
21 Oct 2006 at 19:01:12	3	3	3	5	5	507,460
21 Oct 2006 at 19:15:56	3	2	2	2	5	160,090
21 Oct 2006 at 19:40:32	2	2	2	2	5	96,630

Fig. 4. The snapshot of multi-agent system for integrated planning in SCM

4.1 Problem Definition for Integration of Production and Distribution Planning

In this study, raw materials are supplied from multiple suppliers to multiple manufacturers, and various products produced by each manufacturer are supplied to various distributors. Here, a distributor can be understood as a logistics warehouse delivering finished goods from a manufacturer to a customer. We assume that each manufacturer can produce various products, and can produce all the products ordered within each period. Suppliers are also assumed to be able to supply raw materials or parts in each period. Also, many distributors exist in the SCM environment. These distributors receive orders from many customers, and collect those orders and give orders to manufacturers. In this study, demand in the concerned periods is assumed to be known for several periods in the near future. Each distributor can control inventory cost and cost of orders through adjustment of order quantity, and can take products with minimum transport cost and production cost, due to selection of an adequate manufacturer. Manufacturers can control production preparation cost and inventory cost through the adjustment of production output, and can take materials and parts at minimum prices and transport cost with a selection of an adequate supplier. The objective function in the integrated planning problem is to minimize total cost of the cost of order, inventory cost, production preparation cost, production cost of manufacturers, and the transport cost between manufacturers and distributors and between suppliers and manufacturers.

4.2 Performance Evaluation

The performance of integrated planning in SCM is exhibited through a comparison with the optimal solution and non-integrated planning. First, experiments were conducted to decide the values of parameters of GA. As the values of the parameters used in this study, the size of population was 1000, number of generations 200, seed selection scope 300 and size of elitism 100, and mutation rate and cross rate 0.1, and 0.9, respectively. To generate a problem for evaluation, demand quantity was generated in the unit of 10 between 50 and 200, and the order cost of a distribution center, and inventory cost per period at a factory and a distribution center were set 100 and 5 respectively. Transport cost from a factory to a distribution center was set between 3 and 5, and transport cost from a supplier to a manufacturer was generated between 10 and 15 at integer value. Production cost at each manufacturer was generated between 35 and 40. Table 1 compares the optimal solution obtained in the small size problem with the results obtained through GA-based integrated planning. To obtain the optimal solution of a mathematical model, the ILOG CPLEX 10.0 package was used. The GA was developed based on Java 2 Standard Edition 1.5. The solutions could not be obtained using ILOG in the problems bigger than problem 5 in Table 1. The results of non-integrated planning are the best solutions obtained in 100 repeated experiments. As shown in Table 1, there is a great difference in the results between integrated planning and non-integrated planning. The proposed genetic algorithm produced near optimal solution.

Table 1. Results of performance evaluation for proposed GA

Problem	Problem Size					Optimal Solution	Integrated Planning	Non- Integrated Planning
	# of Distributor	# of Product	# of Manufacturer	# of Supplier	# of Period			
1	2	2	2	2	5	96,630	96,630	107,268
2	3	2	2	2	5	160,090	160,090	175,665
3	2	2	3	2	5	96,630	96,630	107,588
4	3	5	3	3	5	506,394	507,460	534,198
5	4	5	4	4	5	603,997	605,280	659,906

5 Conclusion

This study has developed a methodology of integrated planning based on multi-agent system, by sharing information on the suppliers, manufacturers, distributors, and consumers within the supply chain, and also by using the genetic algorithm that on a real-time basis enables the integrated planning in the SCM including the quantity and timing of orders, the quantity and timing of production, supply quantity, and distribution pattern. Differently from the traditional method of trying to optimize many partial problems in sequence, the proposed method see partial problems simultaneously to seek a better solution, making it possible to respond to the dynamic environmental changes on a real-time basis, and consequently making SCM more efficient. In addition, the GA developed for integrated planning has been proved by performance tests that have been taken for various sizes of problems. Also, the speedy operation and simple structure of the GA is expected to bring practicality.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (The Regional Research Universities Program/Research Center for Logistics Information Technology)

References

1. Dellaert N, Jeunet J, Jonard N (2000) A GA to Solve the General Multi-level Lot-sizing Problem with Time Varying Costs. International Journal of Production Economics 68: 241-257
2. Dhaenens-Flipo C, Finke G (2001) An Integrated Model for Industrial Production-Distribution Problem. IIE Transaction 33:705-715
3. Erenguc SS, Simpson NC, Vakharia AJ (1999) Integrated Production/Distribution Planning in Supply Chains. European Journal of Operational Research 115:219-236
4. Fumero F, Vercellis C (1999) Synchronized Development of Production, Inventory and Distribution Schedules. Transportation Science 33:330-340
5. Li T, Fong Z (2003) A System Architecture for Agent-based Supply Chain Management Platform. Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE) : Toward a caring and human technology, Montreal, Canada
6. Park BJ, Choi HR, Kim HS (2003) A Hybrid GA for Job Shop Scheduling Problems. Computers and Industrial Engineering 45:597-613
7. Shen W, Hao Q, Yoon HJ, Norrie DH (2006) Applications of Agent-based Systems in Intelligent Manufacturing: An updated review. Advanced Engineering Informatics 20:415-431
8. Turoski K (2002) Agent-based e-commerce in case of Mass Customization. International Journal of Production Economics 75:69-81
9. Zhang DZ, Anosike AI, Lim MK, Akanle OM (2006) An Agent-based approach for | e-manufacturing and Supply Chain Integration. Computers and Industrial Engineering 51:343-360

Modeling and Simulation by Petri Networks of a Fault Tolerant Agent Node

Arnulfo Alanis Garza[1], Juan José Serrano[2], Rafael Ors Carot[2],
and José Mario García Valdez[1]

[1] Division of Graduate Studies and Research, Calzada Tecnologico, S/N, Tijuana, Mexico
{alanis,ocastillo,mario}@tectijuana.mx
[2] D. Inf. de Sistemas y Computadoras, Camí de Vera, s/n, 46022 València, España,
00+34 96387,Universidad Politécnica de Valencia (España)
{jserrano,rors}@disca.upv.es

Abstract. Intelligent Agents have originated a lot discussion about what they are, and how they are a different from general programs. We describe in this paper a new paradigm for intelligent agents, This paradigm helped us deal with failures in an independent and efficient way. We proposed these types of agents to treat the system in a hierarchical way. The Agent Node is also described. A new method to visualize fault tolerant system (FTS) is proposed, in this paper witch the incorporation of intelligent agents, which as they grow and specialized create the Agent System (MAS).One is the communications diagrams of the each of the agents, described in diagrams of transaction of states.

1 Introduction

At the moment, the approach of using agents for real applications, has worked with mobile agents, which work at the level of the client-server architecture. However, in systems where the requirements are higher, as in the field of the architecture of embedded industrial systems, the idea is to innovate in this area by working with the paradigm of intelligent agents. Also, it is a good idea in embedded fault tolerant systems, where it is a new and good strategy for the detection and solution of errors.

A rational agent is one that does the right thing. Obviously, this is better than doing the wrong thing, but what does it mean? As a first approximation, we will say that the right action is the one that will cause the agent to be most successful. That leaves us with the problem of deciding how and when to evaluate the agent's success.

By an agent-based system, we mean one in which the key abstraction used is that of an agent. In principle, an agent-based system might be conceptualized in terms of agents, but implemented without any software structures corresponding to agents at all. We can again draw a parallel with object-oriented software, where it is entirely possible to design a system in terms of objects, but to implement it without the use of an object-oriented software environment. But this would at best be unusual, and at worst, counterproductive. A similar situation exists with agent technology; we therefore expect an agent-based system to be both designed and implemented in terms of agents. A number of software tools exist that allow a user to implement software systems as agents, and as societies of cooperating agents.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 707–716, 2007.
springerlink.com

2 Agents

Let's first deal with the notion of intelligent agents. These are generally defined as "software entities", which assist their users and act on their behalf. Agents make your life easier, save you time, and simplify the growing complexity of the world, acting like a personal secretary, assistant, or personal advisor, who learns what you like and can anticipate what you want or need. The principle of such intelligence is practically the same of human intelligence. Through a relation of collaboration-interaction with its user, the agent is able to learn from himself, from the external world and even from other agents, and consequently act autonomously from the user, adapt itself to the multiplicity of experiences and change its behavior according to them. The possibilities offered for humans, in a world whose complexity is growing exponentially, are enormous [1][4][5][6].

We need to be careful to distinguish between rationality and omniscience. An omniscient agent knows the actual outcome of its actions, and can act accordingly; but omniscience is impossible in reality. Consider the following example: I am walking along the Champs Elys´ees one day and I see an old friend across the street. There is no traffic nearby and I'm not otherwise engaged, so, being rational, I start to cross the street. Meanwhile, at 33,000 feet, a cargo door falls off a passing airliner, and before I make it to the other side of the street I am flattened. Was I irrational to cross the street? It is unlikely that my obituary would read "Idiot attempts to cross street." Rather, this points out that rationality is concerned with expected success given what has been perceived. Crossing the street was rational because most of the time the crossing would be successful, and there was no way I could have foreseen the falling door. Note that another agent that was equipped with radar for detecting falling doors or a steel cage strong enough to repel them would be more successful, but it would not be any more rational [23].

3 Petri Nets

A Petri net is a graphical and mathematical modeling tool. It consists of places, transitions, and arcs that connect them. Input arcs connect places with transitions, while output arcs start at a transition and end at a place. There are other types of arcs, e.g. inhibitor arcs. Places can contain tokens; the current state of the modeled system (the marking) is given by the number (and type if the tokens are distinguishable) of tokens in each place. Transitions are active components. They model activities which can occur (the transition fires), thus changing the state of the system (the marking of the Petri net). Transitions are only allowed to fire if they are enabled, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places). When the transition fires, it removes tokens from its input places and adds some at all of its output places. The number of tokens removed / added depends on the cardinality of each arc. The interactive firing of transitions in subsequent markings is called token game.

Petri nets are a promising tool for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of systems.

To study performance and dependability issues of systems it is necessary to include a timing concept into the model. There are several possibilities to do this for a Petri net; however, the most common way is to associate a firing delay with each transition. This delay specifies the time that the transition has to be enabled, before it can actually fire. If the delay is a random distribution function, the resulting net class is called stochastic Petri net. Different types of transitions can be distinguished depending on their associated delay, for instance immediate transitions (no delay), exponential transitions (delay is an exponential distribution), and deterministic transitions (delay is fixed) [20].

4 FIPA (The Foundation of Intelligence Physical Agents)

FIPA specifications represent a collection of standards, which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. The life cycle [9] of specifications details what stages a specification can attain while it is part of the FIPA standards process. Each specification is assigned a specification identifier [10] as it enters the FIPA specification life cycle. The specifications themselves can be found in the Repository [11].

The Foundation of Intelligent Physical Agents (FIPA) is now an official IEEE Standards Committee.

4.1 FIPA ACL Message

Over time, failure has come to be defined in terms of specified service delivered by a system. This avoids circular definitions involving essentially synonymous terms such as defect, etc. This distinction appears to have been first proposed by Melliar-Smith [22]. A system is said to have a failure if the service it delivers to the user deviates from compliance with the system specification for a specified period of time. While it may be difficult to arrive at an unambiguous specification of the service to be delivered by any system, the concept of an agreed-to specification is the most reasonable of the options for defining satisfactory service and the absence of satisfactory service, failure.

The definition of failure as the deviation of the service delivered by a system from the system specification essentially eliminates "specification" faults or errors. While this approach may appear to be avoiding the problem by defining it away, it is important to have some reference for the definition of failure, and the specification is a logical choice. The specification can be considered as a boundary to the system's region of concern, discussed later. It is important to recognize that every system has an explicit specification, which is written, and an implicit specification that the system should at least behave as well as a reasonable person could expect based on experience with similar systems and with the world in general. Clearly, it is important to make as much of the specification as explicit as possible.

It has become the practice to define faults in terms of failure(s). The concept closest to the common understanding of the word fault is one that defines a fault as the adjudged cause of a failure. This fits with a common application of the verb form of the word fault, which involves determining cause or affixing blame. However, this requires an understanding of how failures are caused. An alternate view of faults is to consider them failures in other systems that interact with the system under consideration--either a subsystem internal to the system under consideration, a component of the system under consideration, or an external system that interacts with the system under consideration (the environment). In the first instance, the link between faults and failures is cause; in the second case it is level of abstraction or location.

The advantages of defining faults as failures of component/interacting systems are: (1) one can consider faults without the need to establish a direct connection with a failure, so we can discuss faults that do not cause failures, i.e., the system is naturally fault tolerant, (2) the definition of a fault is the same as the definition of a failure with only the boundary of the relevant system or subsystem being different. This means that we can consider an obvious internal defect to be a fault without having to establish a causal relationship between the defect and a failure at the system boundary.

In light of the proceeding discussion, a fault will be defined as the failure of (1) a component of the system, (2) a subsystem of the system, or (3) another system which has interacted or is interacting with the considered system. Every fault is a failure from some point of view. A fault can lead to other faults, or to a failure, or neither.

A system with faults may continue to provide its service, that is, not fail. Such a system is said to be fault tolerant. Thus, an important motivation for differentiating between faults and failures is the need to describe the fault tolerance of a system. An observer inspecting the internals of the system would say that the faulty component had failed, because the observer's viewpoint is now at a lower level of detail [21].

5 The KQML Language

Communication takes place on several levels. The content of the message is only a part of the communication. Begin able to locate and engage the attention of someone you want to communicate with is apart of the process. Pack-aging your message in a way which makes your purpose in communicating clear is another.

When using KQML, a software agent transmits content messages, composed in a language of its own choice, wrapped inside of a KQML message. The content message can be expressed in any representation language and written in either ASCII strings or one of many binary notations (e.g. network independent XDR representations).All KQML implementations ignore the content portion of the message except to the extent that they need to recognize where it begin sand ends.

The syntax of KQML is based on a balanced parenthesis list. The initial element of the list is the performative and the remaining elements are the performative's arguments as keyword/value pairs. Because the language is relatively simple, the actual syntax is not significant and can be changed if necessary in the future. The syntax reveals the roots of the initial implementations, which were done in Common Lisp, but has turned out to be quite flexible.

KQML is expected to be supported by a software substrate, which makes it possible for agents to locate one another in a distributed environment. Most current implementations come with custom environments of this type; these are commonly based on helper programs called routers or facilitators. These environments are not a specified part of KQML. They are not standardized and most of the current KQML environments will evolve to use some of the emerging commercial frameworks, such as OMG's CORBA or Microsoft's OLE2, as they become more widely used.

The KQML language supports these implementations by allowing the KQML messages to carry information which is useful to them, such as the names and addresses of the sending and receiving agents, a unique message identifier, and notations by any intervening agents. There are also optional features of the KQML language which contain descriptions of the content: its language, the ontology it assumes, and some type of more general description, such as a descriptor naming a topic within the ontology. These optional features make it possible for the supporting environments to analyze, route and deliver messages based on their content, even though the content itself is inaccessible [17].

6 Agent Communication Protocols

There are a variety of interprocess information exchange protocols. In the simplest case, one agent acts as a client and sends a query to another agent acting as a server and then waits for a reply, as is shown between agents A and B in Figure 1. The server's reply might consist of a single answer or a collection or set of answers. In another common case, shown between agents A and C, the server's reply is not the complete answer but a handle which allows the client to ask for the components of the reply, one at a time. A common example of this exchange occurs when a client queries a relational database or a reasoner which produces a sequence of instantiations in response. Although this exchange requires that the server maintain some internal state, the individual transactions are as before - involving a synchronous communication between the agents. A somewhat different case occurs when the client subscribes to a server's output and an indefinite number of asynchronous replies arrive at irregular intervals, as between agents A and D in Figure 1. The client does not know when each reply message will be arriving and may be busy performing some other task when they do.

There are other variations of these protocols. Messages might not be addressed to specific hosts, but broadcast to a number of them. The replies, arriving synchronously or asynchronously have to be collated and, optionally, associated with the query that they are replying to [18].

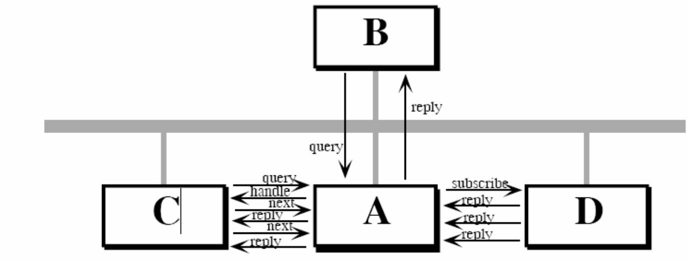


Fig. 1. Several basic communication protocols are support in KQLM

7 Proposed Method

Lets suppose we have a Distributed System (mainly applied to the industrial control), which is made up of a set of Nodes, where each one of them can be constituted by several Devices.

On these Nodes a set of ordered Tasks, is executed all of them to take I finish the functionality of the system. In order to identify this Distributed System the following definitions are set out:

Definition 1. = is N {Nor}, the set of the Nodes of the system, being "n" is the number of units that integrate it.

Definition 2. Be [Di, z], the set of devices that contains Node i. Where "z" can take value 1, if it is wanted to see the Node like only device, or greater than 1 if it is desired to do visible some of the elements that integrate it.

Definition 3. = is T {Tj}, the set of tasks that are executed in the system, being "t" the number of tasks that integrate the system.

Definition 4. A System Distributed like dupla is defined: SD = (N, T) Once characterized what a Distributed System could be denominated Basic (without no characteristic of Tolerance to Failures), one is going away to come to the incorporation on he himself from the paradigm of Intelligent Agents with the purpose of equipping it with a layer with Tolerance to Failures. The Fault tolerant Agents will define themselves now that worked in the SD.

Definition 5. An Agent is ANi to whom Agent denominated itself Node, whose mission is the related one to the tolerance to failures at level of the Node Nor.

Definition 6. {ANi is AN=} a set of Agents Node.

Definition 7. An application of N to AN of form of a is α each node Nor, of the system associates an Agent to him ANi Node, that is to say: α: N→ANi Nor → ANi.

Definition 8. A System is SMATF Fault tolerant Multi-Agent, formed by tripla of AN, AT, AS. That is to say With it a Distributed System Fault tolerant SDTF is defined as: SMATF = <N, AT, AS>.

Definition 9. A Distributed System Fault tolerant SDTF like dupla is defined SDTF =<SD, SMATF>.

The network of Petri of the Agent Node, fig 2, the exit, communication variables is of red color, and the communication variables, of entrance, are of green color. Which interrelate with the agent to whom it must send or receive information to him.

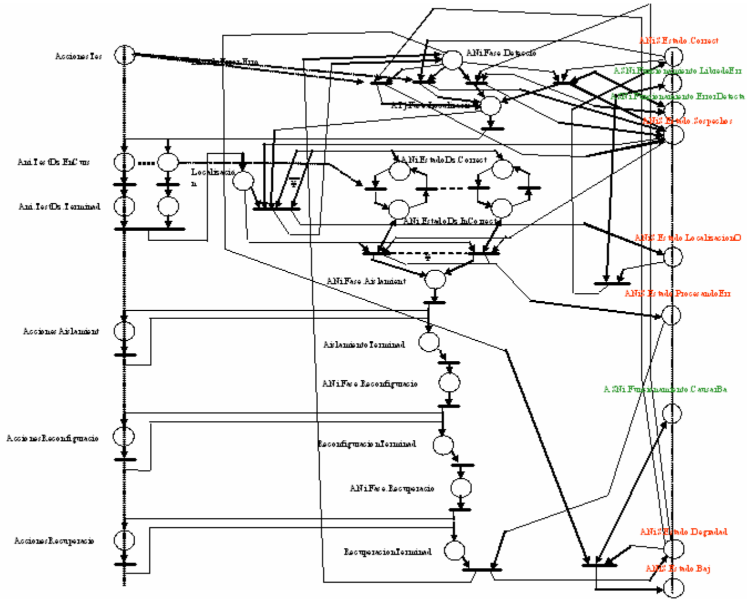


Fig. 2. Petri Networks of the Fault Tolerant Agent Node

8 Communications Diagram

Now we show diagram of transition of states of the Agent Node.

In figure 2 we show the diagram of transition of states, one is, Agent Node (AN), its operation and interchange of messages, as well as the variables that take part in the passage of their internal communication, in addition to concexion with one of the agents of the SMA, the agent system (AS), which as well in its internal states of communication and also, handled its communication with the other agent of the SMATF, the Agent Task.

In this Figure the structure of the Agent can be observed Node, which if it detects an error, activates, the 5 Phases of Tolerance to Failures, and in each one of them, the interchange of messages.

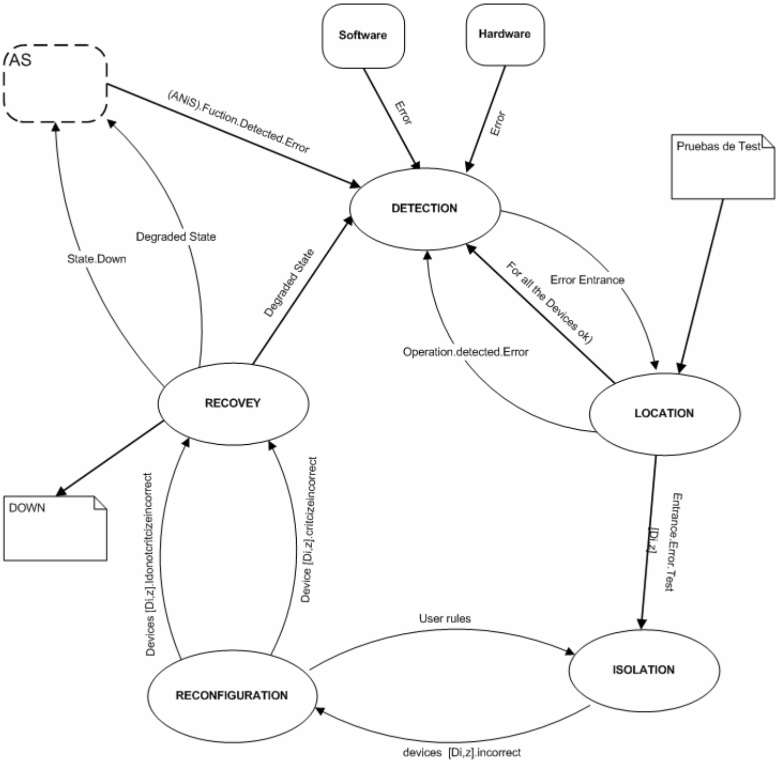


Fig. 3. Several basic communication protocols are supported in KQML

9 Conclusions

The agent counts on a AID, which is "Intelligent Agents as a new paradigm of Distributed Fault tolerant Systems for industrial control" to as Architecture of Reference fipa/Data minimum of an agent is specified in the norms of Fipa (, says: Aid- the agent must have a unique name globally). The agent contains descriptions of transport in the development of his documentation, which fulfills the specifications of fipa (Architecture of Reference fipa/Data minimum of an agent, says: Localizer one or but descriptions of the transport that as well, contains the type of transport by ej. Protocol), but does not specify the protocol that uses like type of transport, this this in phase of analysis. It concerns the communication and cooperation between agents, the document "intelligent Agents as New Paradigm of Distributed Fault tolerant Systems for Industrial Control" says to us that the communication between the agents occurs of ascending or descendent form depending on the type of agent. A a little superficial explanation occurs, without specifying for example that type of language of communication between agents uses, or KQML or the Fipa-acl. We described in this paper our approach for building multi-agents system for achieving fault tolerant control system in industry. The use of the paradigm of intelligent agents has enabled the profile generation of each of the possible failures in an embedded industrial system. In our approach, each of the intelligent agents is able to deal with a failure and stabilize.

References

[1] Stuart Russell and Peter Norvig, Artificial Intelligence to Modern Aproach, Pretence artificial Hall series in intelligence, Chapter Intelligent Agent, pages. 31-52. 1994.

[2] A. Alanis, Of Architectures for Systems Multi-Agentes, (Master Degree thesis in computer sciences), Tijuana Institute of Technology, November, 1996.

[3] Michael J. woodridge, Nicholas R. Jennings. (Eds.), Intelligence Agents, Artificial Lecture Notes in 890 Subseries of Lectures Notes in Computer Science, Amsterdam, Ecai-94 Workshop on Agent Theories, Architectures, and languages, The Netherland, Agust 1994 Proceedings, ed. Springer-Verlag, págs. 2-21.

[4] P.R. Cohen ET al.An Open Agent Architecture, working Notes of the AAAI Spring symp.: Software Agent, AAAI Press, Cambridge, Mass., 1994 págs. 1-8.

[5] Bratko I. Prolog for Programming Artificial Intelligence, Reding, Ma. Addison-Wesley, 1986.

[6] Or Etzioni, N. Lesh, and R. Segal Bulding for Softbots UNIX? (preliminary report). Tech. Report 93-09-01. Univ. of Washington, Seattle, 1993.

[7] Elaine Rich, Kevin Knight, Artificial intelligence, SecondEdition, Ed. Mc Graw-Hill, págs. 476-478.

[8] E. H. Durfee, V. R. Lesser, D. D. Corkill: Trends in cooperative distributed problem solving. IEEE Transactions on Knowledge and Data Engineering KDE-1, 1(March 1989), 63–83.

[9] http://www.fipa.org/specifications/lifecycle.html

[10] http://www.fipa.org/specifications/identifiers.html

[11] http://www.fipa.org/specifications/index.html

[12] M. Yokoo, T. Ishida, K. Kuwabara: Distributed constraint satis-faction for DAI problems. In Proceedings of the 1990 Distributed AI Workshop (Bandara, TX, Oct. 1990).

[13] J. Weizenbaum: ELIZA – a computer program for the study of natural language communication between man and machine. Communications of the Association for Computing Machinery 9, 1965, 36–45.

[14] T. Winograd: A procedural model of language understanding. In Computer Models of Thought and Language, R.Schank and K. Colby, Eds. W.H.Freeman, New York, 1973, pp. 152–186.

[15] FIPA Abstract Architecture Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00001/

[16] FIPA Interaction Protocol Library Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00025/

[17] External Interfaces Working Group ARPA Knowledge Sharing Initiative. Specification of the KQML agent-communication language Working, http://www.cs.umbc.edu/kqml/.

[18] Yannis Labrou and Tim Finin. A semantics approach for KQML { a general purpose communication language for software agents. In Third International Conference on Information and Knowledge Management, November 1994.

[19] Tim Finin, Don McKay, Rich Fritzson, and Robin McEntire. KQML: an information and knowledge exchange protocol. In International Conference on Building and Sharing of Very Large-Scale Knowledge Bases, December 1993.(Ed.), "Knowledge Building and Knowledge Sharing", Ohmsha and IOS Press, 1994.

[20] Jorg Desel, Javier Esparza, "Free Choice Petri Nets", Cambridge University Press, 1995

[21] http://hissa.nist.gov/chissa/SEI_Framework/framework_1.html

[22] Dasgupta P., Narasimhan N., Moser L.E., Melliar-Smith P.M., “MAgNET: Mobile Agents for Networked Electronic Trading”, IEEE Transactions on Knowledge and Data Engineering, 11(4), pp. 509-525, July-Aug. 1999.

[23] N. Jennings, M. Wooldridge: Intelligent agents: Theory and practice. The Knowledge Engineering Review 10, 2 (1995).

Part X

Neural Networks Theory

The Game of Life Using Polynomial Discrete Time Cellular Neural Networks

Eduardo Gomez-Ramirez[1] and Giovanni Egidio Pazienza[2]

[1] LIDETEA, POSGRADO E INVESTIGACION La Salle University 06140, México, D.F.,
egr@ci.ulsa.mx
[2] Department d'Electronica Enginyeria I Arquitectura La Salle Universitat "Ramon Llull",
08022 Barcelona Spain
gpazienza@salle.url.edu

Abstract. There are different approximations to improve the performance and mathematical representation of a cellular neural networks to work with linearly nonseparable data as XOR. But the main goal is to work with problems that only can solved with universal machines such as the game of life. In this paper a new model of Polynomial Cellular Neural Networks that solves the game of life is presented with the learning design to compute the templates.

1 Introduction

The standard uncoupled CNN single-layer structures are not capable of classifying linearly nonseparable data points. The reason is exactly the same as the historic discussion about the perceptrons many years ago published by Marvin Minsky [1]. A more recent description of this idea can be found in [2], using the concepts of linear threshold gates and how it is possible to solve problems with linearly nonseparable data points using quadratic threshold gates. To improve the response of CNN other authors propose using multilayer CNN [3], removing the uniformity constraint on weight values [4] or adding some type of nonlinearities. In [5] a trapezoidal activation function is applied to classify linearly non-separable data points. Some examples are presented realizing Boolean operations (including eXclusive OR) by using only a single-layer CNN.

In this paper a new nonlinear term is added to the traditional discrete model with the result of a Polynomial Discrete time Cellular Neural Network (PDTCNN). This new model improves the capabilities of representation of the network with only one layer. To show the advantage of the algorithm we select a traditional example to explore the capabilities of mathematical representation: *The Game of Life*. This algorithm is traditional used in the area of cellular automata to show that some type of automata can realize universal computing. There is a previous work of Chua et al. to solve this problem but using a multilayer CNN.

The structure of the paper is the following: Section II explains the model of Polynomial Cellular Neural Network. Next section explains the learning methodology using Genetic Algorithm. Section IV shows some experiments and conclusions of the work.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 719–726, 2007.
springerlink.com

2 Polynomial Cellular Neural Networks

2.1 Cellular Neural Networks

Cellular Neural Networks [6] are a particular kind of neural networks with local connections only. In the most common model, cells are arranged in a square grid, and the state of each one depends directly on the input and the output of its 8 nearest cells. In this paper we deal with Discrete-Time CNNs (DTCNNs) [7], a CNN model in which only binary output values are allowed and the system is clocked. Its dynamic is described by the following equations

$$x_{ij}(n) = \sum_{C(k,l)\in Nr(i,j)} A(i,j;k,l)\, y_{kl}(n) + \sum_{C(k,l)\in Nr(i,j)} B(i,j;k,l)\, u_{kl}(n) + i \tag{1}$$

$$y_{ij}(n+1) = f(x_{ij}(n)) = \begin{cases} 1 \; if\; x_{ij}(n) \geq 0 \\ -1 if\; x_{ij}(n) < 0 \end{cases} \tag{2}$$

where u_{ij}, x_{ij} and y_{ij} are the input, the state and the output of the cell in position *(i,j)*, respectively; *Nr(i,j)* is the set of indexes corresponding to the cell *(i,j)* and its neighbors; finally, the matrices *A* and *B* contain the weights of the network and the value *i* is a bias. Usually, the weights of a CNN depend on the difference between cell indexes rather than on their absolute values, therefore the network is space-invariant. In this case the operation performed is defined by the set of weights *{A,B,i}* that is unique for the whole network and is called 'cloning template' (see [8]). Note that throughout this paper capital letters indicate matrices and lower case letters indicate scalar values.

The Einstein notation - according to whom when an index variable appears twice in a single term, it implies that we are summing over all of its possible values - allows to write the Eq. (5) in a compact way as follows:

$$x^c(n) = a^c_d y^d(n) + b^c_d u^d(n) + i \tag{3}$$

2.2 Extension to the Polynomial Model

Despite their versatility, space-invariant Cellular Neural Networks are not able to solve non-linear separable problems, unless a multilayer structure is used [9]. This difficulty can be overcome by using a polynomial CNN (PCNN) [10, 11,12], whose discrete-time state equation is:

$$x^c(n) = a^c_d y^d(n) + b^c_d u^d(n) + i + g(u^d, y^d) \tag{4}$$

where a polynomial term $g(u^d, y^d)$ is added to the standard DTCNN model. The output is calculated as before through the Eq. (4). In [13] it is proved that a polynomial CNN allows to carry out the XOR operation - the most elemental non-linear separable task -, whereas other authors have applied this model to real-life cases, like epilepsy seizure prediction [14]. In the simplest case $g(u^d, y^d)$ is a second degree polynomial, whose general form is:

$$\begin{aligned} g(u^d, y^d) &= \sum_{i=0}^{2} (q_i)_d^c \left(u^d\right)^i \cdot (p_i)_d^c \left(y^d\right)^{2-i} \\ &= (q_0)_d^c \left(1^d\right) \cdot (p_0)_d^c \left(y^d\right)^2 \\ &+ (q_1)_d^c \left(u^d\right) \cdot (p_1)_d^c \left(y^d\right) \\ &+ (q_2)_d^c \left(u^d\right)^2 \cdot (p_2)_d^c \left(1^d\right) \end{aligned} \tag{5}$$

where $(P_0,...,P_2,Q_0,...,Q_2)$ are matrices of dimensions $\in Nr(i,j)$.

3 The Game of Life and Its Rules

The Game of Life (GoL) [15] is a cellular automaton consisting in a two-dimensional grid of binary cells, which at any fixed time are said to be either alive (black cells) or dead (white cells). Each cell interacts with its 8 neighbor cells, and its state changes according to the following rules:

- *Birth*: a cell that is dead at time t becomes alive at time t + 1 only if exactly 3 of its neighbors were alive at time t;
- *Survival*: a cell that was living at time t will remain alive at t + 1 if and only if it had exactly 2 or 3 alive neighbors at time t.

It is noteworthy to emphasize that the evolution depends only on the initial state of the grid, therefore the GoL is not a game in the conventional sense as there are no players, and no winning or losing.

The importance of the GoL comes from the fact that it is proved to be a Turing machine or, in other words, it is capable of universal computation; moreover, a full computer can be built using GoL [16].

4 Remarks on the Relation Between GoL and CNN

A fundamental aspect of CNNs is how to find a cloning template - that is, the set of weights - to perform a given operation. This issue, commonly called ‘CNN learning’, is crucial, and many efforts have been made to deal with it. It is possible to design explicitly a template only in simple cases (e.g. [17,18]) thanks to the formulation of local rules. However, often it is necessary to find the template through a supervised learning algorithm. What is more, gradient-based algorithms, successfully employed with a number of neural networks, do not work in general with CNNs [19]. Therefore, the only way to solve the problem is to apply global learning procedures, like genetic algorithms [20] or simulated annealing [21,22], in which the learning problem is reformulated as a general optimization problem. The cost function is usually the difference between the desired output and the one obtained using a certain configuration for the weights. Now, the point is to establish what parameters of the network must be learnt.

First, as detailed in previous section, the rules of the GoL are applied on a 3×3 cells square, and they focus only on the central cell and the number of black pixels in the neighborhood regardless their position. Consequently, a cloning template contains only matrices with central symmetry. Therefore, just two values of each matrix must be learnt: the central element (subindex *c*) and the peripheral element (subindex *p*). Second, we can state that the last term of Eq. (5) is constant because the input is binary; thus, it can be included in the bias. Also the value $(p_1)^c{}_d(1^d)$ belonging to the first term of Eq. (5) is constant, then it can be ignored during the learning.

To sum up, the polynomial CNN model for the GoL can be written as:

$$x^c(n) = a_d^c y^d(n) + b_d^c u^d(n) + p_d^c y^d \cdot q_d^c u^d + r_d^c (y^d)^2 + i \qquad (6)$$

where

$$A = \begin{bmatrix} a_p & a_p & a_p \\ a_p & a_c & a_p \\ a_p & a_p & a_p \end{bmatrix} \quad B = \begin{bmatrix} b_p & b_p & b_p \\ b_p & b_c & b_p \\ b_p & b_p & b_p \end{bmatrix}$$

$$P = \begin{bmatrix} p_p & p_p & p_p \\ p_p & p_c & p_p \\ p_p & p_p & p_p \end{bmatrix} \quad Q = \begin{bmatrix} q_p & q_p & q_p \\ q_p & q_c & q_p \\ q_p & q_p & q_p \end{bmatrix} \quad R = \begin{bmatrix} r_p & r_p & r_p \\ r_p & r_c & r_p \\ r_p & r_p & r_p \end{bmatrix}$$

In our work we take into consideration a space-invariant linear DTCNN, whose model, described by the Eq. (3), comprises 19 free parameters (9 for the A matrix, 9 for the B matrix and the bias). When the second order polynomial of Eq. (5) is added, there are 45 more terms to consider (9 for each of the five matrices). This results in a huge search space, but making some preliminary consideration about the problem, the search space can be easily shrunk.

In conclusion, thanks to the remarks made in this section, we realized that only 11 parameters $\{a_c, a_p, b_c, b_p, p_c, p_p, q_c, q_p, r_c, r_p, i\}$ must be learnt, which is a great improvement with respect to the initial hypothesis of 46 free parameters.

5 Genetic Algorithm

This algorithm was first proposed by John Holland in 1975 [23]. GA is a knowledge model inspired on some of the evolution mechanisms that are observed in nature such as: inheritance, mutation, selection and crossover [24][25]:

In this work a modified version of the work presented in [26] is used to train the PDTCNN. Next Sections describe these processes in detail.

5.1 Crossover

The crossover operator is characterized by the combination of the individual's genetic material that is selected in function of the good performance (objective function). To explain the multipoint crossover for each fixed number $g=1,..,n_g$, where n_g is the number of total generations, let us introduce the matrix F_g which is the set of parents for a given population. This is a matrix of dimension $F_g : n_p x N_i$, where n_p is the number of parents at the generation g and N_i is the size of every array (chromosomes). In our case this depends of the type of radius $r \in N_r(c)$ of the CNN selected. Let $C_1(F_g, n_t)$ be the crossover operator which can be defined as the combination between the information of the parents set considering the number of intervals n_t of each individual and the number of sons n_s:

$$n_{s_1} = n_p^{\ n_t} \tag{7}$$

Then $C(F_g, n_t): n_p x\, N_i \to n_s x\, N_i$. To show how the crossover operator can be applied, the following example is introduced. Let F_g has $n_p=2$ and $N_i=3$. This means that the array (the information of one father) can be divided in 3 sections and every section is determined with a_i, b_i, and $c_{\rm i}$ respectively. The number of intervals it is equivalent to the number of multipoint crossover plus one. In this case very simple case the number of intervals is 3 and the number of multipoint crossover is equal to 2.. Note that with this operator, the parents F_g of the population g are included in the result of the crossover (←). Remember, that every father corresponds to the strategy of every player.
Example:

$$F_g = \begin{bmatrix} p^1 \\ p^2 \end{bmatrix}, \begin{matrix} p^1 = \begin{bmatrix} a^1 & b^1 & c^1 \end{bmatrix} \\ p^2 = \begin{bmatrix} a^2 & b^2 & c^2 \end{bmatrix} \end{matrix}, \Rightarrow C_1\left(F_g\right) = \begin{bmatrix} a^1 & b^1 & c^1 \leftarrow \\ a^1 & b^1 & c^2 \\ a^1 & b^2 & c^1 \\ a^1 & b^2 & c^2 \\ a^2 & b^1 & c^1 \\ a^2 & b^1 & c^2 \\ a^2 & b^2 & c^1 \\ a^2 & b^2 & c^2 \leftarrow \end{bmatrix}$$

Depending of the size of templates it is possible that the number of crossover combinations is very big In this case, it is necessary to select with some fixed probability the proportion of the population that is going to be selected.

5.2 Mutation

The mutation operator just changes some elements of every template that were selected in a random way from a fixed probability factor P_m; in other words, we just vary the components of some *genes*. This operator is extremely important, because assures the maintenance of the diversity inside the population, which is basic for the evolution. This operator $M : n_s x N_i \rightarrow n_s x N_i$ changes with probability P_m a specific population in the following way:

$$M\left(F_{ij}, P_m\right) = \begin{cases} \overline{F}_{ij} & r(\omega) \leq P_m \\ F_{ij} & r(\omega) > P_m \end{cases} \tag{8}$$

$\overline{F}_{ij} \in [-e, e]$ $e \in \Re$ represents the element of the mutated strategy. Considering that the images used normally belong to discrete intervals between 0 and 255, the mutation is generated using random numbers with fixed increments that can be selected in the program. With this option it is possible to reduce the searching space of the templates and, off course, the computing time to find the optimal solution.

5.3 Selection Process

The Selection Process S_g computes the objective function O_g that represents a specific criterion to maximize or minimize, and it selects the best n_p individuals of A_g as:

$$S_g\left(A_g, n_p\right) = \max{}^{n_p} O_g\left(A_g\right) \tag{9}$$

Then, the parents of the next generation can be computed as:

$$F_{g+1} = S_g\left(A_g, n_p\right) \tag{10}$$

The objective function can be the square error or the hamming distance in the case of binary images. In this step a fixed big number is added to the objective function if the template is not stable to eliminate it from the population.

5.4 Add Random Parents

To avoid local minima a new scheme is introduced and it is called add random parents. If the best individual of one generation is the same than the previous one, a new random individual is included like a parent of the next generation. This step increases the population due to the application of crossover with the new random parent. This step is tantamount to have a very big mutation probability and to search in new points of the solution space.

6 Results

First of all, a training set was generated using a simple 10x10 matrix with random values as an input and the output was computed with the rules of the Game of Life. The parameters used for the training set are:

Table 1. Training Parameters

PARAMETER	VALUE
# of fathers	4
# of maximum random parents	3
Percentage mutation	0.15
Initial population	20000
Increment	1

Table 2. Results of the simulations

a_c	a_p	b_c	b_p	p_c	p_p	q_c	q_p	r_c	r_p	i
0	0	0	-2	-6	2	0	3	0	-2	0
0	0	0	-3	3	-1	-1	-5	-1	-2	0
0	0	-2	-1	-6	2	0	2	0	-2	0

Table 2 shows the simplest solution obtained with the algorithm. The error was computed using a Hamming Distance plus the sum of the absolute of all the terms divided by 100.

7 Conclusions

In this paper different group of templates of a Polynomial Cellular Neural Networks that solve the game of life are presented. This means that the PCNN has the same power of a Universal Turing Machine. With this work there are new possibilities to combine the original idea of the cellular neural networks with the theory of cellular automata.

References

[1] M. Minsky & S. Paper, (1969) Perceptrons: An introduction to Computational Geometry. MIT Press, Cambridge, Mass,.

[2] M.H. Hassoun. (1995) Fundamentals of Artificial Neural Networks, A Bradford Book, MIT Press,

[3] Z. Yang; Nishio, Y.; Ushida, A.; (2002) Templates and algorithms for two-layer cellular neural networks Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on Volume 2, 12-17 May 2002 Page(s):1946 - 1951

[4] M. Balsi, (1992) "Generalized CNN: Potentials of a CNN with Non-Uniform Weights", Proceedings of IEEE Second International Workshop on Cellular Neural Networks and their Applications (CNNA-92), Munich, Germany, Oct. 14-16, 1992, 129-134.

[5] E. Bilgili1, I. C. Göknar and O. N. Ucan (2005), Cellular neural network with trapezoidal activation function. Int. J. Circ. Theor. Appl.; 33:393–417

[6] Chua, L., Yang, L.: (1988) Cellular neural networks: theory. IEEE Trans. Circuits Syst. **35** 1257–1272

[7] Harrer, H., Nossek, J.: (1992) Discrete-time cellular neural networks. International Journal of Circuit Theory and Applications **20** 453–467

[8] Roska, T., Kék, T., Nemes, L., Zarándy, Á., Szolgay, P.: CSL-CNN software library,version 7.3 (templates and algorithms) (1999)

[9] Yang, Z., Nishio, Y., Ushida, A. (2002): Templates and algorithms for two-layer cellular neural networks neural networks. In: Proc. of IJCNN'02. Volume 2. 1946–1951

[10] Schonmeyer, R., Feiden, D., Tetzlaff, R. (2002): Multi-template training for image processing with cellular neural networks. In: Proc. of 2002 7th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA'02), Frankfurt, Germany

[11] Laiho, M., Paasio, A., Kahanen, A., Halonen, K. (2002): Realization of couplings in a polynomial type mixed-mode cnns. In: Proc. of 2002 7th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA'02), Frankfurt, Germany

[12] Corinto, F.: (2005) Cellular Nonlinear Networks: Analysis, Design and Applications. PhD thesis, Politecnico di Torino, Turin

[13] Gómez-Ramírez, E., Pazienza, G.E., Vilasís-Cardona, X. (2006): Polynomial discrete time cellular neural networks to solve the XOR problem. In: Proc. 10th International Workshop on Cellular Neural Networks and their Applications (CNNA'06), Istanbul, Turkey

[14] Niederhofer, C., Tetzlaff, R. (2005): Recent results on the prediction of EEG signals in epilepsy by discrete-time cellular neural networks DTCNN. In: Proc. IEEE International Symposium on Circuits and Systems (ISCAS'05). 5218–5221

[15] Berlekamp, E., Conway, J.H., Guy, R.K.: (1982) Winning ways for your mathematical plays. Academic Press, New York

[16] Rendell, P.: (2006) A Turing machine in Conway's game life. Available: www.cs.ualberta.ca/~bulitko/f02/papers/tmwords.pdf

[17] Matsumoto, T., Chua, L., Furukawa, R. (1990): CNN cloning template: connected component detector. IEEE Trans. Circuits Syst. **37**(5) 633–635

[18] Zarándy, Á. (1992): The art of CNN template design. International Journal of Circuit Theory and Applications 27 5–23

[19] Magnussen, H. (1994): Discrete-time cellular neural networks: theory and global learning algorithms. PhD thesis, Techinical university of Munich, Munich

[20] Kozek, T., Roska, T., Chua, L. (1993: Genetic algorithm for CNN template learning. IEEE Trans. Circuits Syst. **40**(6)) 392–402

[21] Magnussen, H., Nossek, J. (1994): Global learning algorithms for discrete-time cellular neural networks. In: Proc. third IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA'94), Rome, Italy 165–170

[22] Kellner, A., Magnussen, H., Nossek, J. (1994): Texture classification, texture segmentation and text segmentation with discrete-time cellular neural networks. In: Proc. third IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA'94), Rome, Italy 243–248

[23] J. H. Holland (1992), Adaptation in Natural and Artificial Systems, MIT Press /Bradford Books Edition,

[24] J. T. Alander, An Indexed Bibliography of Genetic Algorithms: Years 1957--1993, 1994, Art of CAD ltd

[25] Bedner (1997), Genetic Algorithms and Genetic Programming at Stanford, Stanford Bookstore,

[26] E. Gómez-Ramírez & F. Mazzanti (2002), Cellular Neural Networks Learning using Genetic Algorithm, 7mo. Congreso Iberoamericano de Reconocimiento de Patrones, CIARP 2002, Noviembre 19-22, CIC-IPN, Cd. de México, México.

Reinforcement Learning in Continuous Systems: Wavelet Networks Approach

I.S. Razo-Zapata, J. Waissman-Vilanova, and L.E. Ramos-Velasco

Research Center in Information Technologies and Systems, Autonomous University of the State of Hidalgo, Pachuca, Hidalgo, Mexico
ri096373@uaeh.reduaeh.mx, {julio,lramos}@uaeh.edu.mx

Abstract. In this paper we integrate the self-adjustment environment advantages of reinforcement learning into the adaptive wavelet network controller. The novel approach is called adaptive wavelet reinforcement learning control, which uses wavelet to approximate a continuous Q-function, in order to obtain a optimal control policy. The simulations of applying the proposed method to an inverted pendulum and Pendubot system are performed to demonstrate the properties of the adaptive wavelet network controller.

Keywords: Reinforcement learning, adaptive wavelet networks, continuous systems, underactuated systems.

1 Introduction

Reinforcement learning (RL) is learning to perform sequential decision tasks without explicit instructions, only optimizing a criterion about how the task is perform. So the learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. This method, is goal-directed, and seems better adapted to the solution of control problems[1], which ones about searching a final goal, and the problem is to find a policy that reach this goal [2]. The basic RL algorithms use a look-up table scheme in order to represent the value function $Q(s,a)$. Unfortunately this representation is limited when working with continuous spaces like physical systems. Several approaches can be applied to deal with this problems, such function approximation techniques. And neural networks offers an interesting perspective due to their ability to approximate nonlinear functions. In recent years, wavelet have attracted much attention in many scientific and engineering research areas. Wavelets possess two features that make them especially valuable for data analysis: they reveal local properties of the data and they allow multiscale analysis. Their locally is useful for applications that requires online response to changes, such a controlling process. Wavelets and neural networks have been combined [3,4], to form a class of networks, wavelet networks which are capable of handling moderately high-dimensional problems [5]. Inspired by the theory of multi-resolution analysis of wavelet transform and suitable adaptive control laws, an adaptive wavelet network is proposed for approximating action-value functions. In this paper, we propose an adaptive wavelet reinforcement learning control (AWRLC) whose design is based on the promising function approximation capability of wavelet networks. The goal of the

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 727–736, 2007.
springerlink.com

paper is to propose a control scheme based on RL algorithms and an AWRLC to control underactuated systems. In this work the inverted cart-pole system and the Pendubot were used like examples in order to evaluate the advantages and disadvantages of AWRLC methods for control of underactuated systems. The work is organized as follows. Section 2 presents the reinforcement learning approach. In Section 3 is summariezed the background about wavelets networks while Section 4 shows the control scheme which is implemented in the systems. Section 5 shows the results obtained by numerical simulation. Finally, in Section 6 conclusions from results and future works are presented.

2 Reinforcement Learning

Q-Learning is a reinforcement learning method where the learner builds incrementally a Q-function which attempts to estimate the discounted future rewards for taking actions given states. So, in a common control task maximize the total return R expressed in (1) is the main objective [2]

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{1}$$

And in this way, the output of the Q-function for state s and action a is denoted by $Q(s,a)$. When action a has been chosen and applied, the system is moved to a new state, s_{t+1}, and a reinforcement signal, r, is received, $Q(s,a)$ is updated by [2]:

$$Q(s_{t+1}, a_{t+1}) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{2}$$

where $0 \leq \alpha \leq 1$ is the *learning rate*, and $0 \leq \gamma \leq 1$ is called the *discount*, this parameter is used to decrease r_t in the total return (1).

3 Wavelet Networks

Wavelets have been found to be useful in many applications such a physics, signal processing, statistics and more recently in control. From a control and estimation viewpoint, a key property of wavelets is that they posses local characteristics in both space and spatial frequency. This provides a framework for resolving data at varying levels of detail, which is known as multi-resolution analysis. Wavelet network is a type of building block for approximation of the unknown function by the concept of the multi-resolution approximation. The building block is formed by shifting and dilating the basis function called "mother wavelet" ψ, (the modified version is its "daughter wavelet") and a "father wavelet" ϕ. Most commonly, wavelet bases are derived using shift-invariance and dyadic dilation. In this way we use the dyadic series expansion

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), \quad j,k \in \mathbb{Z} \tag{3}$$

which is integral power of 2 for frequency partitioning. The daughter wavelet (3) is obtained from a mother wavelet function ψ by a binary dilation (i.e. dilation by 2^j) and a dyadic translation (of $k/2^j$). The principle of wavelet construction is as follows:

- $\forall k \in \mathbb{Z}, \psi(x-k)$ are mutually orthogonal,
- the mother wavelet ψ is a scaling function and the daughter wavelets of $2^{j/2}\psi(2^j x - k)$ constitutes an orthonormal basis of $L^2(\mathbb{R})$,
- ψ is called a *orthonormal wavelet* if the family of $2^{j/2}\psi(2^j x - k)$ constitutes an orthonormal basis of $L^2(\mathbb{R})$, that is $<\psi_{ij}, \psi_{kl}> = \delta_{ij}\delta_{kl}, \quad \forall i,j,k,l \in \mathbb{Z}$

It follows that any function $v(x) \in L^2(\mathbb{R})$ can be expressed as a wavelet series expansion [7]

$$v(x) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} c_{ij}\psi_{ij}(x) \tag{4}$$

All of the one dimensional results of wavelets theory described above can be extend to multiple dimensions, i.e. to $L^2(\mathbb{R})$, where $n > 1$ is an integer [8]. One scheme is to generate separable scaling functions and wavelets by tensor products of one dimensional “father wavelet” ϕ and “mother wavelet” ψ, i.e. from

$$\phi(X) = \prod_{j=1}^{n} \phi(x_j), \quad \psi(X) = \prod_{j=1}^{n} \psi(x_j) \tag{5}$$

Since a wavelet network can approximate unknown functions to any desired accuracy with an appropriate structure, in this paper the wavelet network will be used to approximate the unknown function $Q(s,a)$.

4 Control Scheme

In order to deal with continuous states, the approximation of Q-function is implemented with an artificial neural network (ANN), which one is an universal function approximator [5]. Recently ANN's are used with *wavelets* in system identification, and specifically in function approximation, in this way the implementation of wavelet networks in reinforcement learning is justified. The general structure of the wavelet network (**W**) applied in the approximation of the Q-function is shown in Fig. 1.(a). In this scheme the input parameters are the states configuration

$X = [x_1, x_2, ..., x_n]^T$, and in layer 2 there are wavelet basis in order to support each one of the states X . This kind of wavelet basis has support of dimensions n, this support is obtained with the tensor product (6). The wavelet network structure could be represent like a MIMO system, with n input variables and m output variables. Finally in Fig. 1.(a) the last layer has m neurons, which ones are the actions applied by the RL agent.

4.1 Building the Wavelet Network

The process of building the architecture of the wavelet network is performed on-line with the exploration through new states, due in part to the absence of data generated in past operations (training data). This growing of wavelet basis provide support to new states in order to approximate the Q-function with better accuracy.

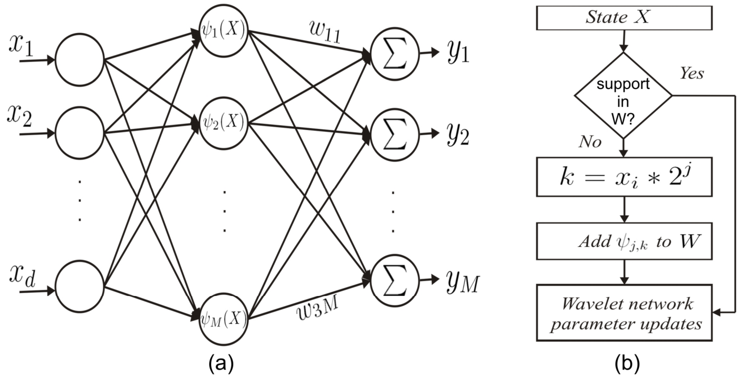


Fig. 1. (a) Wavelet Networks like a MIMO system with n inputs $X = [x_1, x_2, ...x_n]^T$ and m outputs $Y = [y_1, y_2, ..., y_m]^T$, and in the second layer activation functions are wavelets; (b) Building the wavelet network

New wavelet basis are generated with series expansion techniques, in this case was applied (3) with n-dimensions. The mother wavelet implemented in this scheme is Mexican Hat defined as follows

$$\psi(x) = \frac{2}{\sqrt{3\sqrt{\pi}}}(1 - x^2)e^{\frac{-x^2}{2}} \tag{6}$$

The translation parameter in each one of the dimensions is determined by $k = x_i * 2^j$ where j is the scale of the wavelet basis and x_i is an element without support in the structure of **W**, when the learning process begins, the first neuron with support in **W** is created over the initial coordinates $x_1 = 0, x_2 = 0, ..., x_n = 0$. In this way is possible initialize the learning with **W** empty, **W** grows accord to the explored regions, the growing is controlled by the election of a threshold ξ, which one determines the minimum value in order to consider if a given state has support in the structure of the network. The diagram shown in Fig. 1.(b) presents the algorithm for the construction of the wavelet network **W**.

4.2 Training the Wavelet Network

The training process in **W** consists in update the weights of layer 3. This training is performed on line according to the interaction in the environment. The updating rule is a combination between reinforcement learning and the gradient descent method. The reinforcement learning method implemented is Q- Learning due to its capacity of learning over the best action independently of the action selected. The updating rule applied in W is given by:

$$w_{i,j}(t+1) = w_{i,j}(t) + \alpha\left[r_{t+1} + \gamma \max_{yM} y_j(t+1) - y_j(t)\right]\left|\psi(X)\right| \tag{7}$$

where $j = 1,...,m$, α is the learning rate, r_{t+1} is the reward at the time $t+1$, and γ is the discount with the same function that in Eq. 2. Because the structure shown in Fig. 1.(a) is a MIMO system, at the time t there is a vector $Y = [y_1, y_2,..., y_m]^T$ with m outputs. So $y_i(t)$ is the output value of the neuron i, where $i \in 1,2,...,m$. The value of i represents the selected action in the exploration process, treated in the next section. In the training applied to neuronal networks, target values are required. And, in this scheme the target values is the maximum output given by **W** at the time $t+1$. In this way, the approximation of $Q(s,a)$ is achieved on-line without training data. In each iteration several neurons could be activated, but only the weights of neurons with value of activation bigger than the threshold $0 < \xi < 1$ are modified.

4.3 Action Selection

During the learning process, the network **W** allows to make elections between the set of actions $A(s)$. This selection can be applied with some kind of exploration like *Softmax* or ε*-greedy* [2]. The selection of actions is performed with the output values of **W**, $(y_1, y_2,..., y_m)$. Each output represents the approximated value for the function $Q(s,a)$. And in a *greedy* exploration, the neuron with maximum value always is taken as control action.

5 Simulation Results

The control scheme presented in Section 4 was applied to two underactuated systems (cart-pole and *Pendubot*) in order to control both in one equilibrium point. For the cart-pole the equilibrium point is the vertical position of the pendulum and in the Pendubot the equilibrium position is the called UP-UP configuration (the first and the second link in vertical position). So two control tasks are required. Operation parameters are summarized as follows. The number of inputs *n* are 4 $X = [x_1, x_2, x_3, x_4]^T$; *m* outputs are 3 $Y = [y_1, y_2, y_3]^T$; the set of actions is *A(s)* with [+1.0-1]; α and γ with 0.2 and 0.9 respectively; the mother wavelet applied was Mexican Hat with scale *j* = 3; finally Exploration ε -greedy and Threshold ξ with values of 0.1 and 0.2 respectively.

5.1 Cart-Pole System

In order to keep the system in the equilibrium point the rewards were assigned in continuous form, according to $p, \dot{p}, \theta$ and $\dot{\theta}$. The desired value for p is around zero, but during the learning the agent explores several configurations. However p is limited to $|3|$. When $|p|$ is greater than 3, the reward is -10, this is the unique constraint about $|p|$. With respect to θ, the operation is limited to $15°$, in this way when $|\theta|$ is greater than $15°$ the reward is -10 and system is reset to a new random state. Besides in order to improve the control task when $|\theta|$ is greater than $5°$, the reward is -5. In other position the reward is -1 except when $\theta \simeq 0$, $\dot{p} \le \beta$ and $\dot{\theta} \le \beta$, in this case reward is +1, and $\beta = 0.01$. Rewards were assigned in this way because the RL algorithms attempts to maximize a total return for a long run. So, they try to reach every time states with better values.

The simulation was applied with a time step of $0.01s$ during 2000 episodes, Fig. 2.(a) shows the average of steps over the time in the learning process for each episode with two scales, $j = 2$ and $j = 3$. This graph shows that the scale $j = 3$ give a better average of steps by episode than $j = 2$. For $j = 3$ in the firsts explorations the number of steps is relatively small. However, in the last explorations the average of steps in each episode is growing over the time, in this way the RL agent learns to stay in states with better rewards during more time. The main function of wavelet network is to approach the action value function $Q(s, a)$, with high resolution.

The Fig. 3, represents the approach of Q-function with $j = 2$ and $j = 3$ over a range ($-0.1 \le p \le 0.1$ and $-0.1 \le \dot{p} \le 0.1$). In this set of figures is possible to observe that the scale of 3 has a good resolution in order to control the system. In addition, it has provided a smooth approach of the function than scale $j = 2$. Besides, in both figures are observed the regions with higher values. These regions are the positions of the system which ones allow to reach the goal. The growing of the wavelet network **W** is presented in Fig. 2.(b), the creation of neurons grows over the time accord to the environment interaction, due this number of neurons a pruning process was proposed in order to reduce the computational cost.

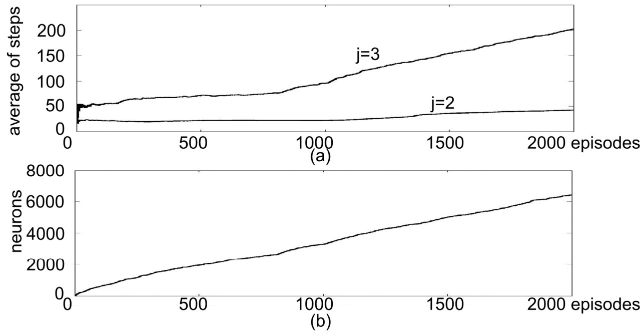


Fig. 2. (a) Steps in each episode; (b) Neurons vs time steps

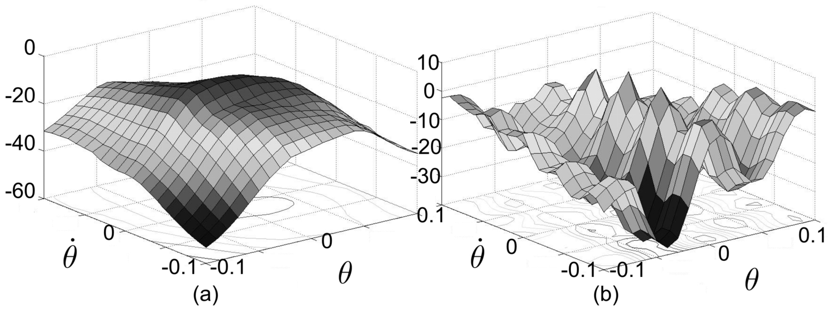


Fig. 3. Q-Function with $[p, \dot{p}] = [0,0]$ (a) $j = 2$ (b) = $j = 3$

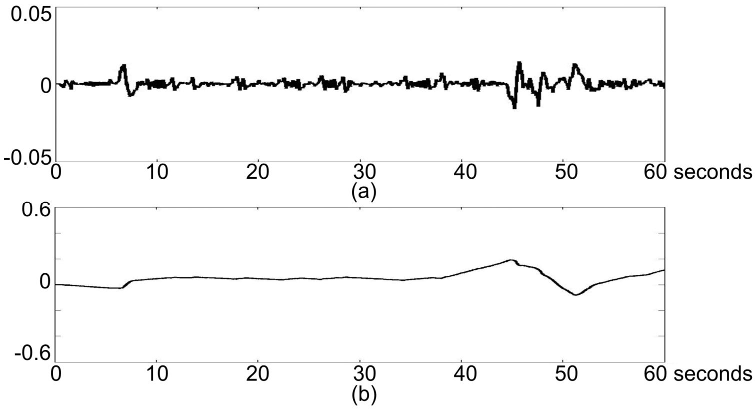


Fig. 4. (a) Pendulum's angle with disturbances, (b) Car position with disturbances

The pruning process consist in to remove neurons without influence in the control task, so 10 operation process were performed over 60 seconds, the neurons with importance in the solution were identified. Originally the network **W** has 6497 neurons, but with the pruning process the number was reduced to 630. The reduced AWRLC was applied to the system allowing to control the pendulum in the desirable position, although to disturbances introduced to the torque in the car. Fig. 4.(a) shows the behavior of the pendulum in 60 seconds of operation. During this time were applied every 5 seconds random disturbances of +3 and -3 which ones replace the correct value estimated for the wavelet network. In this plot the value of θ is around to the desirable value, and the pendulum is in vertical position in spite of the disturbances. In the same way Fig. 4.(b) presents the behavior of the car's position during 60 seconds, and is possible to observe that the position is stable around an ideal value.

5.2 Pendubot System

Like in cart-pole system in this problem the rewards were assigned in continuous form, according to $\theta_1, \dot{\theta}_1, \theta_2$ and $\dot{\theta}_2$. The desired value for θ_1 and θ_2 is around π (UP-UP Configuration). So rewards were given accord to the angular position of both links, this assignation was realized as follows:

- -3 when $\theta_1 \simeq \pi, \theta_2 \neq \pi, \dot{\theta}_1 > 0$ and $\dot{\theta}_2 > 0$
- -2 when $\theta_1 \simeq \pi, \theta_2 \simeq \pi, \dot{\theta}_1 > 0$ and $\dot{\theta}_2 > 0$
- -1 when $\theta_1 \simeq \pi, \theta_2 \simeq \pi, \dot{\theta}_1 \simeq 0$ and $\dot{\theta}_2 > 0$
- +1 when $\theta_1 \simeq \pi, \theta_2 \simeq \pi, \dot{\theta}_1 \simeq 0$ and $\dot{\theta}_2 \simeq 0$

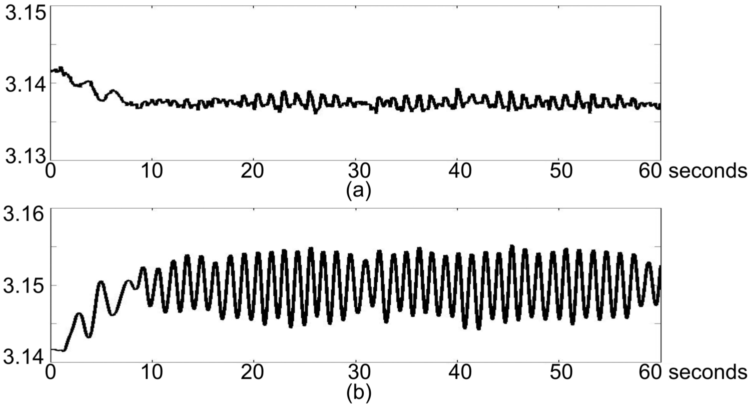


Fig. 5. (a) Angular position of the first link with disturbances, (b) Angular position of the second link with disturbances

The simulation was applied with a time step of $0.01s$ during 2000 episodes, with $j = 3$. A pruning process was applied in order to reduce the number of neurons, 10 operation process were performed and the most important neurons were identified. Originally the network **W** has 2463 neurons, but with the pruning process the number was reduced to 131. The reduced AWRLC was applied to the system allowing to control the system in the desirable position, although to disturbances introduced to the torque of the first link. Fig. 5.(a) shows the behavior of the first link in 60 seconds of operation. Random disturbances were applied every 5 seconds of +2 and -2. In this plot the value of θ_1 is around to the desirable value in spite of the disturbances. In the same way Fig. 5.(b) presents the behavior of the second link during 60 seconds. It is possible to observe that the position is stable around an ideal value.

6 Conclusions

This paper presents an alternative approach to control underactuated systems using a novel approach which is called adaptive wavelet reinforcement learning control (AWRLC). The systems studied are the *cart-pole* and the *Pendubot*.

The control of this systems is specially difficult since are underactuated mechanisms (two degrees of freedom and only one input). Control scheme is based on learning methods, implementing one of the most popular RL algorithms: Q-Learning with adaptive wavelets networks. This approach uses a wavelet networks to approximate a Q-function, where the function gives the optimal control policy. Using only 3 actions in the control scheme is an approach toward a set of basic actions of control. The simulation results show that this controller provides a good performance when keeping the systems in the unstable vertical positions. Results indicate that the AWRLC is a potentially attractive alternative for underactuated systems. The algorithm for build wavelets networks in Fig. 1.(b) represents an advantage working with physical systems which ones has unknown limits of operation, because this method of generating neurons create support in regions accord to the explorations. Multi-resolution is other attractive property handled by this kind of wavelet networks. The scales of resolution allow to approximate with good accuracy unknown functions, and in this case coarse approximations doesn't produces optimal control policies. Besides, a pruning process is proposed in this work, in order to reduce the computational complexity of the AWRLC scheme.

References

1. K. S. Fu, (1970) Learning Control Systems-Review and Outlook. IEEE Transactions on Automatic Control, vol. AC-15, 210-221.
2. R. S. Sutton, A. G. Barto (1998) Reinforcement Learning An Introduction, The MIT Press.
3. Q. Zhang, and A. Beneveniste (1992) Wavelet networks. IEEE Trans. Neural Networks, vol. 3, 889-898.
4. Q. Zhang (1997) Using wavelet networks in nonparametric estimation. IEEE Trans. Neural Networks, vol. 8, 227-236.
5. M. T. Hagan, H. B. Demuth and M. Beale (1996) Neural Network Design, PWS Publishing Company.
6. I. S. Razo-Zapata, L.E. Ramos-Velasco and J. Waissman-Vilanova (2006) Reinforcement Learning of Underactuated Systems. In International Symposium on Robotics and Automation, San Miguel Regla, Hidalgo, México, pp 420-424.

7. S. Mallat (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions Pattern Recognition and Machine Intelligence, vol. 11, 674-693.
8. Jian-xin Xu and Ying Tan (2001) Nonlinear Adaptive Wavelet Control Using Constructive Wavelet Networks. American Control Conference, Arlington, VA, 624-629.
9. W. Sun and Y. Wang and J. Mao (2002) Using Wavelet Network for Identifying the Model of Robot Manipulator, World Congress on Intelligent Control and Automation, pp 1634-1638.

Support Vector Machine-Based ECG Compression

S.M. Szilágyi[1], L. Szilágyi[1,2], and Z. Benyó[2]

[1] Sapientia – Hungarian Science University of Transylvania, Faculty of Technical and Human Sciences, Târgu Mureş, Romania
szs@ms.sapientia.ro

[2] Budapest University of Technology and Economics, Dept. of Control Engineering and Information Technology, Budapest, Hungary

Abstract. This paper presents an adaptive, support vector machine-based ECG signal processing and compression method. After a conventional pre-filtering step, the characteristic waves (QRS, P, T) from the ECG signal are localized. The following step contains a regressive model for waveform description in terms of model parameters. The gained information allows an iterative filtering in permanent concordance with the aimed processing manner. The structure of the algorithm allows real-time adaptation to the heart's state. Using these methods for one channel of the MIT-BIH database, the detection rate of QRS complexes is above 99.9%. The negative influence of various noise types, like 50/60 Hz power line, abrupt baseline shift or drift, and low sampling rate was almost completely eliminated. The vector support machine system allow a good balance between compressing and diagnostic performance and the obtained results can form a solid base for better data storage in clinical environment.

1 Introduction

The computerized ECG signal processing, after several years of significant progress, can be considered a well-developed application. An efficient real-time analyzer and encoder system, based on filtering, beat detection (recognition and clustering), classification, storage and diagnosis, must be able to evaluate the signal with maximum few seconds delay to recognize in time the potentially dangerous and life threatening arrhythmia. Despite the presence of serious noise, a reliable analysis must involve at least the detection of QRS complex, T and P waves, automatic rhythm analysis, classification and diagnosis, allowing physicians to derive more information for cardiac disease diagnosis. It is important to determine the correct position and amplitude of every characteristic event.

An optimal computerized ECG filtering algorithm's performance mainly depends on the ability to separate the signal from artifacts, and from the amount and nature of distortion introduced by the filter. A post-filtering step is essential to reduce the signal distortion. Both guidelines are quite hard to evaluate, because the diagnosis is subjective and depends on the shape of the ECG signal.

The most important task in the ECG signal processing is the accurate detection of the QRS complexes. All further processing steps are based on the position of the QRS waves as basic information. Unfortunately the recorded ECG is often disturbed from different kinds of noises. Data corrupted with noise must be pre-filtered or discarded. The ECG quality assurance requires human and artificial noise detection schemes in

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 737–745, 2007.
springerlink.com

order not to lose clinically significant information. During ECG recording the noise can only be diminished but not eliminated, so it is important to use a method with good noise susceptibility. Due to the non-linear behavior of the human body, all processing methods must be capable to change their state. The design of an optimal matched filter can increase the signal-to-noise ratio, but the non-stationary nature of the signal and noise in an ECG represents an obstacle in the application of these filters for QRS detection. A linear filter cannot whiten the non-linear ECG signal effectively.

Artificial neural networks (ANN) [1] are inherently non-linear models, so an ANN-based filtering is potentially useful. In practical use, the ANN model can adapt far better than linear models. The number of input units corresponds to the filter order that should not be increased too much, in order to allow constantly good transient properties. It is important to choose the right number of hidden layers to allow good learning speed and adaptation at the same time. After pre-processing, filtering, evaluation and model's parameter estimation, the signal reconstruction is needed. In this step the post-filtering method knows the main ECG specific information, and can better separate all artificial noises. For a high performing filter it is necessary to use all ECG and patient depending information. This problem can be handled only if the computer knows the formation of the ECG signal.

In most cases, due to the collected noise during measurement, there is almost no reason to use loss-free compression [2]. In this paper we focused on loosely methods as a compromise between bandwidth and final reconstruction possibility, using sophisticated medical knowledge-based reconstruction algorithms [3]. The signal's main characteristics are represented by exponential parameterization that is delivered by processing system that uses support vector machine (SVM) [4]. This robust model involves the filtering, analysis and compression step of an automatic ECG evaluation method.

2 Methods

The proposed complex ECG signal compression algorithm can be divided into the following steps (see Fig. 1):

- Pre-filtering;
- Segmentation into R-R intervals;
- Create/update a temporal template bank for QRS beats;
- Optimal filter using QRS wave pattern database;
- Determine all recognizable characteristic points (for R, T, and P waves);
- Extract the waveform estimation model's parameters;
- Post-filtering using pattern database and the model-based estimation;
- Complete the template bank for all recognized waves;
- Adaptive smoothing;
- Residual signal estimation, entropy coding and back-estimation.

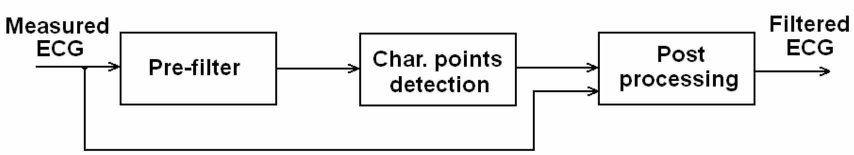


Fig. 1. The filtering process of the ECG signal

In our approach, because the studied signal has a non-linear behavior, we define a non-linear adaptive estimation algorithm. The main drawback of this method is the cumbersomeness to determine the optimal phase of the 50 (60) Hz component of the measured ECG signal, but paper [5] present how to handle this problem. In elimination of high frequency noise the non-linear parameter estimation methods could reach better performance, than transformation methods.

Let $\hat{X}_L(n)$ and $\hat{X}_R(n)$ the *n*-th left and right aimed estimation:

$$\hat{X}_L(n) = p_L\tilde{X}_L(n) = p_L\sum_{i=-q}^{q} a_{L,i}X(n-i) + (1-p_L)\sum_{i=1}^{q} b_{L,i}\hat{X}_L(n-i) \tag{1}$$

$$\hat{X}_R(n) = p_R\tilde{X}_R(n) = p_R\sum_{i=-q}^{q} a_{R,i}X(n-i) + (1-p_R)\sum_{i=1}^{q} b_{R,i}\hat{X}_R(n+i) \tag{2}$$

where $a_{L,i}$, $a_{R,i}$, $b_{L,i}$ and $b_{R,i}$ are prediction coefficients, p_L and p_R are balance probabilities determined by the dispersions $\sigma_{\hat{X}_L-X}(n,l)$, $\sigma_{\hat{X}_R-X}(n,l)$, $\sigma_{\tilde{X}_L-X}(n,l)$ and $\sigma_{\tilde{X}_R-X}(n,l)$. For better separation of the signal from the noise, the length *l* should select more than one R-R period. During on-line processing, the estimation is delayed with at least $3\cdot q$ samples, but preferably with more than one R-R interval, in order to minimize the differences of the efficiency between $\hat{X}_L(n)$ and $\hat{X}_R(n)$; $(p_L + p_R = 1)$. The resulting sample $\hat{X}(n)$ is obtained as follows:

$$\hat{X}(n) = p\sum_{i=-q}^{q} a_i\hat{X}_L(n-i) + (1-p)\sum_{i=-q}^{q} b_i\hat{X}_R(n-i)\,. \tag{3}$$

In a heavily noise tainted environment, a parameter extraction model could be less robust, than a good transformation algorithm. One of the best transformation methods for R wave detection uses wavelets. The selected mother wavelet is:

$$\Psi(t) = \frac{1}{\sqrt{2\pi\sigma}}\cdot\exp\left(-\frac{t^2}{2\sigma}\right)\cdot\sin(\alpha\cdot t\cdot\exp(-\beta|t|))\,. \tag{4}$$

The parameters α and β are selected according to the highest frequency in ideal (noise free) ECG signal and σ is the dispersion, used to modify the wavelet's shape.

After the analysis of more than 100 recordings, we obtained as a good robust result $\alpha = 200 \cdot \pi$ and $\beta = 1/3$. The robustness in this step is far more important, than a local performing index. The WT depends upon two parameters, scale s and position τ.

The dyadic wavelet is determined by a scale s = 2^{j}, where j is an integer. The wavelet transform at scale s = 2^{j} is obtained by:

$$Wf(2^j, \tau) = \tfrac{1}{2^j} \cdot \int_{-\infty}^{\infty} f(t) \cdot \Psi^*(\tfrac{t-\tau}{2^j}) dt . \tag{5}$$

Experiments show that this kind of wavelet allows the creation of more robust algorithms than using conventional ones. This property is due to the sinusoidal term, which can realize a correlation not only with the adjacent beats (the ECG signal is more or less semi-periodical).

Largely the signal's power is included in the QRS beats, so a template collection is essential. During signal processing, the pre-constructed wave database should be alterable. Although automated waveform classification based on a decision-tree algorithm could produce remarkable results, the new self-organizing (SO) adaptive clustering [6] based method have several advantages:

- It is not susceptible to variations of beat morphology and temporal characteristics;
- It can perform a real-time unsupervised learning;
- It needs much less amount of knowledge for the same performance.

The clusters are built up according to the following rules:

- $\sigma_i \le \sigma_{Max}$; σ_{Max} is predetermined; i=0..n;
- The mean value of a cluster is M_i and is determined in such a way, that σ_i to be minimal;
- For every R (T and P) wave, which belongs to a cluster, $\|X\| = \sum_{i=0}^{n} \left(\frac{X_i - M_i}{\sigma_i} \right)^2 \le R_{MAX}$ where R_{MAX} is predetermined; X is a vector, representing a wave in the space.

During ECG processing, for each characteristic wave results an indicator vector $\overline{X}^t = (p_0(X), ..., p_{n-1}(X))$, where n is the number of clusters and $p_I(X)$ is the probability to belong to the cluster C_I, having the value:

$$p_I(X) = \prod_{k=0}^{7} \frac{1}{\sigma_{I,k}} \exp\left(-\frac{(X_k - M_{I,k})^2}{2\sigma_{I,k}} \right) . \tag{6}$$

The main problem is to handle properly the cases when the patient manifests unnormal QRS wave pattern. In this situation a deeper analysis is performed that uses a hearth model based signal estimation. The optimal filter is based on the pre-processed signal and the template bank. Let

$$\overline{X}(n) = \sum_{k=0}^{nr-1} \left(s_k \cdot \sum_{i=-q}^{q} a_{F,i} \cdot X(n-i) \right) \tag{7}$$

and

$$\tilde{X}(n) = p_{F,X-\tilde{X}}(n) \cdot \overline{X}(n) + \left(1 - p_{F,X-\tilde{X}}(n)\right) \cdot \sum_{i=-q}^{q} b_i B(m.i) \tag{8}$$

be the processed data. The low value of p_F $(p_F < 0.2)$ justifies the need of the collection *B*, whose *m*-th element has the maximal correlation value with $\overline{X}(n)$. The characteristic point localizer algorithm is performed in a similar way to the template bank building method. An important difference consists in the appliance manner of pre-filtered data. Firstly the template bank is created for every recognizable event. With the aid of pre-filtered data we could minimize the isoelectric line movement caused problem.

Such a pre-filtered and de-noised signal consist the entrance of a SVM trained ANN. This kind of formulation of learning leads to quadratic programming with linear constraints.

The problem of learning SVM [7] is formulated as a task of separating learning vectors X_i into two classes of destination values $d = +1$ or $d = -1$ using maximal possible separation margin, that allows a high robustness to the obtained solution. The maximization task of function [8, 9] $Q(\alpha)$ is defined as follows:

$$Q(\alpha) = \sum_{i=-p}^{p} \alpha_i - \tfrac{1}{2} \cdot \sum_{i=-p}^{p} \sum_{j=-p}^{p} \alpha_i \cdot \alpha_j \cdot d_i \cdot d_j \cdot K(x_i \cdot x_j), \tag{9}$$

with linear constraints $\sum_{i=-p}^{p} \alpha_i \cdot d_i = 0$, where $0 \le \alpha_i \le C$. The α values are Lagrange multipliers, and function *K* represents the kernel, *p* is the number of learning pairs and *C* is a user defined constant (in our study *C* was selected between 0.1 and 0.5). In this case we applied radial Gaussian kernel function. The output signal $y(x)$ of the SVM network in retrieval mode (after learning) is determined as the combination of kernels

$$y(x) = \sum_{i=1}^{N_{SV}} \alpha_{SVi} \cdot d_i \cdot K(x_{SVi} \cdot x) + w_{opt}, \tag{10}$$

where N_{SV} is the number of support vectors and w_{opt} is the optimal weight vector.

Although SVM separates the data into two classes, the recognition of more ones is straightforward by applying either 'one against one' or 'one against all' methods [10]. After the ANN is trained, we used it to estimate the ECG as an output of a whitening filter.

The non-linear intermediate result is:

$$Z_p(t) = f\left(\sum_{k=-j}^{j} c_{pk}(t)\cdot X(t+k)\right). \tag{11}$$

where $X_k(t) = Y(t+k)$, and $f()$ is a normalized Gauss function. The c_{pq} weight coefficients connect the input and the hidden layers.

The output of the filter is:

$$Y_w(t) = Y(t) - \hat{Y}(t) = Y(t) - \sum_{p=-i}^{i} c_p(t)\cdot f\left(\sum_{k=-j}^{j} c_{pk}(t)\cdot X(t+k)\right). \tag{12}$$

The adaptive behavior of the filter [2] is assured by the permanent variance of the genetic search method based upon least mean square (LMS) algorithm computed coefficients. Both the input signal and the selected template are processed through the main filter. During this process the template bank is changing adaptively. The whitened template is:

$$T_{w,r}(t) = T(r) - \sum_{p=-i}^{i} c_p(t)\cdot f\left(\sum_{k=-j}^{j} c_{pk}(t)\cdot T(r+k)\right) \tag{13}$$

where *r=j,...L-j*, and *L* is the size of the template. The output of the matched filter will be:

$$Y_w(t) = \sum_{r=j}^{L-j} T_{w,r}(t)\cdot Y_w(t-L+r). \tag{14}$$

After the signal is filtered, a smoothing operation should be performed to reduce the size of the compacted data. The compression strength should be selected in accordance with the diagnosis performance decrease from the recovered signal. The main aim of this algorithm is to decrease the length of the compressed signal and to keep the data quality as high as possible. Let be:

$$\tilde{X}_{[sm.]}(n) = \frac{1}{k}\left(\sum_{i=1}^{k-1} \tilde{X}_{[sm]}(n-i\cdot\tau) + \tilde{X}(n)\right) \tag{15}$$

where $k = 2^j; (j \in N)$, $\tau \in N$ and *N* is the natural set and $\tilde{X}(n) = Y_w(n)$. Normally, the adjacent samples are highly correlated, and we select the positive integer τ that minimizes the auto-correlation function of the ECG signal. Usually the sampling delay τ is about half a QRS complex duration.

The inverse transform is given by:

$$\tilde{X}(n) = k\cdot\tilde{X}_{[sm]} - \sum_{i=1}^{k-1}\tilde{X}_{[sm]}(n-i\cdot\tau). \tag{16}$$

In the meantime of the transform, the values of $\tilde{X}(n)$ and $\tilde{X}_{[sm]}(n)$ can be modified with $k/2$ in order to reduce the reconstruction error or the dispersion of the smoothed signal. The efficiency of this algorithm highly depends on the chosen values for k and τ.

Because the scatter of the filtered and optionally smoothed signal $\sigma_{\tilde{X}_{[sm]}}(n,l)$ is too high to allow sufficient compression rate, a linear prediction transform is needed. This method eliminates the redundancy due to correlation between adjacent samples and beats.

The resulting data

$$\begin{aligned} Y(n) = \quad & p_{E,\tilde{X}_{[sm]}-B(m)}(n) \cdot \sum_{i=1}^{q} a_{E,i}\tilde{X}(n-i) + \\ & + \left(1 - p_{E,\tilde{X}_{[sm]}-B(m)}(n)\right) \cdot \sum_{i=-q}^{q} b_{E,i} B(m,i) \end{aligned} \tag{17}$$

allow the calculation of the residual $r(n) = Y(n) - \tilde{X}_{[sm]}(n)$.

Verifying processes determine the compression caused performance decrease in accordance to square error and diagnostic robustness. More iteration should be calculated to determine the optimal set of parameters. In most cases the estimation errors have nearly normal distribution. In order to reduce the length of the residual data, an adaptive method-based entropy coding is needed. For every moment we determine the dispersion $\sigma_r(n,l)$ and the probability $p_{\sigma_r(n,l)}(r(n,l))$ of the errors. If the quantum $q = 2^u$ and u is the length in bits of the word, the output value is obtained by

$$\begin{aligned} N_{[act]}(n,k) = \quad & I_1(n-k+1) + p_1(n-k+1) \cdot I_2(n-k+2) + \\ & + \prod\nolimits_{i=1}^{k-1} p_i(n-k+i) \cdot I_k(n) \end{aligned} \tag{18}$$

using $p_i(n-k+i) = p(r(n-k+i),l)$ and $I_{k-i}(n-i) = \int_{-\infty}^{r(n-i)} p_{k-i}(n-i) \cdot dr$.

3 Results

For a better comparability, we used the MIT-BIH arrhythmia database for evaluation. All processing steps were tested on the ECG registrations of MIT-BIH database (sampled at 360 Hz with 11-bit resolution) and our own recordings (sampled at 200-500 Hz with 12-bit resolution).

Most of these files contain one or two channels. The performance is determined by the number of estimation parameters, smoothing strength, resolution and sampling rate. Due to hearth model estimation and the advanced verifying algorithm, the resulting file size is considerably decreased. The entropy coding can decrease at least 12 times the theoretical "waste", compared with Huffman coding, during signal compacting.

The characteristic wave recognition performance heavily depends from the level of allowed root mean square error during signal compression. Fig. 2 represents the level of detection error of P, T and R waves when the relative RMSE varies from 0 to 20%.

Table 1. Representation of the entropy coder's performance using 32 parameters

Measurement (MIT-BIH)	Theoretic entropy	Huffman code size	SVM-based coding
104	170889	202158	172823
105	167021	197384	168882
108	172093	204214	174734
201	156878	185962	158004
203	185872	218977	187322
222	168126	199880	169634
228	181774	214708	183019

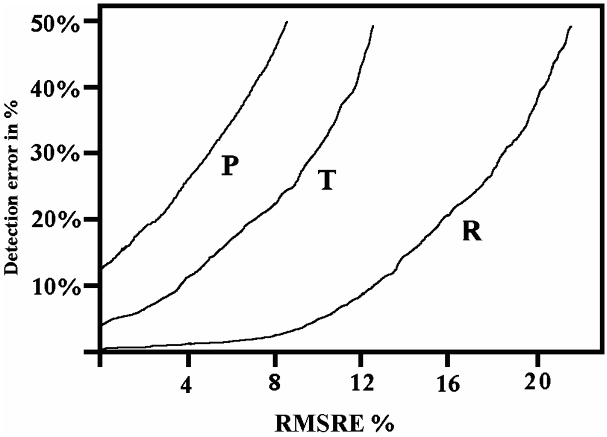


Fig. 2. The recognition ratio of R, T and P waves plotted against RMSRE

4 Discussion and Conclusions

Table 1 illustrates the compaction effectiveness for the most perturbed files, whose noise level and missed R wave detection rate was maximal. The new coding algorithm (equation: 18) has far better results than the adaptive Huffman coding. The elements distribution is close to normal and its change is not recommended without further knowledge. The smoothing strength should be selected by and (formula: 15, 16). Experiments show (see Fig. 2), that R wave can be accurately recognized even if RMSRE is about 10 %. For T and P wave detection the root mean square error must not exceed 3-5 % of the signal's power [11]. S (J), Q points and U wave cannot be recognized in most of the cases if RMSRE is higher than 1%.

An experimental real-time processing using this method needs a powerful computer able to perform massively parallel algorithms. In Holter telemetry and diagnostic systems [12], where a vast amount of data are acquired and transmitted by radio wave, the compression is an unavoidable step of the computerization. Sophisticated long computation and lingering unpack of the signal could be the major disadvantages of this process. Although quite often the calculation term doesn't admit on-line computerization (in multitasking mode), the rapid evolution of the computers will shortly change this fact.

References

1. Xue Q, Hu YH, Tompkins WJ (1992) Neural-network-based adaptive matched filtering for QRS detection. IEEE Trans Biomed Eng 39:317–329
2. Nave G, Cohen A (1993) ECG Compression Using Long-Term Prediction. IEEE Trans Biomed Eng 40:877–885
3. Szilágyi SM, Benyó Z, Dávid L (2003) ECG signal compression and noise distortion effect analysis. Proc World Congr Med Phys Biomed Eng, 4391.pdf
4. Osowski S, Hoai LT, Markiewicz T (2004) Support vector machine-based expert system for reliable heartbeat recognition. IEEE Trans Biomed Eng 51:582–589
5. Szilágyi SM, Benyó Z, Dávid L (2003) Iterative ECG filtering for better malfunction recognition and diagnosis. Proc 5th IFAC Symp Modell Contr Biomed Syst, pp 295–300
6. Lagerholm M, Peterson C, Braccini G, Edenbrandt L, Sörnmo L (2000) Clustering ECG complexes using Hermite functions and self-organizing maps. IEEE Trans Biomed Eng, 47:838–848
7. Vapnik V (1998) Statistical learning theory. New York: Wiley
8. Smola A, Scholkopf B (1998) A tutorial on support vector regression. Royal Holloway College, Univ London, NeuroColt Tech Rep, NV2-TR-1998-030
9. Burges CJC (2000) A tutorial on support vector machines for pattern recognition. In: Fayyad U (ed) Data mining and knowledge discovery, Kluwer Acad Publ, pp 1–43
10. Crammer K, Singer Y (2000) On the learnability and design of output codes for multiclass problems. Proc 13th Conf Comput Learning Theory, pp 35–46
11. Szilágyi SM, Szilágyi L (2000) Wavelet transform and neural-network-based adaptive filtering for QRS detection. Proc World Congr Med Phys Biomed Eng, pp 1267–1270
12. Szilágyi SM (1999) Non-linear adaptive prediction based ECG signal filtering, Proc 21st Ann Int Conf IEEE EMBS, p 286

Tuning FCMP to Elicit Novel Time Course Signatures in fMRI Neural Activation Studies

Mark D. Alexiuk[1], Nick J. Pizzi[2], and Witold Pedrycz[3]

[1] Electrical & Computer Engineering, E2-390 Engineering Building,
University of Manitoba, 75A Chancellor's Circle, Winnipeg MB CA, R3T 5V6
alexiuk@ieee.org
[2] Institute for Biodiagnostics, National Research Council,
435 Ellice Avenue, Winnipeg MB CA, R3B 1Y6
pizzi@nrc-cnrc.gc.ca
[3] Electrical & Computer Engineering, University of Alberta,
ECERF W2-032, Edmonton AB CA T6G 2V4
pedrycz@ece.ualberta.ca

Abstract. Functional magnetic resonance imaging (fMRI) is a preferred imaging modality to infer *in vivo* organ function from blood flow intensities. FMRI analysis is complex due to the variety of hemodynamic response models and the presence of noise. This complexity drives the use of exploratory data analysis (EDA) to elicit intrinsic data structure. This work demonstrates the utility of a fuzzy C-means (FCM) variant that incorporates feature partitions to generalize distance metrics across spatio-temporal features. This method, FCMP, exploits this relation to generate both novel and robust data inferences. A synthetic and a hybrid fMRI dataset are examined with results compared to an industry benchmark, EvIdent®. Efficacy of FCMP is shown in terms of tunable sensitivity to novel time course signatures and adaptability with which specific signatures are integrated into the objective function.

1 Introduction

Exploratory data analysis (EDA) [15] of functional magnetic resonance imaging (fMRI) data is challenging since two analysis of modes, pattern matching and novelty discovery, are conflated [10, 14]. The patterns to be matched relate to the paradigm time course (TC) which models the presence (or absence) of a stimulus to the subject. The paradigm is often modeled as a box car function and correlation quantifies similarity between TCs. Meanwhile, some responses have poor correlation to the paradigm but may provide key insight to the functional response [8]. These novel responses, possibly an higher order transformation of the paradigm [6], will not be elicited by a correlation approach and may not have been anticipated by the researcher. However, they may be captured by a fuzzy cluster in a suitably tuned clustering algorithm, even if the novel responses are relatively few in number. This paper examines the ability of fuzzy clustering with feature partitions (FCMP) [1,2] to capture novel responses present in a fMRI dataset.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 746–755, 2007.
springerlink.com

2 FCMP

Like other pattern recognition algorithms, fuzzy C-means (FCM) [4] is often preceded by preprocessing techniques. Such preprocessing compensates for the FCM implicit assumption that all sample features as equally significant [12]. Specialized treatment of features is sometimes introduced by modifications to the FCM objective function, through the use of alternate metrics [5], covariance matrices, additional terms and even in the choice of validation indices [9,11,13]. FCMP was developed with this in mind and contains a mechanism, the feature partition, to describe feature relationships and to integrate critical data structure information into the algorithm. For f features, a set of h, $1 \leq h \leq f$, feature partitions is represented as

$$\Psi=\{\psi_1, \psi_2 ... \psi_h\} \tag{1}$$

where each feature partition consists of a triple:

$$\psi_i=\{\mu_i, \nu_i, \rho_i\} \tag{2}$$

a metric μ_i, a weight ν_i, and a set of feature indices ρ_i. Particular features included in the subset (group) are encapsulated in ρ; μ defines distance between samples using only features in ρ, and ν is the weight, or relevance, of the distance μ to the objective function. Indices of features with similar meta-properties are grouped in ρ, which may contain, say, the spatial coordinates of a sample. The metric μ associated with the spatial coordinates should have some justification for use with the spatial concept of distance. Sample-sample distance is calculated using a generalized distance metric composed of a weighted sum of metrics over distinct feature indices. Each partition $p\in P$ uses a possibly unique distance function d_p. They are combined in the cluster algorithm using the weighting parameter $\nu=[\nu_1...\nu_j...\nu_h]$ $\forall j$, $0\leq\nu_i\leq1$. This results in a generalized distance

$$D(x,k)=\sum_{p\in P} \nu_p d_p^2(x,v) \tag{3}$$

where the distance between sample x and centroid v uses the partition appropriate metric μ_p and only features in ρ_p are considered.

In fMRI, and more generally, explicit accounting of feature relations benefits analysis by encoding meta-data such as modalities, conditions and other circumstances relating to data acquisition. It also recognizes that individual features belong to distinct statistical distributions, differ in observation error and noise, and often have a *de facto* comparison method associated with them. Embedding feature relationships in FCMP considers features with respect to their acquisition environment, preserve associations between features and distance concepts, and introduce non-linearities that aid discrimination. One benefit of using feature partitions is the ease at which existing analysis can be extended by an explicit feature relation and the resulting clarity of the effect of the relationship on the analysis. Differences in the FCM clustering equations are shown now for dataset X with n samples $x \in X$, C centroids $c,v \in V$, and h feature partitions $\Psi=\{\psi_1...\psi_h\}$.

The FCMP membership update equations includes a generalized distance metric composed of a weighted sum of distances on feature partitions

$$u_{ij} = \left(\sum_{b=1}^{C} \frac{\sum_{a=1}^{h} \nu_a \, d_a^2(x_i, v_j)}{\sum_{a=1}^{h} \nu_a \, d_a^2(x_i, v_b)} \right)^{m-1} = \left(\sum_{b=1}^{C} \frac{D(x_i, v_j)}{D(x_i, v_b)} \right)^{m-1} \tag{4}$$

The FCM centroid update equation remains unchanged for FCMP

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} x_i}{\sum_{i=1}^{n} u_{ij}^{m}} \tag{5}$$

3 FCMP and Buried Novelty

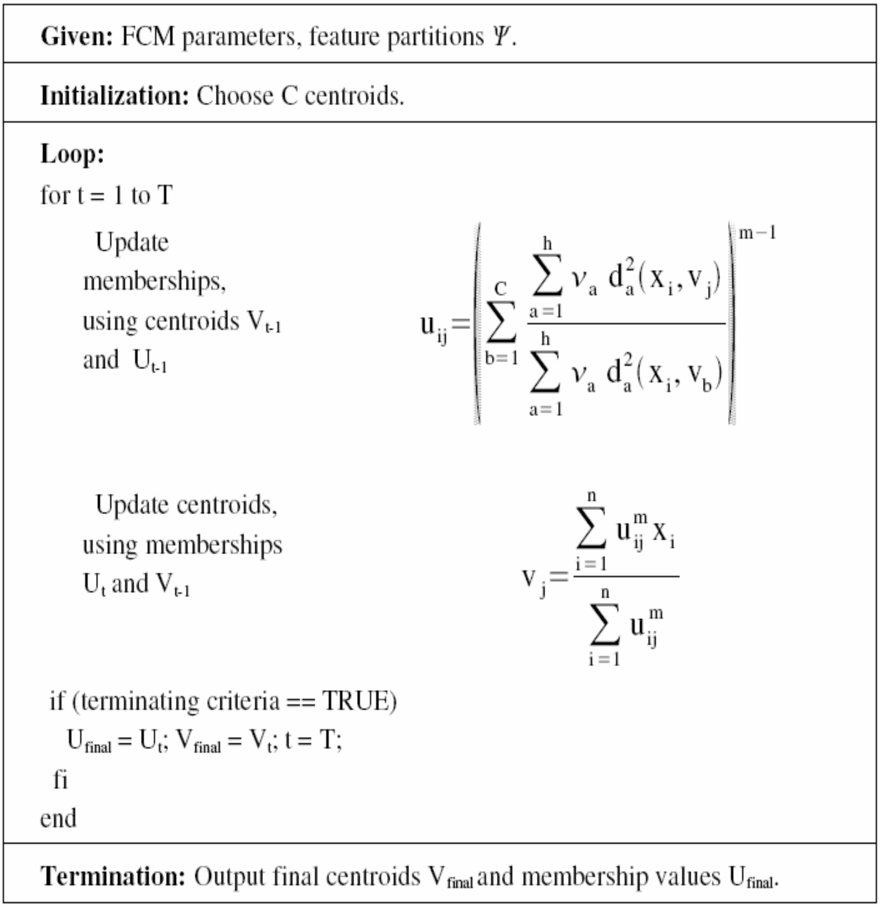
One example of small signal discovery is the detection of a small cluster novel TCs on the activated epochs while also clustering on the total temporal similarity of time courses and spatial proximity. The FCMP objective function can be specialized to focus on transient and minute phenomenon through the use of probes, designed terms appended to the objective function that selectively enhance sensitivity. For example, a distance metric can be tuned to feature types such as spatial coordinate, temporal intensity, (un)activated epochs. Feature partition weights can be modified.

Finally, consider a probe to be a thresholded metric or similarity function such that the output is binary signifying presence (1) or absence (0) of a TC signature. One difficult problem in fMRI analysis is searching for small signals non- linearly related to the paradigm. These small signals presumably reside in a small percent of TCs and may not form their own cluster with a general cluster algorithm. Detecting a cluster with a centroid highly correlated to the paradigm is insufficient to also identify time courses containing this small signal. However, a series of probes may be assembled to detect a variety of non-linear small signals. When the signal is detected in the clustering process (say, after a cluster has defined a centroid sufficiently similar to the paradigm), the probes, being based on the centroid, will be able to detect the small signal. Heuristics can be devised to change metrics when the probe indicates that the small signal is present. In this manner, the clustering process of the entire dataset is combined with a search for small related signals.

3.1 Benchmarks

By considering feature partitions of weight 0 and 1, FCMP reduces to FCM and as such FCM always forms edge parameter points on a parameter evaluation grid. An industry standard for exploring regions of interest in fMRI, EvIdent® will also be used to examine the datasets. EROICA is the main clustering alogirthm used in EvIdent®.

Table 1. The FCMP Algorithm



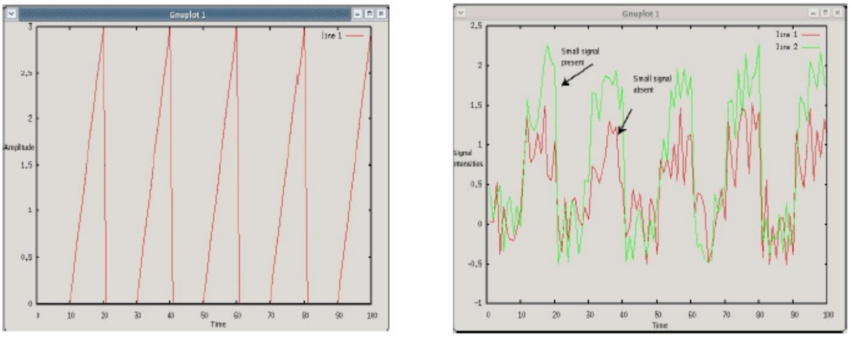
```
Given: FCM parameters, feature partitions $\Psi$.

Initialization: Choose C centroids.

Loop:
for t = 1 to T
    Update memberships, using centroids $V_{t-1}$ and $U_{t-1}$
        $u_{ij} = \left( \sum_{b=1}^{C} \frac{\sum_{a=1}^{h} \nu_a \, d_a^2(x_i, v_j)}{\sum_{a=1}^{h} \nu_a \, d_a^2(x_i, v_b)} \right)^{m-1}$

    Update centroids, using memberships $U_t$ and $V_{t-1}$
        $v_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} x_i}{\sum_{i=1}^{n} u_{ij}^{m}}$

 if (terminating criteria == TRUE)
   $U_{final} = U_t$; $V_{final} = V_t$; t = T;
 fi
end

Termination: Output final centroids $V_{final}$ and membership values $U_{final}$.
```

4 Datasets

Two datasets will be examined in this paper. Syn5 is a synthetic dataset while Baum-null is a hybrid (a fMRI null neural activation with super-imposed synthetic TC activations). Each is briefly discussed in turn.

4.1 Syn5

Syn5 is a synthetic fMRI dataset consisting of fifty noise TCs and fifty TCs that are degraded with respect to the activation paradigm. Each class of TCs is associated with 2-dimensional spatial features located in a circular radii of 0.3 with centres (0.1,0.1) and (0.9,0.9) respectively. The number of temporal features is 100. The SNR level of the degraded TCs is 5 with respect to the activation paradigm [0101010101]. Epoch lengths span 10 sampling instants.



(i) Ramp small signal added to activated epochs of degraded TCs.

(ii) Comparison of degraded TCs with and without the presence of the small signal.

Fig. 1. Small signals in Syn5

Of the fifty degraded TCs, ten have been modified to include a novel small signal only on the activated epochs. These TCs have random spatial coordinates and have had a ramp function added to their values, seen in Fig. 1 (i). A comparison between a degraded TCs with and without the ramp function is seen in Fig. 1 (ii).

A variant of this dataset was also used to fit the quad data format used by the Evident® interface. This modified dataset consists of a 40 by 40 x-y plane which has been partitioned into three classes: noise time courses, paradigm-correlated time courses, and paradigm-correlated time courses with a super-imposed small signal.

5 Baumnull

The BaumNull dataset was synthesized to demonstrate the high type1 (false positives) error rate associated with simple correlation analysis of fMRI data [3]. Two groups of TCs simulate the extreme case when TCs correlate highly to an activation paradigm but when the groups do not correlate at all to each other. The groups contain 46 and 26 TCs respectively and signal intensities over 120 time instants are used. A simulated hemodynamic response used two parameter gamma functions to generate responses to the activations in the respective TC groups. Thus, this dataset may be considered to have two activation paradigms.

A mean intensity image for BaumNull is shown in Fig. 2. Figure 2. shows the mean intensity image after it has been thresholded at fifty of the maximum value. The elongated and convoluted regions of interest generated by this thresholding are common in fMRI studies and correspond to the anatomical sulci structure. When neural regions are stimulated, the maxim that spatially proximal regions are temporally correlated must be modified to consider proximity in terms of anatomical structure, and not simply Euclidean distance. Large numbers of noise TC, those outside of regions of interest and even the entire brain, are significantly correlated to the activation paradigm. However, the distinguishing features of these voxels is the small number of their neighbours that also share significant correlations to the paradigm. Few of the TCs in the dataset have correlation values that would be considered significant in fields outside of fMRI analysis, in the range [0.4, 0.8].

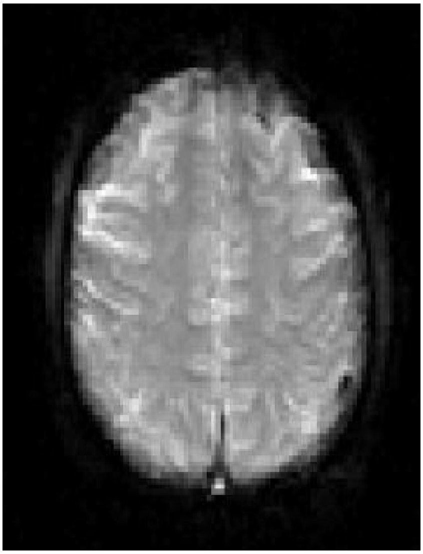


Fig. 1. BaumNull coronal plane, z = 0

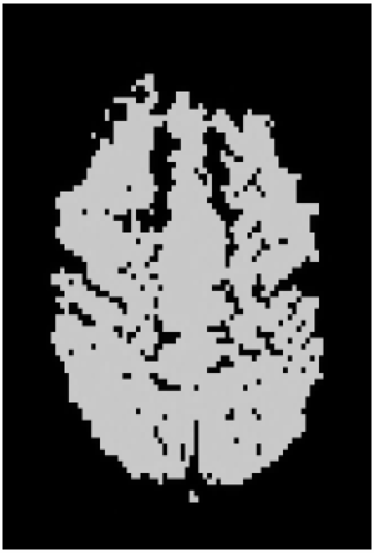
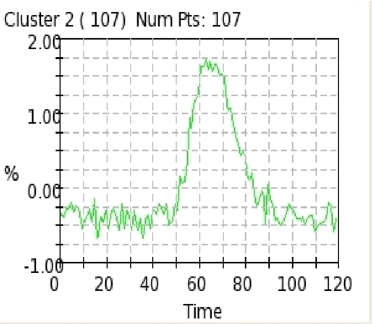
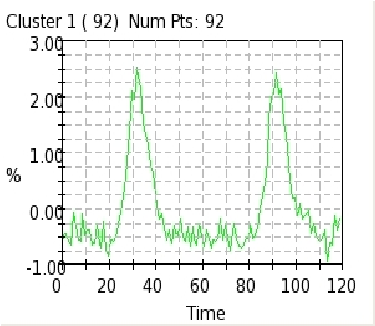
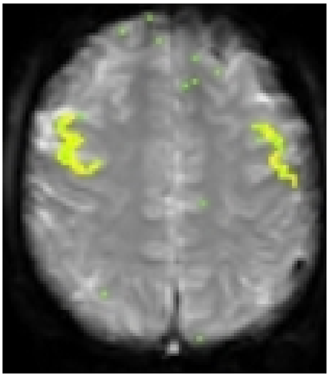
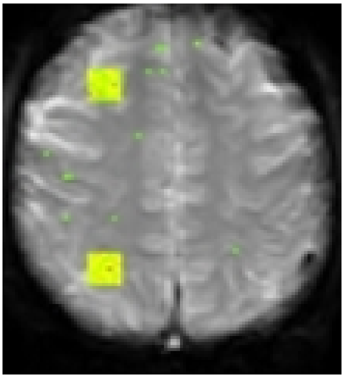


Fig. 2. BaumNull thresholded intensity image

Table 2. Paradigms and regions of interest for Baumnull dataset, as generated with EvIdent®

After scaling to [0,1000], TCs have a mean value of 135.603 and variance of 51305.7. Paradigms have patterns [000001000000] and [000100001000] respectively, see Table 2. Dataset dimensions (X,Y,Z,T) are (128, 128, 1, 120).

Noise TCs lie outside of the two artificial activated regions of interest. Outliers may be considered to be TCs outside of the ROI but that are somewhat correlated to one of the two activation paradigms. The focus of analysis for this dataset will be the interaction of distinct activation paradigms, not correlated to each other, in a synthetic, but typical, fMRI dataset.

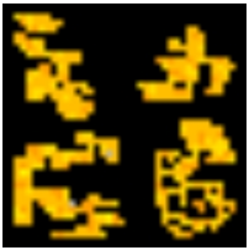
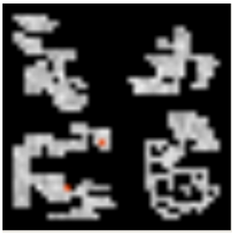
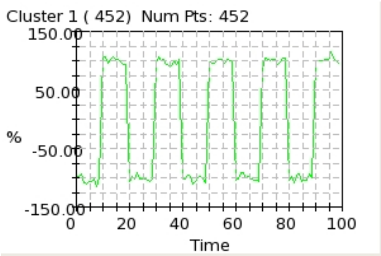
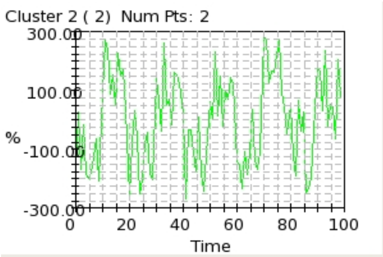
6 Experiment

In order to determine the degree to which clustering algorithms capture novelty while representing overall intrinsic data structure in an fMRI dataset, cluster centroids from FCM, FCMP and EvIdent® are examined on the datasets. Particular attention is paid to centroids correlation to the paradigm and novel TCs. Algorithms were implemented using Scopira^{TM} [7].

7 Results

Preliminary results suggest that feature partition weighting in FCMP is an effective method to tune algorithm sensitivity with respect to novel TCs when a specific attributes of the novelty are known, such as temporal signature, number of instances, and general spatial locations.

Table 3. Modified dataset syn5. Results from EvIdent® show ROI for paradigm-correlated TCs (top left), novel TCs, (top right), and their respective centroids.

7.1 Syn5 Results

EvIdent® has demonstrated the ability to detect the presence of novel signals in a small number of locations over a range of SNR values (40-2 dB). FCMP demonstrated a comparable ability only when the novel signals existed in a single location and shows the limit of a global spatial constraint on the distance measure. It is proposed that, when pattern matching and novelty discovery are being pursued simultaneously, the objective functions or distance metrics associated with the novelty contain the fewest number of constraints. In this case, the distance metric for the novel TCs should have ignored spatial location and attempted to match only temporal similarity while the distance metric for the correlated TCs was correct in constraining the TCs to also be spatially proximal. Correlation values for FCMP were as high as 0.99399 with significant correlation existing over a range of spatio-temporal weights.

7.2 Baumnull Results

FCMP centroid correlation values ranged from: 0.216828, 0.0843776, 0.0915909, 0.0950233, 0.0984579, 0.0907562, 0.0756187, 0.0586063, 0.0693817, 0.0449862, 0.0737046, as spatial-temporal weights ranged from: [1.0,0.0],[0.9,0.1] ... to [0.0,1.0]. Decreasing the p-value in the EROICA from 0.01 (default value) to 0.0001 was seen to effectively reduce proximal TCs not generated to be in the ROI. The mean and median TC filter were applied as preprocessing steps to EROICA and resulted in considerable variance in the resultant centroid significant spectrum see Table 4. SP denotes that spectral peak thresholding was used.

Table 4. EROICA centroids statistics for Baumnull dataset

EROICA Paramaters	*# TCs*	*Paradigm Match*	*ROI #TC in Paradigm 1*	*ROI #TC in Paradigm 2*
Default	116	1	92	107
Mean Filter	265	1	91	121
Median Filter	116	2	107	92
Default SP	95	2	99	88
Mean Filter SP	259	1	91	121
Median Filter SP	95	2	99	88

8 Conclusion

The FCMP objective function lends itself to adaptation for enhanced sensitivity to novel TCs in synthetic fMRI datasets and generates a centroid that correlates well to the novelty. This is advantageous since discovering novel TCs may trigger further analytic investigation. Novelty elicitation is related to minimizing the mathematical data model and requires an inclusive balance to outlier rejection preprocessing. FCMP uses the minimal mathematical model of FCM and shares the sample partitioning power of FCM on general data. FCMP captured novel intensity signatures by adjusting the distance metrics and weights in the feature partitions. Using a signature probe to trigger use of a different metric facilitates novelty discovery among a small number of samples while still capturing dataset structure. A priori knowledge of spectral properties, occurrence in activated (unactivated) epochs, or spatio-temporal heuristics can all be integrated in the FCMP objective function using feature partitions.

9 Future Work

Different types of novelty should be investigated and a group method of data handling (GMDH) type of infrastructure devised to discover the presence of novelties.

Acknowledgement

The National Science and Engineering Research Council of Canada (NSERC) is gratefully acknowledged for their financial support of this work.

References

1. Alexiuk, M.D., N.J. Pizzi, "Robust centroid determination of noisy data using FCM and domain specific partitioning," in *Proc 22nd Intl Conf North American Fuzzy Information Processing Society (NAFIPS)*, July 24-26, Chicago, 233-238, 2003.
2. Alexiuk, M.D. N.J. Pizzi, "Fuzzy C-Means with Feature Partitions: A Spatio-temporal Approach to Clustering fMRI data," *Pattern Recognition Letters*, 26, 1039-1046, 2005.
3. Baumgartner R., C. Windischberger, E. Moser, "Quanitification in Magnetic Resonance Imaging: Fuzzy clustering vs. Correlation Analysis," *Magnetic Resonance Imaging* 16(2), 115-125, 1998.
4. Bezdek J.C., "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms," *IEEE Trans PAMI,* 2(1), 1-8, 1980.
5. Bobrowski L., J.C. Bezdek, "C-Means Clustering with the L-1 and L-infinity Norms", *IEEE Trans SMC*, 545-554, 1991.
6. Chatfield C. *The Analysis of Time Series: An Introduction*, Chapman and Hall, New York, 1996.
7. Demko, A., N.J. Pizzi, R.L. Somorjai, "Scopira: A system for the analysis of biomedical data," in *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering, (CCECE 2002), Winnipeg, Canada, 12 - 15 May 2002* , Vol. 2 (Kinsner, W., Sebak, A. and Ferens, A., Eds.) Piscataway, IEEE, 1093-1098, 2002.
8. Duann J.R., T.P. Jung, W.J. Kuo, T.C. Yeh, S. Makeig, J.C. Hsieh, T.J. Sejnowski. Measuring variability of hemodynamics responses in event-related BOLD signals", 3rd International Conference on Independent Component Analysis and Blind Signal Separation, December 9-12, San Diego, *528-533, 2001.*
9. Höppner F., F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition, John Wiley and Sons, Ltd., New York, 1999.
10. Kuperman V., Magnetic Resonance Imaging: Physical Principles and Applications, Academic Press, New York, 2000.

11. Leski, J., “Towards a Robust Fuzzy Clustering,” Fuzzy Sets and Systems, 137(2), 215-233, 2003.
12. Pedrycz W., L.A. Zadeh, *Fuzzy Sets Engineering*, CRC Press, Inc. Boca Raton, 1995.
13. Pedrycz W., “Collaborative Fuzzy Clustering,” *Pattern Recognition Letters,* 23, 1675-1686, 2002.
14. Pizzi,N.J., M. Alexiuk, W. Pedrycz, “Stochastic Feature Selection for the Discrimination of Biomedical Spectra,” *Proc Int'l Joint Conf Neural Networks IJCNN'05,* July 31-August 4, Montréal, CA, 3029-3033, 2005.
15. Tukey J.W., *Exploratory Data Analysis*, Addison-Wesley Publishing Company, Reading, 1977.

Part XI

Robotics

Moving Object Tracking Using the Particle Filter and SOM in Robotic Space with Network Sensors

TaeSeok Jin[1] and JinWoo Park[2]

[1] Dept. of Mechatronics Engineering, DongSeo University,
San 69-1 Churye-dong, Sasang-ku, Busan 617-716, Korea
jints@dongseo.ac.kr
[2] Institute for Information Technology Advancement,
Hwaam-dong 58-4, Yusung-Gu, Daejeon 303-348, Korea
jinu@iita.re.kr

Abstract. Position estimation is one of the most important functions for the mobile robot navigating in Robotic Space. In order to achieve these goals, we present a method for representing, tracking and human following by fusing distributed multiple vision systems in Robotic Space, with application to pedestrian tracking in a crowd. And the article presents the integration of color distributions into SOM(Self Organizing Map) based particle filtering. Particle filters provide a robust tracking framework under ambiguity conditions. We propose to track the moving objects by generating hypotheses not in the image plan but on the top-view reconstruction of the scene. Comparative results on real video sequences show the advantage of our method for multi-motion tracking. Simulations are carried out to evaluate the proposed performance. Also, the method is applied to the intelligent environment and its performance is verified by the experiments.

1 Introduction

Detection of moving objects has been utilized in industrial robotic systems, for example, in the recognition and monitoring of unmanned systems that also require compression of moving images [1],[2]. Trajectory prediction of moving objects is required for a mobile manipulator that aims at the control and observation of motion information such as object position, velocity, and acceleration. Prediction and estimation algorithms have generally been required for industrial robots. For a simple example, in a pick-and-place operation with a manipulator, the precise motion estimation of the object on the conveyor belt is a critical factor in stable grasping [3],[4]. A well-structured environment, such as the moving-jig that carries the object on the conveyor belt and stops when the manipulator grasps the object, might obviate the motion estimation requirement. However, a well-structured environment limits the flexibility of the production system, requires skillful designers for the jig, and incurs a high maintenance expense; eventually it will disappear from automated production lines.

To overcome these problems, to tracking a moving object stably without stopping the motion, the trajectory prediction of the moving object on the conveyor belt is necessary [5]. The manipulator control system needs to estimate the most

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 759–768, 2007.
springerlink.com

accurate position, velocity, and acceleration at any instance to capture the moving object safely without collision and to pick up the object stably without slippage. When the motion trajectory is not highly random and continuous, it can be modeled analytically to predict the near-future values based on previously measured data [6]. However, this kind of approach requires significant computational time for high-degrees-of-freedom motion, and its computational complexity increases rapidly when there are many modeling errors. In addition, performance is highly sensitive to the change of the environment. Those state-of-the-art techniques perform efficiently to trace the movement of one or two moving objects but the operational efficiency decreases dramatically when tracking the movement of many moving objects because systems implementing multiple hypotheses and multiple targets suffer from a combinatorial explosion, rendering those approaches computationally very expensive for real-time object tracking [7].

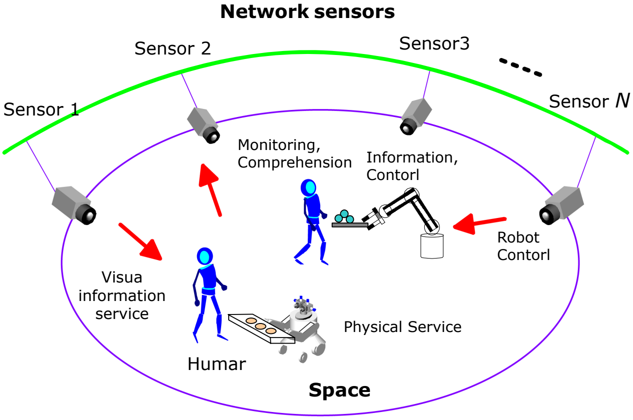


Fig. 1. Intelligent environment by distributed cameras

It is necessary for the intelligent environment to acquire various information about humans and robots in the environment. When the environment does not know where humans and robots are respectively, the environment can not give the enough service to the appropriate user as for the physical service especially. Therefore, it is considered that how to get the location information is the most necessary of all. The system with multiple color CCD cameras is utilized as one of the means to acquire the location information in an intelligent environment. It can achieve the human centered system because the environment acquires the location of human noncontactly and the equipment of the special devices isn't required for human. Moreover, camera has the advantage in wide monitoring area. It also leads to acquisition of details about objects and the behavior recognition according to image processing. Our intelligent environment is achieved by distributing small intelligent devices which don't affect the present living environment greatly.

2 Vision Systems in Robotic Space

2.1 Structure of Robotic Space

Fig. 2 shows the system configuration of distributed cameras in Intelligent Space. Since many autonomous cameras are distributed, this system is autonomous distributed system and has robustness and flexibility. Tracking and position estimation of objects is characterized as the basic function of each camera. Each camera must perform the basic function independently at least because over cooperation in basic level between cameras loses the robustness of autonomous distributed system. On the other hand, cooperation between many cameras is needed for accurate position estimation, control of the human following robot [8], guiding robots beyond the monitoring area of one camera [9], and so on. These are advanced functions of this system. This distributed camera system of Intelligent Space is separated into two parts as shown in Fig. 2. This paper will focus on the tracking of multiple objects in the basic function. Each camera has to perform the basic function independently of condition of other cameras, because of keeping the robustness and the flexibility of the system. On the other hand, cooperation between many cameras is needed for accurate position estimation, control of mobile robots to supporting human [10],[11], guiding robots beyond the monitoring area of one camera[12], and so on. These are advanced functions of this system.

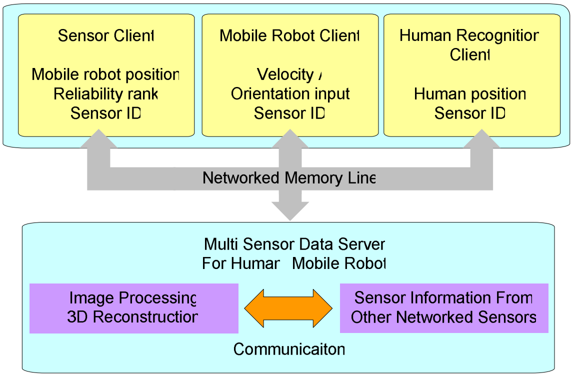


Fig. 2. Configuration of distributed camera system

2.2 Previous Research for Tracking

Neural networks can be classified into two categories: supervised learning- and unsupervised learning methods. In most of the previous research, the supervised learning method was adopted to overcome the nonlinear properties [10], [12]. Since the supervised learning method requires the relation between input and output [9] at all times, it is not suitable for real-time trajectory estimation for which the input-output relation cannot be obtained instantaneously in the unstructured environment. Therefore, in this study, SOM (Self Organizing Map), that is, a type of unsupervised learning method, was selected to estimate the highly nonlinear trajectory that cannot be properly predicted by the Particle filter. Also, SOM is a data-sorting algorithm, which is necessary for real-time image processing since there is so much data to be processed. Among the most popular data-sorting algorithms, VQ (Vector Quantization), SOM, and LVQ (Learning Vector Quantization), SOM is selected to sort the data in this approach since it is capable of unsupervised learning. Since VQ is limited to the very special case of a zero neighborhood and LVQ requires preliminary information for classes, neither of them is suitable for the unsupervised learning of the moving trajectory. Fig. 3 shows the estimation and tracking system for this research. The input for the dynamic model comes from either the Particle filter or SOM according to the following decision equation:

$$Predicted\,value = k \cdot ParticleFilter_{out} + (1-k) \cdot SOM_{out} \tag{1}$$

where k=1 for $error \le threshold$ and k=0 for $error > threshold$.

The threshold value is empirically determined based on the size of the estimated position error.

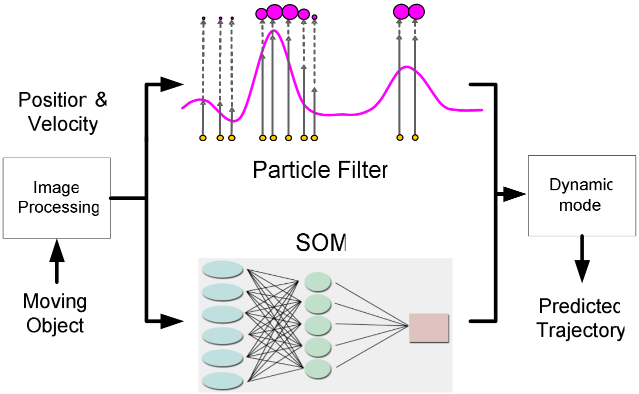
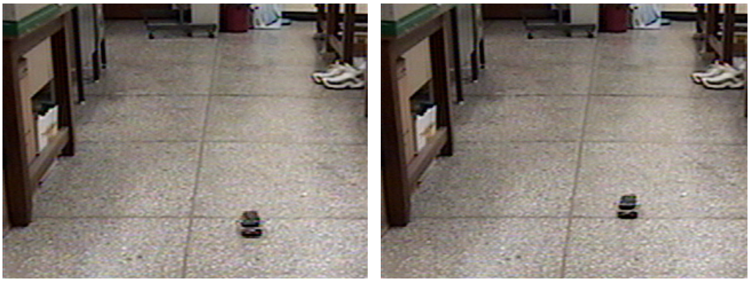


Fig. 3. Estimation and tracking system of robotic space

3 Processing Flow

3.1 Extraction of Object

Classifying the moving-object pattern in the dynamically changing unstructured environment has not yet been tackled successfully [13]. Therefore, in this research, the camera was fixed on a stable platform in order to capture static environment images. To estimate the states of the motion characteristics, the trajectory of the moving object was pre-recorded and analyzed. Fig. 4(a) and Fig. (b) represent the object images at (t-1) instance and (t) instance, respectively.



(a) (t-1) instance

(b) (t) instance

Fig. 4. Detected image of moving object at each instance

As recognized in the images, most parts of the CCD image correspond to the background. After eliminating the background, the difference between the two consecutive image frames can be obtained to estimate the moving-object motion. To compute the difference, either the absolute values of the two image frames or the assigned values can be used. The difference method is popular in image pre-processing for extracting desired information from the whole image frame, which can be expressed as

$$\mathrm{Output}(x, y) = \mathrm{Image1}(x, y) - \mathrm{Image2}(x, y) \tag{2}$$

The difference image between Fig. 4(a) and Fig. 4(b) is represented in Fig. 5. When the difference image for the whole time interval can be obtained, the trajectory of the moving object can be calculated precisely.

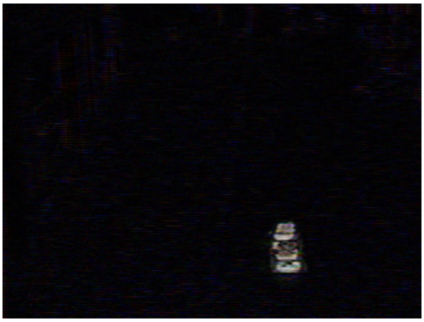


Fig. 5. Difference image between (t) and (t-1) instance images

3.2 Target Regions Encoded in a State Vector Using Particle Filter

Particle filtering provides a robust tracking framework, as it models uncertainty. Particle filters are very flexible in that they not require any assumptions about the probability distributions of data. In order to track moving objects (e.g. pedestrians) in video sequences, a classical particle filter continuously looks throughout the

$2D$-image space to determine which image regions belong to which moving objects (target regions). For that a moving region can be encoded in a state vector. In the tracking problem the object identity must be maintained throughout the video sequences. The image features used therefore can involve low-level or high-level approaches (such as the colored-based image features, a subspace image decomposition or appearance models) to build a state vector. A target region over the $2D$-image space can be represented for instance as follows:

$$\mathrm{r} = \{\mathrm{l},\ \mathrm{s},\ \mathrm{m},\ \gamma\} \tag{3}$$

where l is the location of the region, s is the region size, m is its motion and γ is its direction. In the standard formulation of the particle filter algorithm, the location l, of the hypothesis, is fixed in the prediction stage using only the previous approximation of the state density. Moreover, the importance of using an adaptive-target model to tackle the problems such as the occlusions and large-scale changes has been largely recognized. For example, the update of the target model can be implemented by the equation

$$\overline{\mathrm{r}}_t = (1-\lambda)\overline{\mathrm{r}}_{t-1} + \lambda \mathrm{E}[\mathrm{r}_t] \tag{4}$$

where λ weights the contribution of the mean state to the target region. So, we update the target model during slowly changing image observations.

4 Tracking Moving Objects

4.1 State-Space over the Top-View Plan

In a practical particle filter [5],[6] implementation, the prediction density is obtained by applying a dynamic model to the output of the previous time-step. This is appropriate when the hypothesis set approximation of the state density is accurate. But the random nature of the motion model induces some non-zero probability everywhere in state-space that the object is present at that point. The tracking error can be reduced by increasing the number of hypotheses (particles) with considerable influence on the computational complexity of the algorithm. However in the case of tracking pedestrians we propose to use the top-view information to refine the predictions and reduce the state-space, which permits an efficient discrete representation. In this top-view plan the displacements become Euclidean distances. The prediction can be defined according to the physical limitations of the pedestrians and their kinematics. In this paper we use a simpler dynamic model, where the actions of the pedestrians are modeled by incorporating internal (or personal) factors only. The displacements $\mathrm{M}^{t}_{topview}$ follows the expression

$$\mathrm{M}^{t}_{topview} = \mathrm{A}(\gamma_{topview})\mathrm{M}^{t-1}_{topview} + \mathrm{N} \tag{5}$$

where $\mathrm{A}(.)$ is the rotation matrix, $\gamma_{topview}$ is the rotation angle defined over top-view plan and follows a Gaussian function $g(\gamma_{topview};\sigma_{\gamma})$, and N is a stochastic component. This model proposes an anisotropic propagation of M : the highest probability is obtained by preserving the same direction. The evolution of a sample set is calculated by propagating each sample according to the dynamic model. So, that procedure generates the hypotheses.

4.2 Estimation of Region Size

The height and width of the robot can be obtained using geometric analysis.

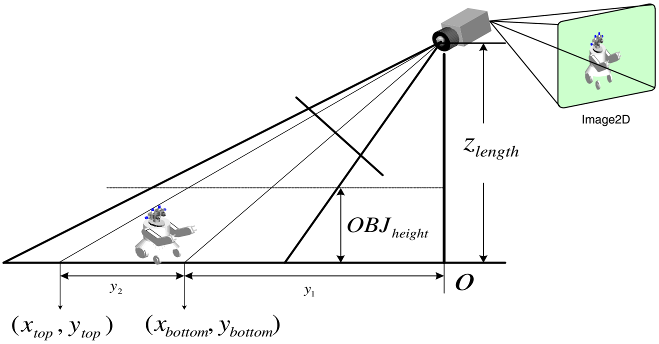


Fig. 6. Approximation of Top-view plan by image plan with a monocular camera

As shown in Fig. 6, the distance y_1 from the lowest coordinates of the object to the origin is calculated using y_{bottom} by geometric analysis as,

$$y_1 = y_{bottom} - O \tag{6}$$

where O represents the origin.

In the same manner, y_{top} can be calculated by replacing y_{bottom} as y_{top} and $P_{y,bottom}$ as $P_{y,top}$. Therefore, the distance y_2 from the highest coordinates of the object to y_{bottom} is calculated as,

$$y_2 = y_{top} - y_{bottom} . \tag{7}$$

When the coordinates, y_1 and y_2 are obtained, the height OBJ_{height} of the robot can be calculated as,

$$OBJ_{height} = \frac{z_{length} \times y_2}{(y_1 + y_2)} \tag{8}$$

from the similarity properties of triangles.

Following the same procedure, the width of the mobile robot can be obtained as follows: The real length $length_{pixel}$ per pixel is calculated as follow:

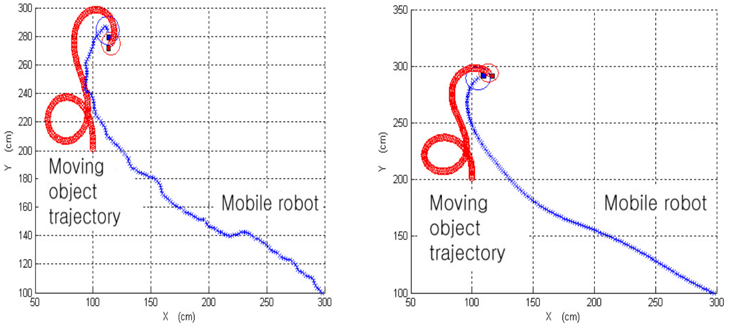
$$length_{pixel} = OBJ_{height} /(P_{y,top} - P_{y,bottom}) \,. \tag{9}$$

Then, the width, OBJ_{width} , of the object is calculated as

$$OBJ_{width} = length_{pixel} \times (p_{x,right} - p_{x,left}) \,. \tag{10}$$

5 Experiments

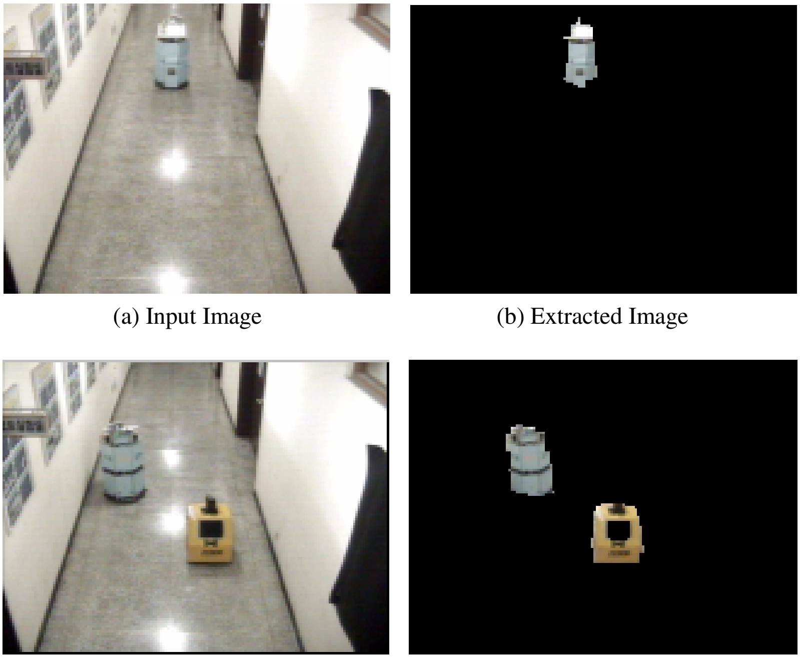
To compare the tracking performance of a mobile robot using the algorithms of the Particle filter and SOM, experiments of capturing a micro mouse with random motion by the mobile robot were performed. As can be recognized from Fig. 7, SOM based Particle Filter provided better performance to the mobile robot in capturing the random motion object than the other algorithms. As shown in Fig. 7, the mobile robot with SOM based Particle Filter has a smooth curve to capture the moving object. As the result, the capturing time for the moving object is the shortest with SOM based Particle Filter. Finally, as an application experiments were performed to show the tracking and capturing a mobile object in robotic space.



(a) SOM (b) SOM + Particle Filter

Fig. 7. Tracking trajectory by SOM and SOM based particle filter

Fig. 8 shows the experimental results for tracking a moving object that is an 8x6[cm] red-colored mouse and has two wheels with random velocities in the range of 25-35[cm/sec]. First, mobile robot detects the moving object using an active camera. When a moving object is detected within view, robot tracks it following the proposed method.



(a) Input Image

(b) Extracted Image

(c) Input Image with two robots

(d) Extracted Image with two robots

Fig. 8. Experimental results for tracking a moving object

6 Conclusions

In this paper, the proposed tracking method adds an adaptive appearance model based on color distributions to particle filtering. The color-based tracker can efficiently and successfully handle non-rigid and fast moving objects under different appearance changes. Moreover, as multiple hypotheses are processed, objects can be tracked well in cases of occlusion or clutter. This research proposes estimation and tracking scheme for a moving object using images captured by multi cameras. In the scheme, the state estimator has two algorithms: the particle filter that estimates the states for the linear approximated region, and SOM for the nonlinear region. The decision for the switchover is made based on the size of the position estimation error that becomes low enough for the linear region and becomes large enough for the nonlinear region. The effectiveness and superiority of the proposed algorithm was verified through experimental data and comparison. The adaptability of the algorithm was also observed during the experiments. For the sake of simplicity, this research was limited to the environment of a fixed-camera view. However, this can be expanded to the moving camera environment, where the input data might suffer from higher noises and uncertainties. As future research, selection of a precise learning pattern for SOM in order to improve the estimation accuracy and the recognition ratio, and development of an illumination robust image processing algorithm, remain.

References

1. Senior, A.: Tracking with Probabilistic Appearance Models. In Proc. ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems (2002) 48-55.
2. Bierlaire, M., Antonini, G., Weber, M.: Behavioural Dynamics for Pedestrians," in K. Axhausen (Ed.), Moving through nets: the physical and social dimensions of travel, Elsevier (2003) 1-18.
3. Nummiaro, K., Koller-Meier, E., Van Gool, L.J.: Object Tracking with an Adaptive Color-Based Particle Filter. DAGM-Symposium Pattern Recognition (2002) 353-360.
4. P. K. Allen, A. Tmcenko, B. Yoshimi, and P. Michelman, "Trajectory filtering and prediction for automated tracking and grasping of a moving object," *IEEE International Conference on Robotics and Automation* (1992) 1850-1856.
5. Yi Ma, J. Kosecka, and S. S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Transaction on Robotics and Automation*, vol. 15, no. 3, pp. 521-536, 1999.
6. Choo, K., Fleet, D.J.: People tracking using hybrid Monte Carlo filtering. In Proc. Int. Conf. Computer Vision, vol. II (2001) 321-328.
7. Anderson, B., Moore, J.: Optimal Filtering. Prentice-Hall, Englewood Cliffs (1979).
8. Kitagawa, G.: Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. Journal of Computational and Graphical Statistics, Vol. 5 (1996) 1-25.
9. Yi-Yuan Chen and Kuu-young Young, "An intelligent radar predictor for high-speed moving-target tracking," *TENCON '02. Proceedings. IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, vol. 3, pp. 1638 -1641, 2002.
10. J. M. Roberts, D. J. Mills, D. Charnley, and C. J. Harris, "Improved Kalman filter initialization using neuro-fuzzy estimation," *Int'l. Conf. on Artificial Neural Networks*, pp. 329-334, 1995.
11. Norlund, P., Eklundh, J.O.: Towards a Seeing Agent. Proc. of First Int. Workshop on Cooperative Distributed Vision (1997) 93-120.
12. Atsushi, N., Hirokazu, K., Shinsaku, H., Seiji, I,: Tracking Multiple People using Distributed Vision Systems. Proc. of the 2002 IEEE Int. Conf. on Robotics & Automation (2002) 2974-2981.
13. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: finder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19 (1997) 780-785.
14. Gardner, W.F., Lawton, D.T.: Interactive model based vehicle tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.18 (1996) 1115-1121.
15. Swain, M.J., Ballard, D.H.: Color indexing. Int. J. of Computer Vision, Vol.7, No.1, (1991)11-32.

Robust Stability Analysis of a Fuzzy Vehicle Lateral Control System Using Describing Function Method

Jau-Woei Perng[1], Bing-Fei Wu[1], Tien-Yu Liao[1], and Tsu-Tian Lee[2]

[1] Department of Electrical and Control Engineering, National Chiao Tung
University 1001, Ta-Hsueh Road, 300 Hsinchu, Taiwan
jwperng@cn.nctu.edu.tw, bwu@cc.nctu.edu.tw,
tienyu@cssp.cn.nctu.edu.tw

[2] Department of Electrical Engineering, National Taipei
University of Technology 1, Sec. 3, Chung-Hsiao E. Rd. 106 Taipei, Taiwan
ttlee@ntut.edu.tw

Abstract. In this paper, the robust stability analysis of a fuzzy vehicle lateral system with perturbed parameters is presented. Firstly, the fuzzy controller can be linearized by utilizing the describing function method with experiments. After the describing function is obtained, the stability analysis of the vehicle lateral control system with the variations of velocity and friction is then carried out by the use of parameter plane method. Afterward some limit cycle loci caused by the fuzzy controller can be easily pointed out in the parameter plane. Computer simulation shows the efficiency of this approach.

Keywords: Describing function, vehicle lateral system, fuzzy control, parameter plane, limit cycle.

1 Introduction

It is well known that the describing function is an useful frequency domain method for analyzing the stability of a nonlinear control system especially when the system has hard nonlinear elements, such as relay, deadzone, saturation, backlash, hysteresis and so on. The fundamental description of describing functions can be referred in [1-3]. Some academic and industry researches have been applied the describing function to meet the design specifications. For PI controller design, an iterative procedure for achieving gain and phase margin specifications has been presented in [4] based on two relay tests and describing function analysis. In [5], the describing function utilized for the stability analysis and limit cycle prediction of nonlinear control systems has been developed. The hysteresis describing function was applied to the class AD audio amplifier for modeling the inverter [6]. Ackermann and Bunte [7] employed the describing function to predict the limit cycles in the parameter plane of velocity and road tire friction coefficient. The algorithm computes the limit cycles for a wide class of uncertain nonlinear systems was considered in [8]. As for the intelligent control, some researchers have developed the experimental and

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 769–779, 2007.
springerlink.com

analytic describing functions of fuzzy controller in order to analyze the stability of fuzzy control systems [9-11]. Besides, the describing function was also applied to find the bounds for the neural network parameters to have a stable system response and generate limit cycles [12]. In fact, the uncertainties are often existed in the practical control systems. It is well known that the frequency domain algorithms of parameter plane and parameter space [13, 14] have been applied to fulfill the robust stability of an interval polynomial.

In this paper, we apply the describing function of fuzzy controller mentioned in [11] and parameter plane method [14] to consider the robust stability of a vehicle lateral control system with perturbed parameters. A systematic procedure is proposed to solve this problem. The characteristics of limit cycles can be found out in the parameter plane. Computer simulation results verify the design procedure and show the efficiency of this approach.

2 Description of Vehicle Model and Design Approach

In this section, the classical linearized single track vehicle model is given first. The describing function of fuzzy controller is also introduced. In order to analyze the stability of perturbed parameters, a systematic procedure is proposed to solve this problem by the use of parameter plane method. In addition, the control factors are also addressed.

2.1 Vehicle Model [7]

Figure 1 shows the single track vehicle model and the related symbols are listed in Table 1. The equations of motion are

$$\begin{bmatrix} mv(\dot{\beta}+r) \\ ml_f l_r \dot{r} \end{bmatrix} = \begin{bmatrix} F_f + F_r \\ F_f l_f - F_r l_r \end{bmatrix} \tag{1}$$

The tire force can be expressed as

$$F_f(\alpha_f) = \mu c_{f0} \alpha_f, \quad F_r(\alpha_r) = \mu c_{r0} \alpha_r \tag{2}$$

with the tire cornering stiffnesses c_{f0}, c_{r0}, the road adhesion factor μ and the tire side slip angles

$$\alpha_f = \delta_f - (\beta + \frac{l_f}{v} r), \alpha_r = -(\beta - \frac{l_r}{v} r) \tag{3}$$

The state equation of vehicle dynamics with β and r can be represented as

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{\mu(c_{f0}+c_{r0})}{mv} & -1+\frac{\mu(c_{r0}l_r - c_{f0}l_f)}{mv^2} \\ \frac{\mu(c_{r0}l_r - c_{f0}l_f)}{ml_f l_r} & -\mu\frac{(c_{f0}l_f^2 + c_{r0}l_r^2)}{ml_f l_r v} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} \frac{\mu c_{f0}}{mv} \\ \frac{\mu c_{f0}}{ml_r} \end{bmatrix} \delta_f \tag{4}$$

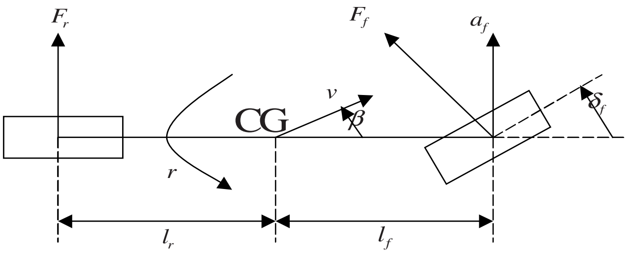


Fig. 1. Single track vehicle model

Table 1. Vehicle system quantities

F_f, F_r	lateral wheel force at front and rear wheel
r	yaw rate
β	side slip angle at center of gravity (CG)
v	velocity
a_f	lateral acceleration
l_f, l_r	distance from front and rear axis to CG
$l = l_f + l_r$	wheelbase
δ_f	front wheel steering angle
m	mass

Table 2. Vehicle parameters

c_{f0}	50000 N/rad
c_{r0}	100000 N/rad
m	1830 kg
l_f	1.51 m
l_r	1.32 m

Hence, the transfer function from δ_f to r is

$$G_{r/\delta_f} = \frac{c_{f0} m l_f \mu v^2 s + c_{f0} c_{r0} l \mu^2 v}{l_f l_r m^2 v^2 s^2 + l(c_{r0} l_r + c_{f0} l_f) m \mu v s + c_{f0} c_{r0} l^2 \mu^2 + (c_{r0} l_r - c_{f0} l_f) m \mu v^2} \tag{5}$$

The numerical data in this paper are listed in Table 2.

According to the above analysis of a single track vehicle model, the transfer function from the input of front deflection angle δ_f to the output of yaw rate r can be obtained as

$$G_{r/\delta_f}(s,\mu,v) = \frac{(1.382\times10^8 \mu v^2 s + 1.415\times10^{10} \mu^2 v)}{6.675\times10^6 v^2 s^2 + 1.08\times10^9 \mu v s + (1.034\times10^7 \mu v^2 + 4\times10^{10} \mu^2)} \tag{6}$$

The operating area Q of the uncertain parameters μ and v is depicted in Fig. 7.

In addition, the steering actuator is modeled as

$$G_A(s) = \frac{\omega_n^2}{s^2 + \sqrt{2}\omega_n s + \omega_n^2} \tag{7}$$

where $\omega_n = 4\pi$.

The open loop transfer function $G_O(s)$ is defined as

$$G_O(s,\mu,v) = G_A(s) G_{r/\delta_f}(s,\mu,v) \tag{8}$$

2.2 Describing Function of Fuzzy Controller

In this subsection, the fuzzy controller given in [11] is adopted here. Figure 2 shows the block diagram for determining the describing function of the fuzzy controller from experimental evaluations. The membership functions of the fuzzy controller are shown in Fig. 3 (a)-(c) and the rules of the fuzzy controller are listed in Table 3. Figure 4 shows the control surface. According to the analysis in [11], the describing function $N(A)$ with input signal ($x(t) = A\sin\omega t$) and scaling factors ($k_p = 6$, $k_d = 0.001$) can be obtained in Fig. 5.

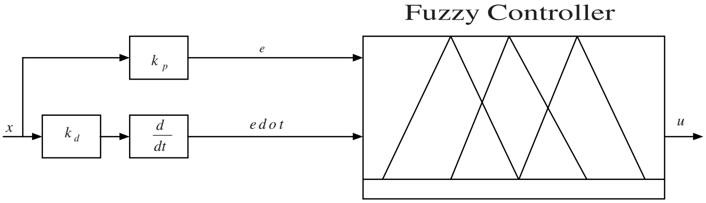
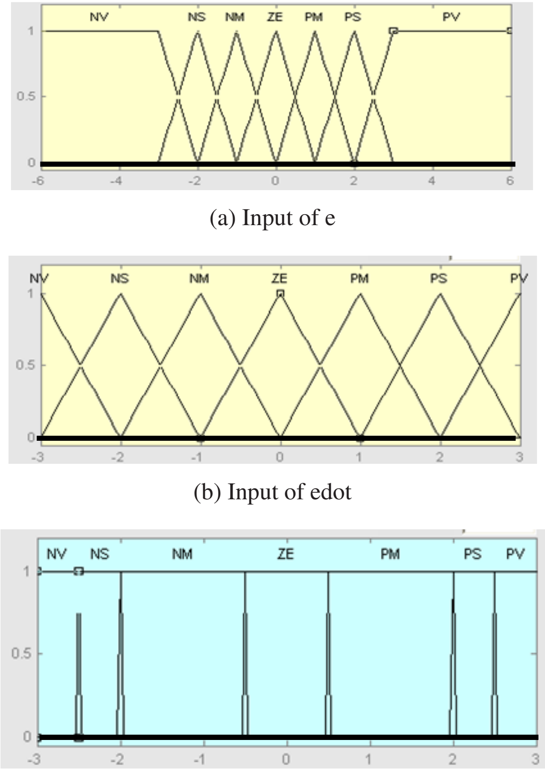


Fig. 2. Fuzzy Controller



(a) Input of e

(b) Input of edot

(c) Output of u

Fig. 3. Fuzzy membership functions

Table 3. Rules of the fuzzy controller

e edot	NV	NS	NM	ZE	PM	PS	PV
NV	NV	NV	NV	NV	NS	NM	ZE
NS	NV	NV	NV	NS	NM	ZE	PM
NM	NV	NV	NS	NM	ZE	PM	PS
ZE	NV	NS	NM	ZE	PM	PS	PV
PM	NS	NM	ZE	PM	PS	PV	PV
PS	NM	ZE	PM	PS	PV	PV	PV
PV	ZE	PM	PS	PV	PV	PV	PV

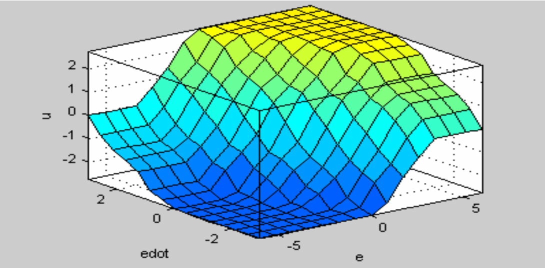


Fig. 4. Control Surface

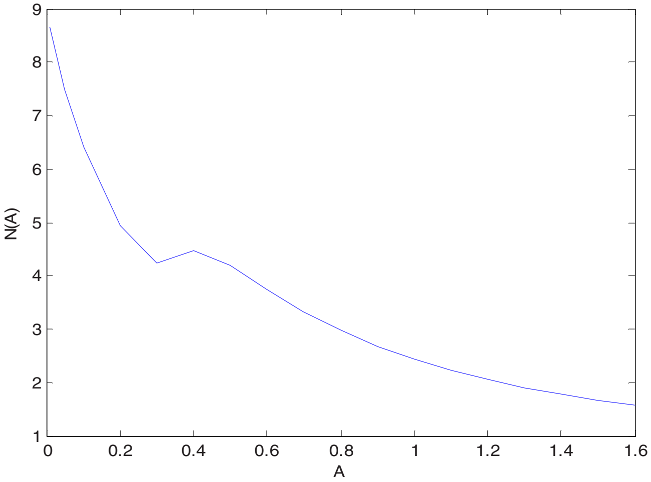


Fig. 5. Describing function of the fuzzy controller

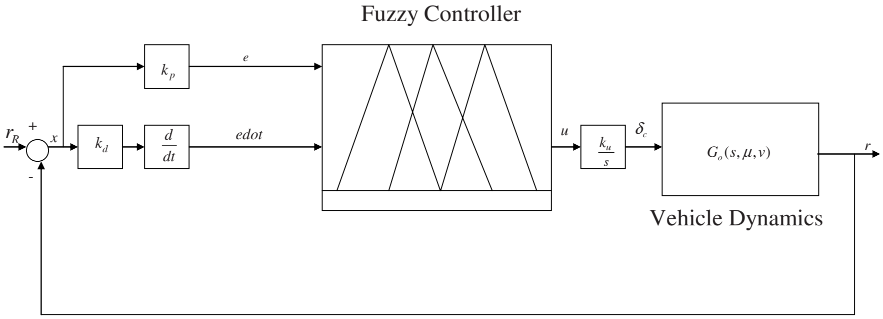


Fig. 6. Block diagram of a fuzzy vehicle lateral control system

2.3 Parameter Plane Method

Figure 6 shows the overall block diagram of a fuzzy vehicle lateral control system. In this study, $k_u = 0.1$ is selected. After some simple manipulations by using (5) to (8), the characteristic equation in Fig. 6 can be obtained as

$$\begin{aligned} & f(s,\mu,v) \\ & = C_4\mu^2 + C_3v^2 + C_2\mu v + C_1\mu^2 v + C_0\mu v^2 \ , \\ & = 0 \end{aligned} \tag{9}$$

where

$$\begin{aligned} C_4 &= 4.0045\times 10^{10} s(s^2+17.7715s+157.9137)\,, \\ C_3 &= 6.675\times 10^{6} s^3(s^2+17.7715s+157.9137)\,, \\ C_2 &= 1.0746\times 10^{9} s^2(s^2+17.7715s+157.9137)\,, \\ C_1 &= 2.2345\times 10^{11} N_1\,, \\ C_0 &= 1.03395\times 10^{8} s(s^2+17.7715s+157.9137) + 2.1818\times 10^{9} N_1 s\,. \end{aligned}$$

Let $s = j\omega$, (9) is divided into two stability equations with real part X and imaginary part Y of characteristic equation, one has

$$f(j\omega,\mu,v) = X + jY = 0 \tag{10}$$

where

$$\begin{aligned} X = & -7.1165\times 10^{11}\omega^2\mu^2 + 1.0075\times 10^{12}\omega^4 v^2 + (1.0746\times 10^{9}\omega^4 \\ & -1.6970\times 10^{11}\omega^2)\mu v + 2.2345\times 10^{11} N_1\mu^2 v - 1.8375\times 10^{9} N_1\omega^2\mu v^2, \\ Y = & (6.3236\times 10^{12}\omega - 4.0045\times 10^{10}\omega^3)\mu^2 + (6.6750\times 10^{6}\omega^5 \\ & -1.0541\times 10^{9}\omega^3)v^2 - 1.9098\times 10^{10}\omega^3\mu v + (1.6327\times 10^{10}\omega - 1.03395\times 10^{8}\omega^3 \\ & + 2.1818\times 10^{9} N_1\omega)\mu v^2. \end{aligned}$$

In order to obtain the solution of μ and v, the following equation is solved

$$\begin{cases} X = 0 \\ Y = 0 \end{cases}, \tag{11}$$

when k_p, k_d, k_u, N_1 are fixed and ω is changed from 0 to ∞. As the amplitude A is changed, the solutions of μ and v called limit cycle loci can be displayed in the parameter plane.

3 Simulation Results

In this work, the scaling factors of $k_p = 6$, $k_d = 0.001$, and $k_u = 0.1$ in Fig. 6 are selected. The membership functions and rules of fuzzy controller are the same with the above section. So the describing function of the fuzzy controller shown in Fig. 5 can be applied to analyze the stability of the closed loop system. After doing so, the solution of (11) can be solved when A is fixed and ω is changed from 0 to ∞. Figure 7 shows the operating area of the vehicle system. Two limit cycle loci in the μ - v parameter plane as $A = 0.1$ and $A = 0.2$ are plotted in Fig. 7, respectively. In order to test the predicted result in Fig. 7, eight operating points with Q1 ($\mu = 1$, $v = 67$), Q2($\mu = 1$, $v = 56$), Q3($\mu = 0.75$, $v = 65.8$), Q4($\mu = 0.63$, $v = 54.2$), Q5($\mu = 1$, $v = 30$), Q6($\mu = 0.37$, $v = 30$), Q7($\mu = 1$, $v = 5$), Q8($\mu = 0.1$, $v = 5$) are chosen for

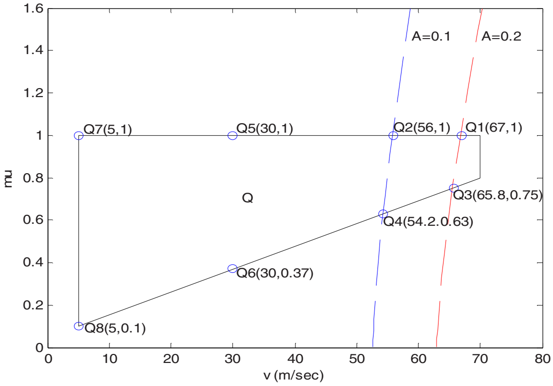


Fig. 7. Operating area and limit cycle loci

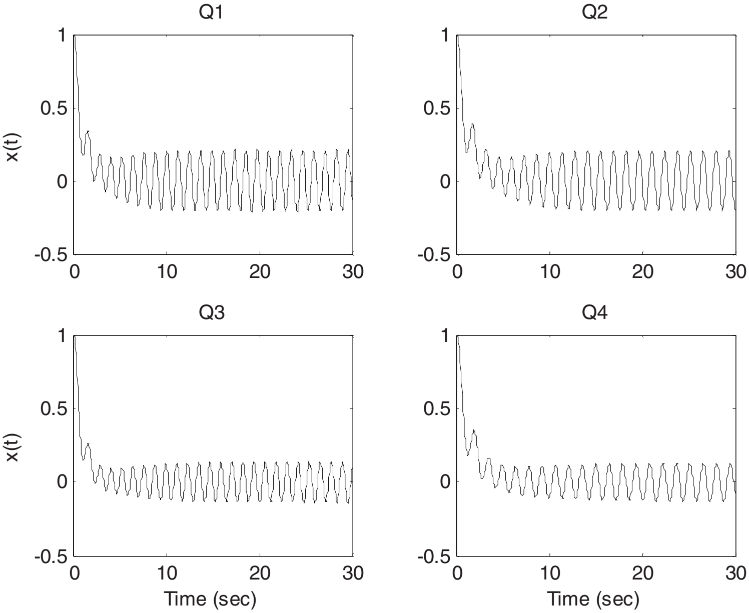


Fig. 8. Input signal of Q1 to Q4

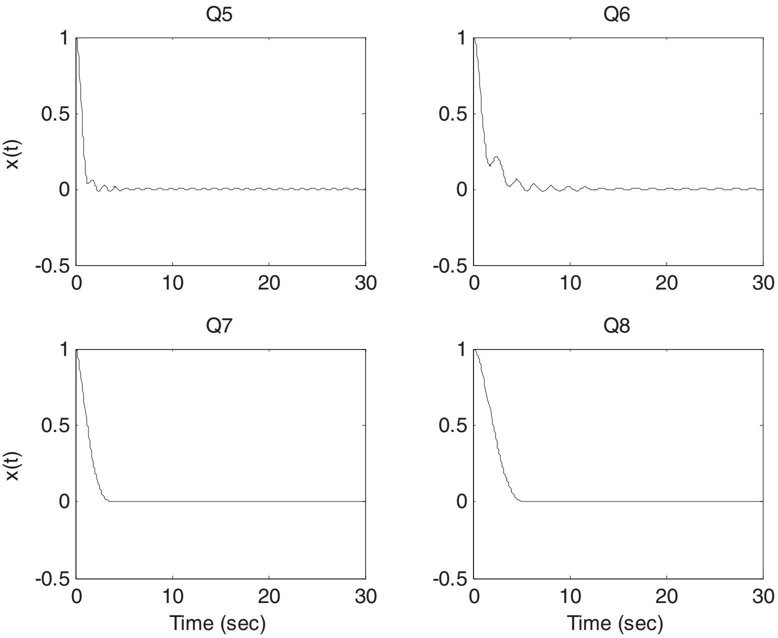


Fig. 9. Input signal of Q5 to Q8

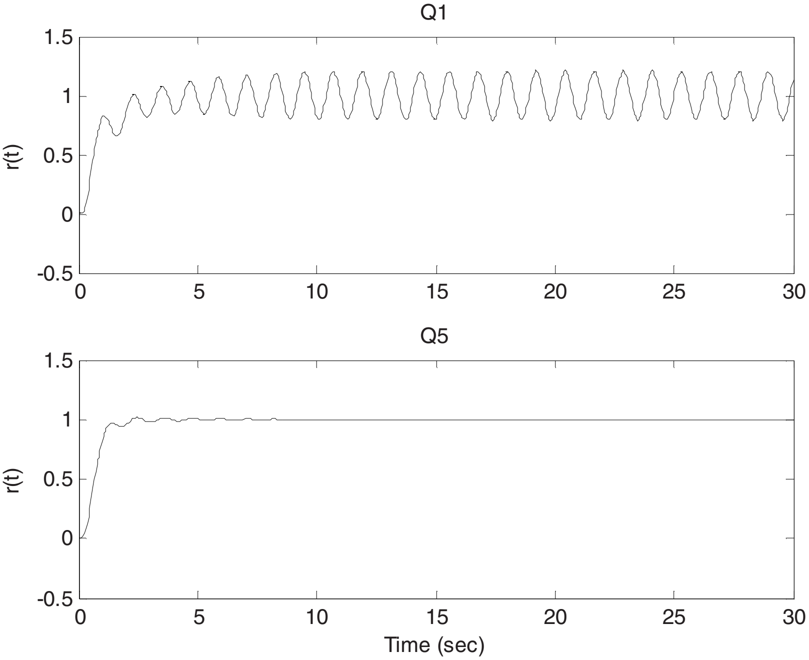


Fig. 10. Output signal of Q1 and Q5

the time simulations. Figure 8 shows the time responses of the input signal $x(t)$ at Q1 to Q4. The limit cycles are existed when operating at these four points and it consists with the predicted results (the amplitude of limit cycles) in Fig. 7.

On the other hands, the time responses of the input signal $x(t)$ at the other four points are displayed in Fig. 9. No limit cycle occurs in these four cases and this result is also matched with Fig. 7. Finally, the unit step responses of the output signal $r(t)$ at two operating points Q1 and Q5 are given in Fig. 10.

4 Conclusions

Based on the approaches of describing function and parameter plane, the robust stability of a fuzzy vehicle lateral system is considered in this paper. A systematic procedure is presented to deal with this problem. Simulation results show that more information about the characteristic of limit cycle can be obtained by this approach.

Acknowledgments. This paper is supported by the National Science Council in Taiwan through Grant NSC95-2752-E009-010.

References

1. Gelb A, Velde W. E. V.: Multiple Input Describing Functions and Nonlinear System Design. McGraw-Hill, New York (1968)
2. Siljak, D. D.: Nonlinear Systems - The Parameter Analysis and Design. New York: Wiley (1969).
3. Atherton, D. P.: Nonlinear Control Engineering. Van Nostrand Reinhold Company, London (1975)
4. Arruda, G. H. M., Barros, P. R.: Relay-Based Gain and Phase Margins PI Controller Design. IEEE Trans. Instru, and Measure. Vol 52 5 (2003) 1548-1553
5. Krenz, G. S., Miller, R. K.: Qualitative Analysis of Oscillations in Nonlinear Control Systems: A Describing Function Approach. IEEE Trans. Circuits and Systems. Vol. CAS-33 5 (1986) 562-566
6. Ginart, A. E., Bass, R. M., Leach, W. M. J., Habetler, T. G.: Analysis of the Class AD Audio Amplifier Including Hysteresis Effects. IEEE Trans. Power Electronics. Vol. 18 2 (2003) 679-685
7. Ackermann, J., Bunte, T.: Actuator Rate Limits in Robust Car Steering Control. Prof. IEEE Conf. Decision. and Control. (1997) 4726-4731
8. Nataraj, P. S. V., Barve, J. J.: Limit Cycle Computation for Describing Function Approximable Nonlinear Systems with Box-Constrained Parametric Uncertainties. Int. J. of Robust and Nonlinear Cont. Vol. 15 (2005) 437-457
9. Gordillo, F., Aracil, J., Alamo, T.: Determining Limit Cycles in Fuzzy Control Systems. Proc. IEEE Int. Conf. Fuzzy Syst. Vol. 1 (1997) 193-198
10. Kim, E., Lee, H., Park, M.: Limit-Cycle Prediction of a Fuzzy Control System Based on Describing Function Method. IEEE Trans. Fuzzy Syst. Vol. 8 1 (2000) 11-21
11. Sijak, T., Kuljaca, O., Tesnjak, S.: Stability Analysis of Fuzzy Control System Using Describing Function Method. MED2001. Dubrovnik, Croatia (2001)

12. Delgado, A.: Stability Analysis of Neurocontrol Systems Using a Describing Function. Proc. Int. Join. Conf. Neural Network. Vol. 3 (1998) 2126-2130
13. Siljak, D. D.: Parameter Space Methods for Robust Control Design: A Guide Tour. IEEE Trans. Automat. Contr. Vol. 34 7 (1989) 674-688
14. Han, K. W.: Nonlinear Control Systems - Some Practical Methods. Academic Cultural Company, California (1977)

Self-tunable Fuzzy Inference System: A Comparative Study for a Drone

Hichem Maaref, Kadda Meguenni Zemalache, and Lotfi Beji

IBISC Laboratory, CNRS-FRE 2873, Université d'Evry, 40 rue du Pelvoux
91020 Evry Cedex, France
{hichem.maaref, kadda.zemalache, lotfi.beji}@ibisc.univ-evry.fr

Abstract. The work describes an automatically on-line Self-Tunable Fuzzy Inference System (STFIS) of a mini-flying called XSF drone. A Fuzzy controller based on an on-line optimization of a zero order Takagi-Sugeno fuzzy inference system (FIS) by a back propagation-like algorithm is successfully applied. It is used to minimize a cost function that is made up of a quadratic error term and a weight decay term that prevents an excessive growth of parameters. Simulation results and a comparison with a Static Feedback Linearization controller (SFL) are presented and discussed. A path-like flying road, described as straight-lines with rounded corners permits to prove the effectiveness of the proposed control law.

Keywords: Fuzzy Inference System, Optimization, Static Feedback Linearization controller, Tracking control, Dynamic systems, Drone.

1 Introduction

Modelling and controlling aerial vehicles (blimps, mini rotorcraft, drone) are one of the principal preoccupation of the IBISC-group. Within this optic, the attracted contest of the DGA-ONERA is the XSF project which consists of a drone with revolving aerofoils [2], (Fig. 1 (left)). It is equipped with four rotors where two are directionals, what we call in the following X4 Bidirectional Rotors or X4 Stationary Flyer (XSF). The XSF is an engine of $68cm \times 68cm$ of total size and not exceed $2Kg$ in mass. It is designed in a cross form and made of carbon fibber. Each tip of the cross has a rotor including an electric brushless motor, a speed controller and a two-blade ducted propeller. The operating principle of the XSF can be presented thus: two rotors turn clockwise, and the two other rotors turn counter clockwise to maintain the total equilibrium in yaw motion. A characteristic of the XSF compared to the existing quad-rotors, is the swivelling of the supports of the motors 1 and 3 around the pitching axis thanks to two small servomotors controlling the swivelling angles ξ_1 and ξ_3. This swivelling ensures either the horizontal rectilinear motion or the rotational movement around the yaw axis or a combination of these two movements which gives the turn (Fig. 2 (right)). This permits a more stabilized horizontal flight and a suitable cornering.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 780–789, 2007.
springerlink.com

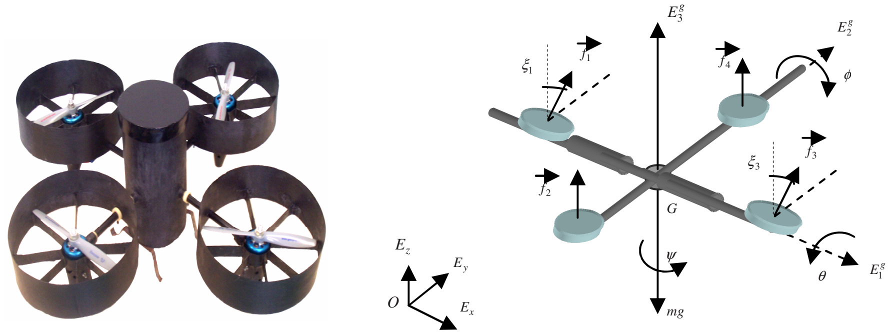


Fig. 1. Conceptual form of the XSF drone (left). Frames attached to the XSF (right).

Several recent works were completed for the design and control in pilotless aerial vehicles domain such that Quadrotor [1, 8], X4-flyer [3, 5]. A model for the dynamic and configuration stabilization of quasi-stationary flight conditions of a four rotors VTOL was studied by Hamel et al [5] where the dynamic motor effects are incorporating and a bound of perturbing errors is obtained for the coupled system.

All the reviewed techniques require the well knowledge of the system model and parameters. In this paper, a STFIS control strategy is presented based on the systems output measures. This technique early used for autonomous wheeled robot, is adapted for the use with the XSF [10, 11]. In autonomous wheeled robot, many developed learning techniques have arisen in order to generate or to tune fuzzy rules. Most of them are based on the so-called "Neuro-Fuzzy learning algorithms" as proposed by [6, 7].

In this paper, the Self-Tunable Fuzzy Inference System (STFIS) controller is presented and compared with a Static Feedback Linearization controller (SFL) to stabilize the XSF by using the point to point steering stabilization. The dynamic model of the XSF drone is briefly given in the second section. The developed ideas of control for the XSF by the STFIS and the SFL controllers are described in the third section. Motion planning and simulation results are introduced in the fourth section. The robustness of the proposed controller is then evaluated in the fifth section. Finally, conclusion and future works are given in the last section.

2 Motion Dynamic

We consider the translation motion of $\Re_G$ with respect to $\Re_O$. The position of the mass centre wrt $\Re_O$ is defined by $\overline{OG} = (x, y, z)^T$, its time derivative gives the velocity wrt to $\Re_O$ such that $d\overline{OG}/dt = (\dot{x}, \dot{y}, \dot{z})^T$, while the second time derivative permits to get the acceleration $d^2\overline{OG}/dt^2 = (\ddot{x}, \ddot{y}, \ddot{z})^T$ (Fig. 1 (right)).

The model is a simplified one's. The constraints as external perturbation and the gyroscopic torques are neglected. The aim is to control the engine vertically along z

axis and horizontally according to x and y axis. The vehicle dynamics, represented on Figure 1 (right), is modelled by the system of equations $Eq.\,1$, [2, 3, 9].

$$\begin{aligned}
&m\ddot{x} = S_{\psi} C_{\theta} u_2 - S_{\theta} u_3 \\
&m\ddot{y} = \left(S_{\theta} S_{\psi} S_{\phi}\right) u_2 + \left(C_{\theta} S\phi\right) u_3 \\
&m\ddot{z} = \left(S_{\theta} S_{\psi} C_{\phi}\right) u_2 + \left(C_{\theta} C\phi\right) u_3 \\
&\ddot{\theta} = \tilde{\tau}_{\theta}; \quad \ddot{\phi} = \tilde{\tau}_{\phi}; \quad \ddot{\psi} = \tilde{\tau}_{\psi}
\end{aligned} \tag{1}$$

Where C_{θ} and S_{θ} represent $\cos\theta$ and $\sin\theta$, respectively. m is the total mass of the XSF. The vector u_2 and u_3 combines the principal non conservative forces applied to the engine airframe including forces generated by the motors and drag terms. Drag forces and gyroscopic due to motors effects are not considered in this work.

3 Advanced Strategies of Control

The aim is to make comparison between model based approaches and experts analysis involving fuzzy systems. Classical model based techniques such as the static feedback linearization and backstepping techniques have been investigated and used for stabilization with motion planning [3, 9].

3.1 Static Feedback Linearization Controller

Control input for z-y motions: We propose to control the y/z motion through the input u_2/u_3. So, we have the following proposition.

Proposition 1. Consider $(\psi,\theta) \in \left]-\pi/2, \pi/2\right[$, with the static feedback laws.

$$\begin{aligned}
&u_2 = m v_y C_{\phi} C_{\psi}^{-1} - m\left(v_z + g\right) S_{\phi} C_{\psi}^{-1} \\
&u_3 = m v_y \left(S_{\phi} C_{\theta}^{-1} - C_{\phi} tg_{\psi} tg_{\theta}\right) + m\left(v_z + g\right)\left(C_{\phi} C_{\theta}^{-1} - S_{\phi} tg_{\psi} tg_{\theta}\right)
\end{aligned} \tag{2}$$

The dynamic of y and z are linearly decoupled and exponentially-asymptotically stable with the appropriate choice of the gain controller parameters. v_y and v_z are detailed in the following.

We can regroup the two dynamics as:

$$\begin{pmatrix} \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} S_{\theta} S_{\psi} S_{\phi} + C_{\psi} C_{\theta} & C_{\theta} S_{\phi} \\ S_{\theta} S_{\psi} C_{\phi} - C_{\psi} S_{\theta} & C_{\theta} C_{\phi} \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} - \begin{pmatrix} 0 \\ g \end{pmatrix} \tag{3}$$

For the given conditions in ψ and θ the (2×2) matrix in $Eq.\,3$ is invertible. Then a nonlinear decoupling feedback permits to write the following decoupled linear dynamics.

$$\begin{aligned} \ddot{y} &= \nu_y \\ \ddot{z} &= \nu_z \end{aligned} \tag{4}$$

Then we can deduce from $Eq.\ 4$ the linear controller:

$$\begin{aligned} \nu_y &= \ddot{y}_r - k_y^1(\dot{y} - \dot{y}_r) - k_y^2(y - y_r) \\ \nu_z &= \ddot{z}_r - k_z^1(\dot{z} - \dot{z}_r) - k_z^2(z - z_r) \end{aligned} \tag{5}$$

with the k_y^i, k_z^i are the coefficients of a Hurwitz polynomial.

Our second interest is the (x, z) dynamics which can be also decoupled by a static feedback law.

Input u_2 for the motion along x: The aim here is to stabilize the dynamics of x with the control input u_2. While we keep the input u_3 for the altitude z stabilization.

Proposition 2. With the new inputs

$$\begin{aligned} u_2 &= (S_\psi C_\phi - S_\theta C_\psi S_\phi)^{-1}(m\nu_x C_\phi C_\theta + m(\nu_z + g)S_\theta) \\ u_3 &= (S_\psi C_\phi - S_\theta C_\psi S_\phi)^{-1}(-m\nu_x(S_\psi S_\theta C_\phi - C_\psi C_\phi) + m(\nu_z + g)S_\psi C_\theta) \end{aligned} \tag{6}$$

The dynamic of x and z are decoupled and exponentially stable. ν_x and ν_z can be deduced as in $Eq.\ 7$ [3].

$$\begin{aligned} \ddot{x} &= \nu_x \\ \ddot{z} &= \nu_z \end{aligned} \tag{7}$$

Remark. To switch between the two controllers $Eq.\ 2$ and $Eq.\ 6$ continuity is recommended allowing the avoidance of the peak phenomenon. This can be asserted if we impose some constraints to the reference trajectories. In order to ensure this, we take $u_2(\psi = 0) = u_2(\psi = \pi/2) = 0$ with $\phi = \theta = 0$. For $\psi = 0$, one uses (2) and for $\psi = \pi/2$ expression $(Eq.\ 6)$ is considered. As soon as we get $u_3(\psi = 0) = u_3(\psi = \pi/2) = mg$ taking $\phi = \theta = 0$.

3.2 Self-tunable Fuzzy Inference System

A Fuzzy controller based on an on-line optimization of a zero order Takagi-Sugeno fuzzy inference system is successfully applied. It is used to minimize a cost function that is made up of a quadratic error term and a weight decay term that prevents an excessive growth of parameters of the consequent part. The main idea is to generate the conclusion parts w_i (so-called weight) of the rules automatically thanks to an optimization technique. The used method is based on a back-propagation algorithm where the parameters values are free to vary during the optimization process. The shape of the used membership functions is triangular and fixed in order to extract and represent the knowledge from the final results easily. To deduce the truth value, we use the MIN operator for the composition of the input variables. For the control of the XSF, we use the architecture known as “mini-JEAN” as illustrated in the Figure 2 (left) [7].

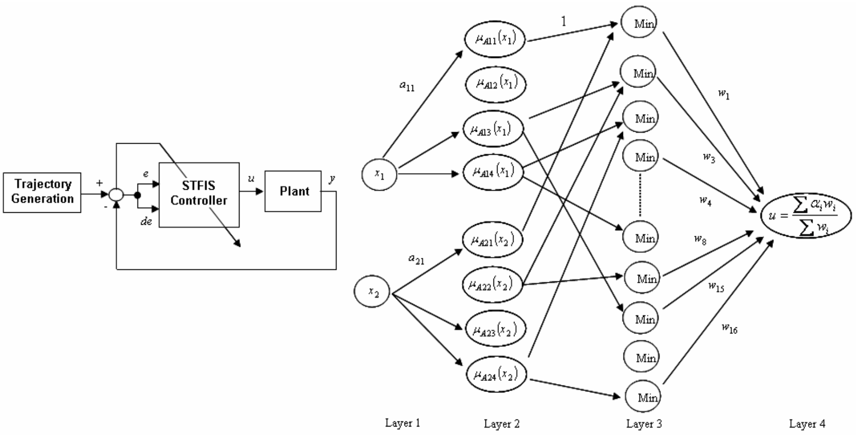


Fig. 2. Control architecture mini-JEAN (left). Architecture of STFIS controller (right).

Self-tunable fuzzy inference system presentation: A Sugeno type fuzzy system is determined in three stages:

1. Given an input x a membership degree μ is obtained from the antecedent part of rules.
2. A truth value degree α_i is obtained, associated to the premise of each rule R_i : IF x_1 is X_1 AND IF x_2 is X_2 THEN u IS w_i .
3. An aggregation stage to take into account all rules by $y = \sum_{i=1}^{r} \alpha_i w_i \Big/ \sum_{i=1}^{r} w_i$ permits to obtain a crisp value y .

4 Motion Planning and Simulation Results

The XSF is tested in simulation in order to validate some motion planning algorithms considering the proposed STFIS control. The objectives are to test the capability of the engine to fly with rounded intersections and crossroads. The two internal degree of freedom lead to a longitudinal/lateral forces which permit to steer the system with rounded corner flying road. The proposed control inputs permit to perform the tracking objectives; flying road with straight and round corners like connection.

Table 1. Weight for z displacement

de/e	*NB*	*NS*	*Z*	*PS*	*PB*
NB	29,62	29,71	7,59	2,55	2,99
NS	29,52	31,34	10,63	3,02	1,69
Z	34,22	29,73	19,54	4,84	2,79
PS	39,72	41,37	22,32	1,74	9,61
PB	39,92	39,82	28,27	7,96	9,62

Starting with a preinitialized rules table, when XSF begins to fly, it performs the acquisition of the distances (observations), calculates the cost function to back-propagation, updates the triggered rules in real time, begins to move and so on. The weights w_i are then adjusted locally and progressively.

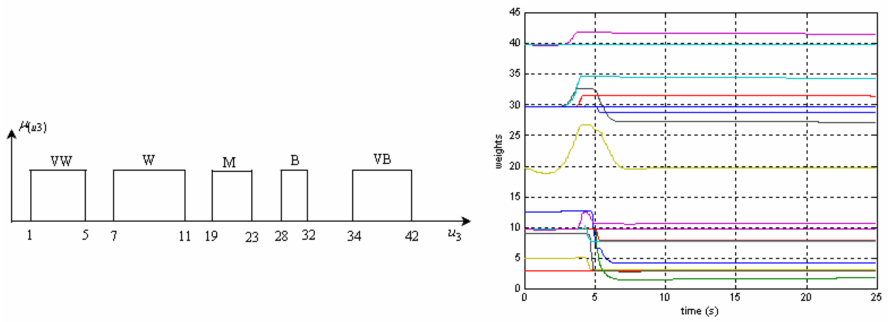


Fig. 3. Linguistic translation representation of the controller u_3 (left). The weights evolutions (right).

The universes of discourse are normalized and shared in five fuzzy subsets for all displacements. The linguistic labels are defined as NB : Negative Big, NS : Negative Small, Z : Approximately Zero, PZ : Positive Small and PB : Positive Big. The results of the simulation are reported in the Table 1 for the z displacement. The optimization phase tends to stable weights (Figure 3 (right)). In these circumstances the outputs linguistic labels could be interpreteted as follows (Figure 3 (left)) VW: {1, 5} Very Weak, W: {7, 11} Weak, M: {19, 23} Medium, B: {28, 32} Big and VB: {34, 42} Very Big.

Table 2. Linguistic table for z displacement by learning (left) and by expertise (right)

de/e	*NB*	*NS*	*Z*	*PS*	*PB*	de/e	*NB*	*NS*	*Z*	*PS*	*PB*
NB	B	B	W	VW	VW	*NB*	B	B	W	VW	VW
NS	B	B	W	VW	VW	*NS*	B	B	W	VW	VW
Z	VB	B	M	W	VW	*Z*	VB	B	M	W	VW
PS	VB	VB	M*	VW*	W	*PS*	VB	VB	M*	VW*	W
PB	VB	VB	B	W	W	*PB*	VB	VB	B	W	W

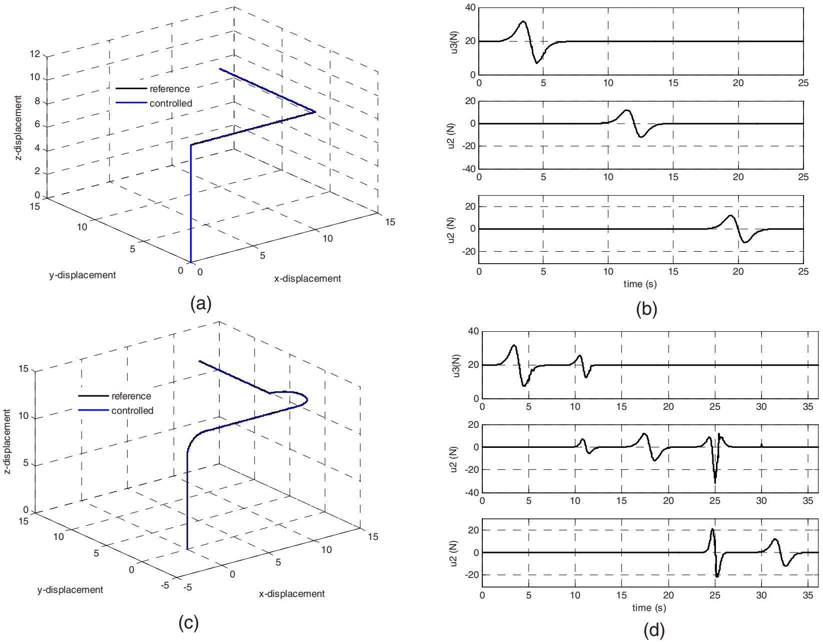


Fig. 4. Inputs u_3 and u_2 (b) for the realization of a straight corners (a). Inputs u_3 and u_2 (d) for the realization of a round corners (c).

The Table 2 (left), illustrates the linguistic translation of the table obtained by on-line optimization for the z displacement (Table 1). By comparing the table proposed by human expertise, Table 2 (right) and Table 2 (left), we can observe that the two sets of linguistic rules are quite close. Two cases noted with (*) are different and they differ from only one linguistic concept (M instead B and VW instead W). So, we can claim that the extracted rules are quite logical and coherent.

On the other hand, the main advantage of the described technique is the optimization of the controller with respect to the actual characteristics of the engine.

The use of a cost function gathering a quadratic error and a term of regression of the weights enabled to achieve our goal. For this behavior, the building of the navigation controller is done entirely on-line by the optimization of a zero order Takagi-Sugeno fuzzy inference system by a back-propagation-like algorithm.

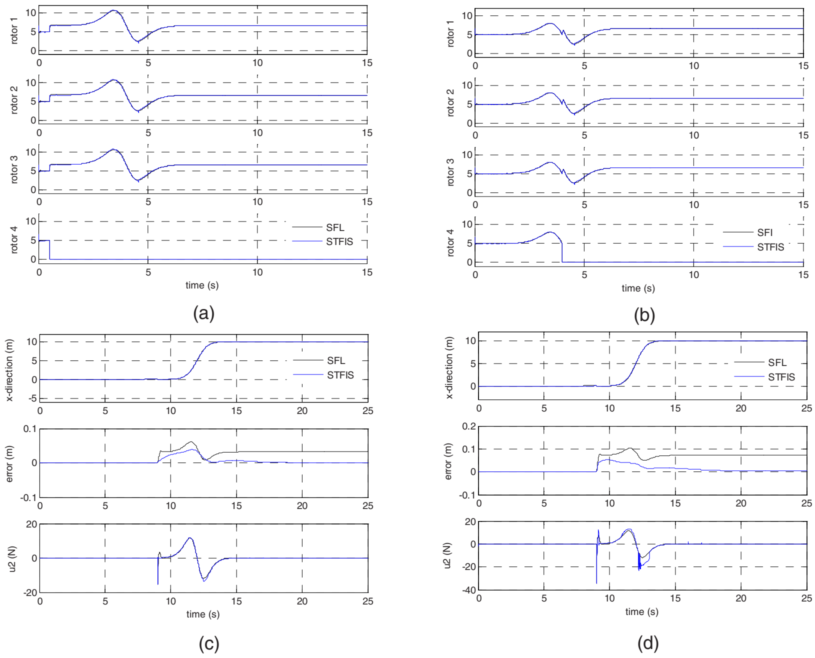
The Figure 4 (b) and (d) illustrate the controlled positions xyz using STFIS controller where u_3 and u_2 denote the command signals for z and for x and y directions respectively. Note that the input $u_3 = mg$ at the equilibrium state is always verified. The input u_2 tend to zero after having carried out the desired orientation of the vehicle. The 3D displacement is depicted with straight and round corners like connection on the Figure 4 (a) and (c). These figures show the effectiveness of the used controller.

5 Controllers Robustness

The robustness of the developed controllers are evaluated regarding external disturbances and performance degradations in the case of actuator/sensor failures and wind influence. In the case of the XSF, a resistance or a drag force is opposed to its movement in flight. The work produced by this force involves an additional energy consumption at the actuators levels which limits its maneuvering capacities in flight. This force can be expressed as follow:

$$F_i = \frac{1}{2} C_x \rho A V_i^2 \tag{8}$$

where $F_i\left[N\right]$ is the drag force following the i axis, $V_i\left[m/s\right]$ is the drone velocity, $A\left[m^2\right]$ is the coss-sectional area perpendicular to the force flow and $\rho\left[Kg/m^3\right]$ is the body density. The equation (8) induced a drag coefficient C_x which is a dimensionless quantity that describes a characteristic amount of aerodynamic drag depending on the XSF structure and which is experimentally determined by windtunnel tests. This coefficient is equal to 0.5 for the x and y directions and 0.08 for the z displacement. The surface characteristic A of the XSF drone is equal to $A = 0.031m^2$ and its density is considered equal to $\rho = 1.22\,Kg/m^3$.



(a)

(b)

(c)

(d)

Fig. 5. XSF Forces in the case of motor 4 failures (a) at $t = 0.5\,\text{sec}$, (b) at $t = 4\,\text{sec}$. (c) Wind influence with a drag force of $1.4N$ and (d) $2.1N$ for the x direction.

The Figure 5 (a) and (b) illustrate the simulation results in the case of actuator 4 failure after take-off at the instant $t_1 = 0.5\,\text{sec}$ and $t_2 = 4\,\text{sec}$ in the z direction. To maintain its equilibrium, the three remain actuators share the drone load compensation where the practically results are in an equitable distribution of the developed forces ($F_1 + F_2 + F_3 = mg$ at steady state). The STFIS and the SFL controllers behave in the same way.

The Figure 5 (c) and (d) present the simulation results in the case of a drag force of $F_{dg} = 1.4N$ and of $F_{dg} = 2.1N$ according to the x displacement. The STFIS controller exhibits chattering signal problems in the transition phase while the SFL controller presents static errors that varies proportionally with the drag force amplitude F_{dg} . The same observations are found according to the two directions y and z .

6 Conclusion

In this paper, we studied a new configuration of flyer engine called XSF. We have considered in this work, the stabilizing/tracking control problem for the three decoupled displacements. The objectives are to test the capability of the engine to fly with straight and rounded intersections. We have presented and implemented an optimization technique allowing an on-line adjustment of the fuzzy controller parameters. The descent gradient algorithm, with its capacities to adapt to unknown situations by the means of its faculties of optimization, and the fuzzy logic, with its capacities of empirical knowledge modelling, are combined to control a new bidirectional drone. Indeed, we have obtained an on-line optimized Takagi-Sugeno type SIF of zero order. A comparison between the STFIS set rules and that deduced by human expertise, shows the validity of the proposed technique. An analysis of the STFIS and the SFL controllers and their robustness regarding disturbances, shows the advantages and the disadvantages of these two techniques.

Future works will essentially investigate the real time implementation of the STFIS and the based-model control techniques. Obstacles avoidance and flying multi-drones are also envisaged thanks to the SIF faculties and its optimization capabilities.

References

1. Altug E (2003) Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, quadrotor. Ph.D. thesis, Faculties of the University of Pennsylvania
2. Azouz N, Bestaoui Y (2005) Modelling and simulation of a mini quad-rotor helicopter. Proceeding of DTM2005, ASME Design Theory and Methodology Coferences, Long Beach, California, USA
3. Beji L, Abichou A, Zemalache K M (2005) Smooth control of an X4 bidirectional rotors flying robots. Fifth International Workshop on Robot Motion and Control, Dymaczewo, Poland

4. Castillo P, Dzul A, Lozano R (2004) Real-time stabilization and tracking of a four rotor mini-rotorcraft. IEEE Transactions on Control Systems Technology, Vol. 12, N 4, pp 510-516
5. Hamel T, Mahony R, Lozano R, Ostrowski J P (2002) Dynamic modelling and configuration stabilization for an X4-flyer. IFAC 15th World Congress on Automatic Control, Barcelona, Spain
6. Jang J S R (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions System, Man and Cybernet.23, pp 665-685
7. Maaref H, Barret C (2001) Progressive optimization of a fuzzy inference system. IFSA-NAFIPS'2001, Vancouver, pp 665-679
8. Tayebi A, McGilvray S (2004) Attitude stabilization of a four-rotor aerial robot. 43rd IEEE Conference on Decision and Control, Atlantis, Bahamas
9. Zemalache K M, Beji L, Maaref H (2005) Control of an under-actuated system: application to a four rotors rotorcraft. 5th IEEE International Conference on Robotics and Biomimetics, Hong Kong, China, Vol. 1, pp 160-230
10. Zemalache K M, Beji L, Maaref H (2006) Neuro-Fuzzy system controlling a X4-Flyer. 5th International Symposium on Robotics and Automation 2006, San Miguel Regla Hidalgo, México, pp 460-465
11. Zemalache K M, Beji L, Maaref H (2006) Motion planning and stabilisation of a X4 Bidirectional rotors using a Neuro-Fuzzy controller. 9th International Symposium on Climbing and Walking Robots and Associated Technologies, Royal Military Academy, Brussels, Belgium, pp 678-683

Optimal Path Planning for Autonomous Mobile Robot Navigation Using Ant Colony Optimization and a Fuzzy Cost Function Evaluation

M.A. Porta García[1], Oscar Montiel[1], Oscar Castillo[2], and Roberto Sepúlveda[1]

[1] Centro de Investigación y Desarrollo de Tecnología Digital CITEDI-IPN
maporta@gmail.com, o.montiel@ieee.org, rsepulveda@citedi.mx
[2] Department of Computer Science, Tijuana Institute of Technology
P.O. Box 4207, Chula Vista CA 91909, USA
ocastillo@galgo.tectijuana.mx

Abstract. In this work, a method for finding the optimal path from an initial point to a final one in a previously defined static search map is presented, based on Ant Colony Optimization Meta-Heuristic (ACO-MH). The proposed algorithm supports the avoidance of dynamic obstacles; that is, once the optimal path is found and the robot starts navigating, if the robot's route is interrupted by a new obstacle that was sensed at time *t*, it will recalculate an alternative optimal path from the actual robot position in order to surround this blocking object and reach the goal.

Keywords: Ant Colony Optimization, Autonomous Mobile Robot Navigation, Fuzzy Logic, Path Planning.

1 Introduction

Nowadays, robotics is an essential part in manufacturing processes automatization. Concerning about mobile robots, autonomous navigation entails a great challenge. Mobile robots can be very useful in different situations where humans could be in danger or when they are not able to reach certain target because of terrain conditions. Then, the path planning is an interesting and challenge subject for research, and it has many different approaches. This paper proposes an ACO-MH based method for path planning founding the best route according to certain cost function.

The ACO-MH is inspired in the foraging behavior of real ants for finding the optimal path from the nest to where the food is. For communication method, some ant species use stigmergy, a term introduced by French biologist Pierre-Paul Grassé in 1959. With stigmergy, each ant communicates with one another by modifying their local environment. The ants achieve this task by laying down pheromones along their trails [2]. ACO-MH solves mainly combinatorial optimization problems defined over discrete search spaces. The ant-based algorithms developed as a result of studies of ant colonies are referred as instances of ACO-MH [4]. As a first stage in this research work, the proposed method is based on the ACO-MH instance mainly: Simple Ant Colony Optimization (SACO).

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 790–798, 2007.
springerlink.com

There are some similar works like [6], where the robot has to visit multiple targets, like the traveling salesman problem but with the presence of obstacles. The robot in this case is modeled as a point robot; that is, the robot occupies an exact cell in the discrete representation of the workspace. Using several robots as ants, this robot team architecture has to be in constant communication with each other at all times to share pheromone information. There are some other approaches for similar tasks, like [10], where a new meta-heuristic method of ACO is proposed to solve the vehicle routing problem, using a multiple ant colony technique where each colony works separately. On the other hand, there are many path planning approaches by other soft computing techniques, such like Genetic Algorithms [1,5,8]. Then, as it can be observed in actual reports [7,9], the focus on ant algorithms is growing becoming an interesting alternative of solution for path planning.

2 Simple Ant Colony Optimization Algorithm

The SACO is an algorithmic implementation that adapts the behavior of real ants to solution of minimum cost path problems on graphs. A number of artificial ants build solutions for the optimization problem by issue and exchange information about the quality of these solutions making allusion to the communication system of the real ants [3].

Let be the graph G = (V,E), where V is the set of nodes and E is the matrix of links between nodes. G has $n_G = |V|$ nodes. Let be L^k the number of hops in the path built by ant *k* from the origin node to the destiny node. Therefore, it needs to be found:

$$Q = \{q_a, \ldots, q_f \mid q_i \in C\} \tag{1}$$

Where Q is the set of nodes representing a path at least continuous with no obstacles; $q_a, \ldots, q_f$ are former nodes of the path and C is the set of possible configurations of the free space. If $x^k(t)$ denotes a Q solution in time t, $f(x^k(t))$ expresses the quality of the solution. In general terms, the steps of SACO are as follows:

- Each link (i,j) is associated with a pheromone concentration denoted as τ_{ij}
- A number $k = 1, \ldots, n_k$ are placed in the origin node (the nest).
- On each iteration or epoch all ants build a path to the destiny node (the food source). For the next node selection it is used the probabilistic formula:

$$p_{ij}^k(t) \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)}{\sum_{j \in N_i^k} \tau_{ij}^{\alpha}(t)} & if \quad j \in N_i^k \\ 0 & if \quad j \notin N_i^k \end{cases} \tag{2}$$

In Eq. 2, N_i^k is the set of feasible nodes connected to node *i* with respect to ant *k*; τ_{ij} is the total pheromone concentration of link *ij*, where α is a positive constant used as gain for the pheromone concentration influence.

- Remove cycles and compute each route weight $f(x^k(t))$. A cycle could be generated when there are no feasible candidates nodes, that is, for any node *i* and ant *k*, N_i^k = Ø; then predecessor of that node *i* is included as a former node of the path.

- Compute pheromone evaporation using the Eq. 3.

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) \tag{3}$$

In Eq. 3, ρ is the evaporation rate value of the pheromone trail. The evaporation is added to the algorithm in order to force the exploration of the ants, and avoid premature convergence to sub-optimal solutions. For ρ = 1, the search is completely random. While an ant takes more time for crossing a path, there is more time for the pheromone trail to evaporate. On a short path, which is crossed quickly, the density of the pheromone is higher. Evaporation avoids convergence to local optimums. Without evaporation, the paths generated by the first ants would be excessively attractive for the subsequent ones. In this way, exploration of the search space is not too restricted.

- Update pheromone concentration by using Eq. 4.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t) \tag{4}$$

- The algorithm can be ended in three different ways:
- – When a maximum number of epochs has been reached.
- – When it has been found an acceptable solution, with $f(x^k(t)) < \varepsilon$.
- – When all ants follow the same path.

3 The ACO-MH Based Proposed Method

3.1 The Search Area Design

This approach makes some improvements over the SACO algorithm and adaptations for the mobile robot routing problem of this research work. The map where the mobile robot navigates is a search space discretized into a matrix representing a graph of 50x50 nodes, where 0 means a feasible node (plain terrain) and 1 are obstacles. It is remarkable to say that each artificial ant of the algorithm is a scale representation of the real mobile robot, which means the proposed method considers robot's dimensions; for example, there are going to be situations during the optimization process, where some paths are rejected if the robot doesn't fit in the space between two obstacles. In this case the Boe-Bot® of Parallax is used for the model. The 50x50 map represents a 4m^2 area.

For this method, is assumed all nodes are interconnected. In a map with no obstacles, there are 2500 feasible nodes; therefore the matrix of links *E* would be extremely large. For this reasons *E* is not used, and the pheromone amount value is assigned at each node, which reduces considerably the complexity of the algorithm and then the processing time. This is equivalent to assign the same pheromone concentration to the eight links around every node. If an analogy with reality is made, this can be seen as ants leaving food traces in each node they are visiting, instead of a pheromone trail on the links.

3.2 The Node Selection Process

Once the ants are placed in the origin node, each ant starts navigating, and the decision process for choosing the next node consist in a 3x3 window of the whole graph. The ant can choose one of the 8 nodes around it, and the transition probability (2) is now:

$$p_{ij}^{k}(t)\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)}{\sum_{j\in N_i^k}\tau_{ij}^{\alpha}(t)*\xi^{\beta}} & if \quad j\in N_i^k\\ 0 & if \quad j\notin N_i^k\end{cases} \tag{5}$$

Where ξ represents the Euclidean distance between the candidate node and the destiny node, $\beta\in[0,\infty)$ amplifies the influence of ξ, and a memory capability γ is added to ants. The value γ represents how many nodes can remember the ant. After γ iterations of the building routes process, this memory is “erased”, and the count starts again. Further analysis of these new parameters is made in the Simulation Results section of this paper.

3.3 The Fuzzy Cost Function

The cost of the path $f(x^k(t))$ to determine the optimal one is evaluated by a Fuzzy Inference System (FIS), which contemplates not only the length of the path but the difficulty for the navigation. The FIS considers two inputs: *effort* and *distance*. The first one represents the energy spent by the robot to make turns across the path; for example, the effort become increased if the robot has to make a left turn after a long straight line, because it has to decelerate more. *Distance* is the accumulated Euclidean distance at the moment between the visited nodes. The output is a weight assigned to the cost of the path; the more weight is given, the less desirable becomes the path. The output of the FIS is added to the total Euclidean distance of the path, giving the

Table 1. FIS variables

Input variables		Output variable
Effort	Distance	Weight
NE: Normal Effort BE: Big Effort VBE: Very Big Effort	VSD: Very Small Distance SD: Small Distance D: Distance BD: Big Distance VBD: Very Big Distance	MW:Minimum Weight SW: Small Weight W: Weight BW: Big Weight VBW: Very Big Weight

Table 2. Fuzzy associative memory of the FIS

		Distance				
		VSD	SD	D	BD	VBD
Effort	NE	MW	SW	W	BW	VBW
	BE	W	W	BW	VBW	VBW
	VBE	W	W	BW	VBW	VBW

final weight of each one generated by ants. In case there are different routes with the same length, the FIS should make a difference of cost giving preference to the straighter paths. The FIS variables can be seen in Table 1 and the associative memory of the FIS is shown in Table 2.

3.4 Dynamic Obstacles Generation

The algorithm has the capability of sensing changes in the environment, if a new obstacle is placed over the robot's route at time *t*, it starts a rerouting process in order to avoid the blocking object and get to the destiny node. It has to be considered that after some epochs, the pheromone concentration τ_{ij} is already increased over the visited nodes; then, when a new obstruction appears, it causes evaporation of the pheromone trail around it. This premise prevents stagnation around the obstacle, and τ_{ij} of the surrounding area is given by the minimum pheromone value over the search map at *t*.

4 Simulation Results

The experiments results reported in this paper have the same parameter adjustment for $k = 3$, $\tau_{ij} = 0.5$, $\rho = 0.2$ and $\alpha = 2$. The first considered scenario is the one with no obstacles. Figure 1(a) shows the path generated by the first ant in the first iteration of the algorithm, with β=0 and γ=1. As it can be seen, unnecessary and excessively exploration of the map is made by the ant, since there is no obstacles in the whole area. Figure 1(b) displays how a bigger value of γ reduces considerably the excessive exploration, by adding the memory capability to the ant, but without the help of γ, in Figures 1(c) and 1(d) it can be observed how increasing the value of β is enough to make the algorithm more efficient. For β=1, the optimal path -the diagonal between the nodes- can be found in less than three epochs, and it takes approximately 1 second to get the optimal solution, which is shown in Figure 2(a).

At the moment, an analysis about β and γ has been made. Figure 3(a) presents a more complex map and the optimal path given by the algorithm, where the algorithm

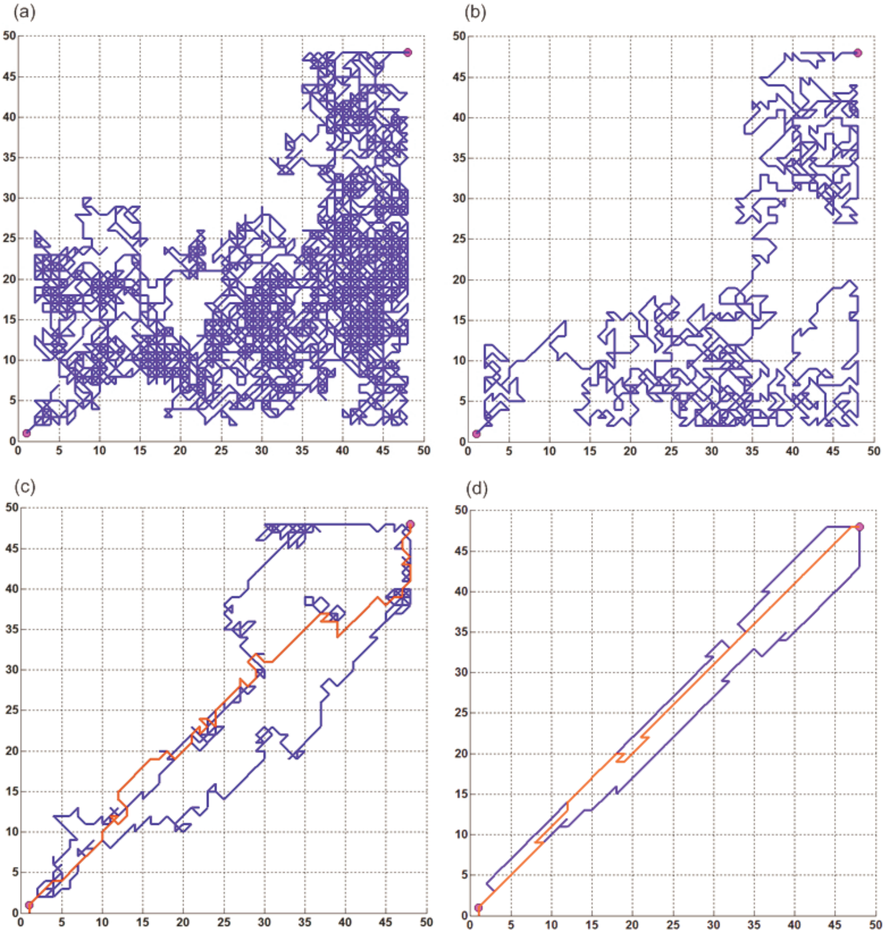


Fig. 1. (a) Route generated by the first ant in the first epoch, with β = 0 and γ = 1, (b) same situation but with β = 0 and = 100, (c) The routes of three ants in the first epoch with β = 0.1 and γ = 1, (d) same situation but with β = 0.5 and γ = 1

is more sensitive to variations of the parameters. Changes in the values of α, ρ and *k* mostly, influence the execution time and the effectiveness of the algorithm considerably.

Figure 2(b) shows the same first scenario, but now it has been added 3 new obstacles dynamically in different times *t*, *t+1* and *t+2*. The ants were able to surround the blocking object and finally reach the goal.

In Figure 3(a), with *k* =3, β =0 and γ =1, adjusting α =0.5 and ρ =0.2, the algorithm takes approximately 180 seconds and 130 epochs to find the optimal path. This causes more exploration but less exploitation. Making α =2 and ρ = 0.35, it takes around 75 seconds to find the optimal path in 20 epochs.

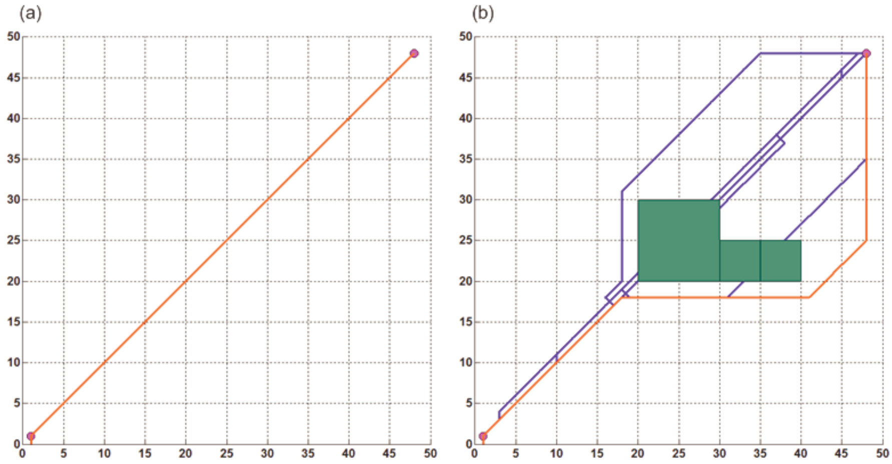


Fig. 2. (a) Optimal path found by the algorithm after one or two epochs with β = 1 and γ = 1, (b) generated routes after addition of obstacles dynamically at time *t*, *t+1* and *t+2*

Figure 3(b) shows a frame from the Boe-bot® model animation navigating over the path, ensuring the mobile robot dimensions fits in the free space area crossed by the route. Another example of dynamic search map can be observed in Figure 4(a), where the robot model is crossing over the alternative route given by the algorithm.

Figure 4(b) displays a completely different map, with major randomness in the obstacles positioning. At first sight it can be noticed the path is not the optimal one. There are some turns that are not desirable, like the one around the circle.

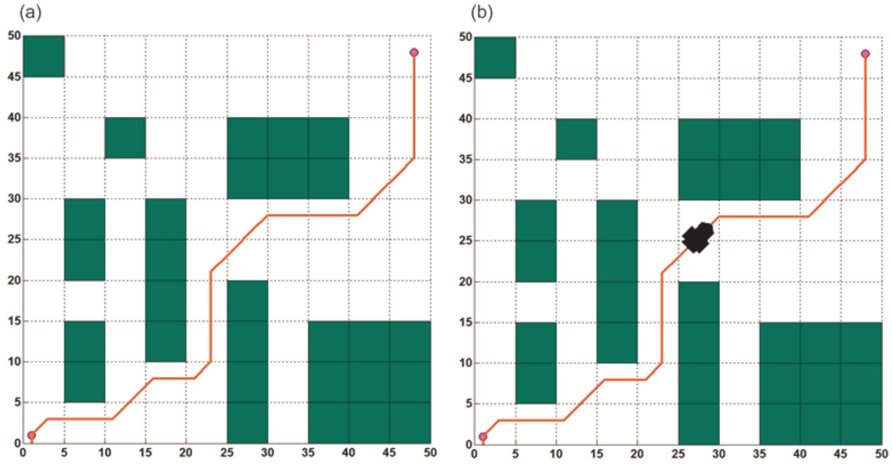


Fig. 3. (a) Example of the optimal path found by the algorithm, (b) A frame of the Boe-bot® model animation following the final route

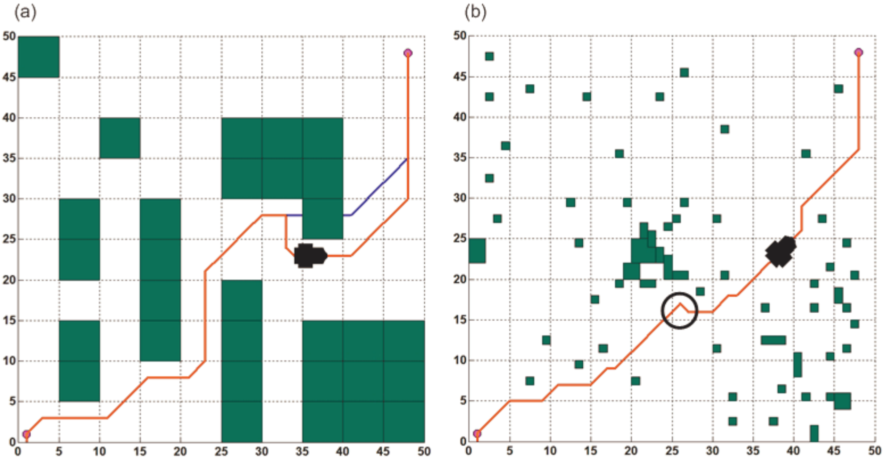


Fig. 4. (a) Alternative route given by the algorithm after a new obstacle appears at time *t* and the Boe-bot® model following the modified path, (b) Another search area, with major randomness in the obstacles positioning. The circle indicates a not desirable movement in the path

This kind of issues are intended to be corrected by the fuzzy cost function, but the method is not fully tested yet and every different scenario has its own particular features; they may need different parameters settings in order to get optimal solutions at a reasonable time. However, these are satisfactory results that allow seeing the proposed method as an effective path planning tool and it is getting improvements.

5 Conclusions

The ACO-MH proposed method seems to be a promising path planning system for autonomous mobile robot navigation since the given solutions are not only paths, but the optimal ones. It is remarkable to mention the reduced time of execution compared to other soft computing techniques. Nevertheless, as seen in the previous section, there are several details to consider for making a more robust and efficient algorithm. For future work, another ACO-MH is going to be used, such as MMAS (Max-Min Ant System) or ANTS (Approximated Non-deterministic Tree Search); the fuzzy cost function and the use of β and γ is still under research. At the moment, the method is been used with the real robot.

References

1. Chen H, Xu Z (2005) Path Planning Based on a New Genetic Algorithm. IEEE.
2. Dorigo M, Birattari M, Stützle T (2006) Ant Colony Optimization. IEEE Computational Intelligence Magazine, pp 28-39.
3. Dorigo M, Stützle T (2004) Ant Colony Optimization. Bradford, Cambridge, Massachusetts.

4. Engelbrecht AP (2005) Fundamentals of Computational Swarm Intelligence. Wiley, England.
5. Gemeinder M, Gerke M (2002) An Active Search Algorithm Extending GA Based Path Planning for Mobile Robot Systems. Soft Computing and industry, Springer, pp 589-596.
6. Gopalakrishnan K, Ramakrishnan S (2006) Optimal Path Planning of Mobile Robot With Multiple Targets Using Ant Colony Optimization. Smart Systems Engineering, New York, pp 25-30.
7. Mohamad M, Dunningan W (2005) Ant Colony Robot Motion Planning. IEEE, pp 213-216.
8. M. Tarokh (2000) Path planning of rovers using fuzzy logic and genetic algorithm, World Automation Conf. ISORA-026, pp. 1-7, Hawaii.
9. Ye W, Ma D, Fan H (2006) Path Planning for Space Robot Based on The Self-adaptive Ant Colony Algorithm. IEEE.
10. Zhishuo L, Yueting C (2005) Sweep based Multiple Ant Colonies Algorithm for Capacitated Vehicle Routing Problem. IEEE International Conference on e-Business Engineering.

Intelligent Control and Planning of Autonomous Mobile Robots Using Fuzzy Logic and Multiple Objective Genetic Algorithms

Oscar Castillo, Jose Soria, Hector Arias, Jose Blas Morales, and Mirsa Inzunza

Division of Graduate Studies, Tijuana Institute of Technology, Mexico

Abstract. This paper describes the use of a Genetic Algorithm (GA) for the problem of Offline Point-to-Point Autonomous Mobile Robot Path Planning. The problem consist of generating "valid" paths or trajectories, for an Holonomic Robot to use to move from a starting position to a destination across a flat map of a terrain, represented by a two dimensional grid, with obstacles and dangerous ground that the Robot must evade. This means that the GA optimizes possible paths based on two criteria: length and difficulty**.**

1 Introduction

The problem of Mobile Robot Path Planning is one that has intrigued and has received much attention thru out the history of Robotics, since it's at the essence of what a mobile robot needs to be considered truly "autonomous". A Mobile Robot must be able to generate collision free paths to move from one location to another, and in order to truly show a level of intelligence these paths must be optimized under some criteria most important to the robot, the terrain and the problem given. GA's and evolutionary methods have extensively been used to solve the path planning problem, such as in (Xiao and Michalewicz, 2000) where a *CoEvolutionary* method is used to solve the path planning problem for two articulated robot arms, and in (Ajmal Deen Ali et. al., 2002) where they use a GA to solve the path planning problem in non-structured terrains for the particular application of planet exploration. In (Farritor and Dubowsky, 2002) an *Evolutionary Algorithm* is used for both off-line and on-line path planning using a linked list representation of paths, and (Sauter et. al., 2002) uses a *Particle swarm optimization* (PSO) method based on *Ant Colony Optimization* (ACO). However, the research work presented in this paper used as a basis for comparison and development the work done in (Sugihara, 1999). In this work, a grid representation of the terrain is used and different values are assigned to the cells in a grid, to represent different levels of difficulty that a robot would have to traverse a particular cell. Also they present a codification of all monotone paths for the solution of the path-planning problem.

2 Basic Theory

This section is intended to present some basic theory used to develop the GA's in this paper for use in the path planning problem, covering topics like basic Genetic

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 799–807, 2007.
springerlink.com

Algorithm theory, Multi Objective optimization, Triggered Hypermutation and Autonomous Mobile Robot Point-to Point Path Planning.

2.1 Genetic Algorithms

A Genetic Algorithm is an evolutionary optimization method used to solve, in theory "any" possible optimization problem. A GA (Man et. al., 1999) is based on the idea that a solution to a particular optimization problem can be viewed as an *individual* and that these individual characteristics can be coded into a finite set of parameters. These parameters are the *genes* or the *genetic information* that makes up the *chromosome* that represents the real world structure of the individual, which in this case is a solution to a particular optimization problem. Because the GA is an evolutionary method, this means that a repetitive loop or a series of *generations* are used in order to evolve a *population S* of *p* individuals to find the *fittest* individual to solve a particular problem. The *fitness* of each *individual* is determined bye a given *fitness function* that evaluates the level of aptitude that a particular *individual* has to solve the given optimization problem. Each *generation* in the genetic search process produces a new set of individuals thru *genetic operations* or *genetic operators: Crossover* and *Mutation,* operations that are governed by the *crossover rate* γ and the *mutation rate* μ respectively. These operators produce new *child chromosomes* with the intention of bettering the overall fitness of the population while maintaining a global search space. Individuals are selected for *genetic operations* using a *Selection method* that is intended to select the fittest individuals for the role of *parent chromosomes* in the *Crossover* and *Mutation* operations. Finally these newly generated child chromosomes are reinserted into the population using a *Replacement method.* This process is repeated a *k* number of *generations.*

2.2 Multiple Objective Genetic Algorithms

Real-world problem solving will commonly involve (Oliveira et. al., 2002) the optimization of two or more objectives at once, a consequence of this is that it's not always possible to reach an optimal solution with respect to all of the objectives evaluated individually. Historically a common method used to solve multi objective problems is by a linear combination of the objectives, in this way creating a single objective function to optimize (Sugihara, 1997) or by converting the objectives into restrictions imposed on the optimization problem. In regards to evolutionary computation, (Shaffer, 1985) proposed the first implementation for a multi objective evolutionary search. The proposed methods in (Fonseca and Fleming, 1993), (Srinivas, 1994) and (Goldberg, 1989), all center around the concept of *Pareto optimality* and the *Pareto optimal set*. Using these concepts of optimality of *individuals* evaluated under a multi objective problem, they each propose a *fitness* assignment to each individual in a current population during an evolutionary search based upon the concepts of *dominance* and *non-dominance* of *Pareto optimality*. Where the definition of *dominance* is stated as follows:

Definition 1. *For an optimization (minimization) problem with n-objectives, solution u is said to be dominated by a solution v if:*

$$\forall i = 1,2,....,n. \qquad f_i(u) \geq f_i(v) \ , \tag{1}$$

$$\exists j = 1,2,....,n, \qquad \because \ f_i(u) > f_i(v) \tag{2}$$

2.3 Triggered Hypermutation

In order to improve on the convergence of a GA, there are several techniques available such as (Man et. al. 1999) expanding the memory of the GA in order to create a repertoire to respond to unexpected changes in the environment. Another technique used to improve the overall speed of convergence for a GA is the use of a Triggered Hypermutation Mechanism (Cobb, 1990), which consists of using *mutation* as a control parameter in order to improve performance in a dynamic environment. The GA is modified by adding a mechanism by which the value of μ is changed as a result of a dip in the fitness produced by the best solution in each generation in the genetic search. This way μ is increased to a high *Hypermutation* value each time the top fitness value of the population at generation *k* dips below some lower limit set beforehand.

2.4 Autonomous Mobile Robots

An Autonomous Mobile Robot as defined in (Xiao and Michalewicz, 2000) can be seen as a vehicle that needs the capability of generating collision free paths that take the robot from a starting position *s* to a final destination *d*, and needs to avoid obstacles present in the environment. The robot must be able to have enough relevant information of his current position relative to *s* and *d,* and of the state of the environment or terrain that surrounds it. One advantage about generating paths or trajectories for these kinds of robots, compared to the more traditional robot arms, is that in general there are far less restrictions in regards to the precision with which the paths must be generated. The basic systems that operate in an Autonomous Mobile robot are:

1) Vehicle Control.
2) Sensor and Vision.
3) Navigation
4) Path Planning

2.5 Point-to-Point Path Planning Problem

The path planning problem when analyzed with the point-to-point technique, (Choset et. al., 1999) comes down to finding a path from one point to another (start and destination). Obviously, one of the most important reasons to generate an appropriate path for a robot to follow, is to help it avoid possible danger or obstacles along the way, for this reason an appropriate representation of the terrain is needed generating a sufficiently complete map of the given surroundings that the robot will encounter along its route. The general path-planning problem, that all autonomous mobile robots will face, has been solved (to some level of satisfaction) with various techniques, besides the evolutionary or genetic search, such as, using the *Voroni Generalized Graph* (Choset et. al., 1999), or using a *Fuzzy Controller* (Kim et. al., 1999*)*, yet another is by the use of *Artificial Potential Fields* (Planas et. al., 2002).

3 Proposed Method

The first step before we can continue and give the details of the GA implementation used to solve the path-planning problem, is to explicitly define the problem and what is it that we are expecting out of the subsequent genetic search. To this end, we propose what will be the *input/output* pair that we are expecting from our GA as follows:

Input: 1) An ***n*** x ***n*** grid, where the starting cell ***s*** for the robot is in one corner and the destination cell ***d*** is diagonally across from it.

2) Each cell with a corresponding *difficulty weight* ***wd*** assigned to it ranging from [0, 1].

Output: A path, defined as a sequence of adjacent cells joining ***s*** and ***d***, and that complies with the following restrictions and optimization criteria:

1) The path most not contain cells with ***wd*** = 0 (solid obstacles).
2) The path must stay inside of the grid boundaries.
3) Minimize the path length (number of cells).
4) Minimize the total difficulty for the path, that means, the combined values of ***wd*** for all the cells in a given path.

We must also establish a set of ground rules or assumptions that our GA will be operating under.

1) The ***n*** x ***n*** grid isn't limited to all cells in the grid having to represent a uniform or constant size in the terrain, each cell is merely a conceptual representation of spaces in a particular terrain.
2) Each cell in a terrain has a given *difficulty weight* ***wd*** between the values of [0,1], that represents the level of difficulty that a robot would have to pass through it, where the lower bounds 0 represents a completely free space and the higher bounds 1 represents a solid impenetrable obstacle.
3) The terrain is considered to be static in nature.
4) It is assumed that there is a sufficiently complete knowledge in regards to the state of the terrain in which the robot will operate.
5) The paths produced by the GA are all monotone paths.

4 Architecture of the Genetic Algorithm

We now turn to the actual implementation of our GA, used to solve the path-planning problem for one and two optimization objectives. So we describe each of the parts of our GA and give a brief description of each, clearly stating any differences between the one and two optimization objectives implementations.

4.1 Individual Representation

Basically, the chromosome structure was taken from the work done in (Sugihara, 1999) where a binary string representation of monotone paths is used. The binary string chromosome is made up of ***n***-1 (where ***n*** is the number of columns and rows in the grid representing the map of a given terrain) pairs of *direction/distance* of length *3*

+ *log[2]**n**,* and an extra bit *a* which determines if the path is *x-monotone* (*a=0*) or *y-monotone* (*a=1*). And each pair of *direction/distance* codes the direction in which a robot moves inside the grid and the number of cells it moves thru in that direction. The coding used greatly facilitates its use in a GA, because of its constant length no special or revamped genetic operators are needed, a problem that would be very cumbersome to solve if using a linked list chromosome representation of the path as done in (Xiao and Michalewicz, 2000).

4.2 Initial Population

The population *S* used in the genetic search is initialized with *p* total individuals. Of the *p* individuals in *S, p-2* of them are generated randomly while the remaining two represent straight line paths from *s* to *d,* one of this paths is *x-monotone* and the other is *y-monotone.*

So we can clearly define the population *S* as being made up by:

$$S = \{ x_0, x_1, x_2 \ldots\ldots\ldots\ldots x_{p-2}, a, b \} \tag{3}$$

Where x_i are randomly generated individuals, and by *a* and *b* that are *x-monotone* and *y-monotone* paths respectively that take a straight-line route from *s* to *d.*

4.3 Path Repair Mechanism

Each path inside of the population *S* is said to be either *valid* or *non-valid*. Where criteria for *non-validity* are:

- Path contains a cell with a solid obstacle (***wd*** = 1).
- Path contains cells out of bounds.
- The paths final cell isn't *d.*

Using this set of three rules to determine the state of validity of a given path for a particular genetic search, we can define a *subpopulation S',* which is made up by entirely *non-valid* paths in *S.*

The Path Repair Mechanism used with the GA is a *Lamarckian* process designed to take *non-valid x',* where *x' ε S',* and determine if they can be salvaged and return to a *valid* state, so as to be productive in the genetic search, because just because a particular path is determined to be *non-valid* this does not preclude it from having possible information coded in its chromosome that could prove to be crucial and effective in the genetic search process, this is way *non-valid* paths are given low fitness values with the penalty scheme used in the fitness evaluation, only after it has been determined that its *non-valid* state cant be reversed.

4.4 Fitness Evaluation

As was mentioned earlier, we introduce here both single and two objective optimization of the path planning problem, taking into account the length a given path and the difficulty of the same as the two criteria for optimization for paths in the population hence, the way in which each implementation of the GA assigns fitness values differs for obvious reasons.

4.4.1 Single Objective

Considering our Conventional GA, we can say that for paths inside *S* we optimize for only one objective, which is the path length, therefore we define fitness $f_1(x)$ as given by:

$$f_1(x) = (n^2) - (c) \tag{4}$$

Where *c* is the number of cells in a given path *x*.

4.4.2 Multiple Objective

Besides the fitness $f_1(x)$ used in Section 4.4.1 given for path length, a second fitness assignment $f_2(x)$ is given for path difficulty is given, and is calculated by,

$$f_2(x) = (n^2) - \Sigma wd_i \tag{5}$$

Where the second term in (5) is the sum of ***wd*** for each cell in a given path *x*. With this we are forced to use Pareto optimality for a rank-based system for individuals in population *S*. So for a path *x* where *x* ε *S* its final fitness values is given by their *rank* value inside of *S* determined by,

$$\text{rank}(x) = p - t \tag{6}$$

Where *p* is the size of population *S* and *t* is the number of individuals that dominate *x* in *S*.

5 Simulation Results

We use the benchmark test presented in Figure 1, which was used in (Sugihara, 1997) due to its capability of projecting an accurate general performance score for the GA, and the performance measure of *probability optimality* $L_{opt}(k)$, which is a representation of the probability that a GA has of finding an optimal solution to a given

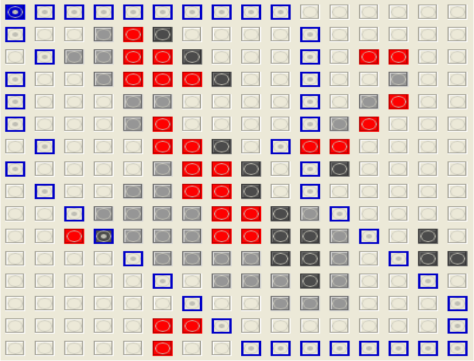


Fig. 1. Benchmark Test, with two paths on the Pareto Optimal Front

problem. In this case, is the probability of finding a solution on the Pareto optimal front. Using $L_{opt}(k)$ as the performance measure we present a set of optimal operating parameters for our MOGA using both a Generational and Elitist replacement scheme, Figures 2 to 3 show the simulation results that support this values. We also compare the two methods along with the GA proposed in (Sugihara, 1999) and the comparison is made under a normalized value for *kp=30,000* keeping the overall computational cost equal for each GA.

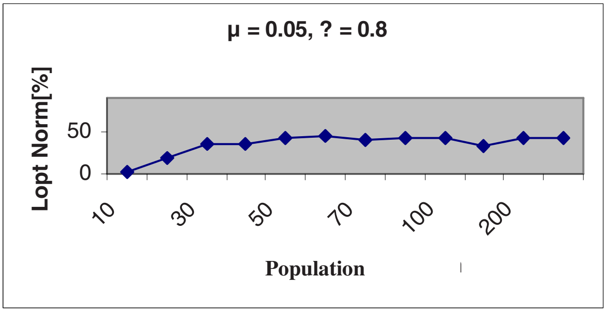


Fig. 2. Normalized $L_{opt}(k)$ and population size with Generational Replacement

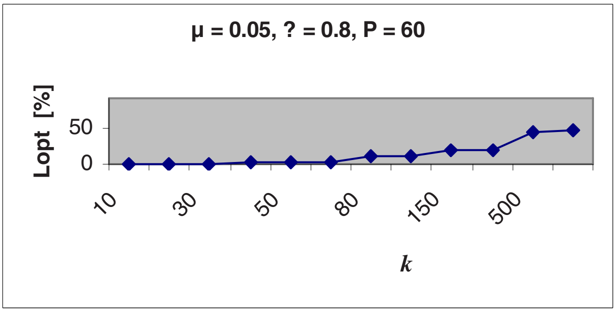


Fig. 3. $L_{opt}(k)$ and number of generations with Generational Replacement

6 Conclusions

This paper presented a GA designed to solve the Mobile Robot Path Planning Problem. We showed with simulation results that both a Conventional GA and a MOGA, based on Pareto optimality, equipped with a basic repair mechanism for *non-valid* paths, can solve the point-to-point path planning problem when applied to grid representations of binary and continuous simulation of terrains respectively. From the simulation results gathered from experimental testing the Conventional GA with a Generational Replacement scheme and Triggered Hypermutation (which is commonly referred to as a conversion mechanism for dynamic environments) gave consistent performance to varying degrees of granularity in the representation of terrains with out a significant increase in population size or number of generations needed in order to complete the search in a satisfactory manner, while the MOGA based on Pareto Optimality combined with a Elitist replacement scheme clearly improves upon previous (Sugihara, 1999) work done with multiple objective path planning problem based on linear combination, with the added advantage of providing more than one equally usable solution.

References

Xiao, J. and Michalewicz, Z. (2000), An Evolutionary Computation Approach to Robot Planning and Navigation, in Hirota, K. and Fukuda, T. (eds.), *Soft Computing in Mechatronics*, Springer-Verlag, Heidelberg, Germany, 117 – 128.

Ajmal Deen Ali, M. S., Babu, N. and Varghese, K. (2002), Offline Path Planning of cooperative manipulators using Co-Evolutionary Genetic Algorithm, *Proceedings of the International Symposium on Automation and Robotics in Construction*, 19th (ISARC), 415-124.

Farritor, S. and Dubowsky, S. (2002), A Genetic Planning Method and its Application to Planetary Exploration, *ASME Journal of Dynamic Systems, Measurement and Control*, **124**(4), 698-701.

Sauter, J. A., Matthews, R., Parunak, H. V. D. and Brueckner, S. (2002), Evolving Adaptive Pheromone Path Planning Mechanisms, *First International Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, 434-440.

Sugihara, K. (1999), Genetic Algorithms for Adaptive Planning of Path and Trajectory of a Mobile Robot in 2D Terrains, *IEICE Trans. Inf. & Syst.*, Vol. E82-D, 309 – 313.

Sugihara, K. (1997), *A Case Study on Tuning of Genetic Algorithms by Using Performance Evaluation Based on Experimental Design*, Tech. Rep. ICS-TR-97-01, Dept. of Information and Computer Sciences, Univ. of Hawaii at Manoa.

Sugihara, K. (1997), Measures for Performance Evaluation of Genetic Algorithms, *Proc. 3rd. joint Conference on Information Sciences*, Research Triangle Park, NC, vol. I, 172-175.

Cobb, H. G. (1990), *An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments*, Technical Report AIC-90-001, Naval Research Laboratory, Washington, D. C..

Fonseca, C. M. and Fleming, C. J. (1993), Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, *5th Int. Conf. Genetic Algorithms*, 416-423.

Srinivas, M. and Deb, K. (1994), Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, **2**(3), 221-248.

Man, K. F., Tang, K. S. and Kwong, S. (1999), *Genetic Algorithms*, Ed. Springer, 1st Edition, London, UK.

Oliveira, G. M. B., Bortot, J. C. and De Oliveira, P. P. B. (2002), Multiobjective Evolutionary Search for One-Dimensional Cellular Automata in the Density Classification Task, in Proceedings of *Artificial Life VIII*, MIT Press, 202-206.

Schaffer, J. D. (1985), Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, 93-100.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Spandl, H. (1992), *Lernverfahren zur Unterstützung der Routenplanung fur eine Mobilen Roboter*, Diss. Universität Karlsruhe, auch VDI-Forsch-Heft Reihe 10 Nr. 212, Düsseldorf, Germany, VDI-Verlag.

Choset, H., La Civita, M. L. and Park, J. C. (1999), Path Planning between Two Points for a Robot Experiencing Localization Error in Known and Unknown Environments, *Proceedings of the Conference on Field and Service Robotics (FSR'99)*, Pittsburgh, PA.

Kim, B. N., Kwon, O. S., Kim, K. J., Lee, E. H. and Hong, S. H. (1999), "A Study on Path Planning for Mobile Robot Based on Fuzzy Logic Controller", *Proceedings of IEEE TENCON'99*, 1-6.

Planas, R. M., Fuertes, J. M. and Martinez, A. B. (2002), Qualitative Approach for Mobile Robot Path Planning based on Potential Field Methods, *Sixteenth International Workshop on Qualitative Reasoning (QR'02)*, 1 -7.

Part XII

Fuzzy Logic Applications

Generalized Reinforcement Learning Fuzzy Control with Vague States

Mohammad Hossein Fazel Zarandi[1], Javid Jouzdani[1], and Ismail Burhan Turksen[2]

[1] Department of Industrial Engineering, AmirKabir University of Technology,
Tehran, Iran, P.O. Box: 15875-4413
Zarandi@aut.ac.ir, Javid@aut.ac.ir
[2] Department of Mechanical and Industrial Engineering, University of Toronto, Toronto,
ON., Canada M5S2H8
Turksen@mie.utoronto.ca

Abstract. This paper presents a generalized method for tuning a fuzzy logic controller based on reinforcement learning in a dynamic environment. We extend the Generalized Approximate Reasoning-base Intelligent Controller (GARIC) model of Berenji and Khedkar to be able to work in presence of vagueness in states. Similar to GARIC, the proposed architecture, i.e., Generalized Reinforcement Learning Fuzzy Controller (GRLFC), has the self-tuning capability even when only a weak reinforcement signal such a binary failure signal is available. The proposed controller shows a better performance, regarding learning speed and robustness to changes in controlled system dynamics, than similar models even in the presence of uncertainty in states.

Keywords: fuzzy control, reinforcement learning, fuzzy systems.

1 Introduction

In the literature, there exist two methods for development of a fuzzy model: direct approach and indirect approach. In direct approach to fuzzy modeling, the rules governing the system are described linguistically using terms of natural language and then they are turned into fuzzy rules. This linguistic description is constructed subjectively according to prior knowledge about the system. Therefore, if the expert's knowledge about the system is faulty, an unrobust model may be obtained. These rules should be fine-tuned in order to be used for control purposes.

Another direction in development of fuzzy models is through using input-output data. This is what is called indirect approach to fuzzy modeling. The problems of extracting fuzzy rules from data arose in the first years after the birth of fuzzy modeling concepts. Since the fuzzy rules are constructed based on data, if the data are faulty, damaged or noisy, the obtained rules cannot be reliable; i.e., crude rules may be obtained that need to be fine-tuned.

Reinforcement learning can be used to fine-tune a fuzzy logic controller; either for structure identification (e.g., [1], [2]) or for parameter identification (e.g., [3], [4]). In this paper we concentrate on the latter issue.

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 811–820, 2007.
springerlink.com

2 Fuzzy Inference System

In this research, for the sake of simplicity, triangular membership functions are used in both antecedent and consequent parts of Fuzzy Inference System (FIS). These membership functions are determined by three parameters: centre (C), right spread (RS) and left spread (LS). Under this assumption, the degree of matching between the input variable and its corresponding antecedent label can be easily calculated by using MaxMin operator:

$$MaxMin(A,B)=\begin{cases}1 & RS_a=\infty\ or\ LS_b=\infty\\ 0 & RS_a=0\ and\ LS_b=0\\ \psi\left(\dfrac{C_a-C_b+RS_a+LS_b}{RS_a+LS_b}\right) & otherwise\end{cases} \tag{1}$$

where, ψ is a function defined as:

$$\psi(x)=\begin{cases}0 & x\le 0\\ x & x>0\end{cases} \tag{2}$$

The degree of applicability of a rule can be determined by applying a t-Norm on the degrees of matching between each input variable of the rule and their corresponding antecedent labels. GRLFC inherits its t-Norm from GARIC in which Softmin, defined in (3), is used as the t-Norm. In addition, we assume that $k=10$ [3]. Any other kind of t-Norm would be appropriate for our system as well, because we don't tune the antecedent labels and we don't need its differentiability.

$$SoftMin(x_1,x_2,\dots,x_n)=\frac{\sum_{i=1}^{n} x_i \exp(-kx_i)}{\sum_{i=1}^{n}\exp(-kx_i)} \tag{3}$$

So, using the Softmin operator and degrees of matching, the degree of applicability or the degree of firing of rule R_i is calculated using:

$$w_i=SoftMin(MaxMin(A_1,B_{i1}),\dots,MaxMin(A_n,B_{in})) \tag{4}$$

where, w_i is the degree of firing of R_i, A_j is the j^{th} input variable and B_{ij} is the j^{th} antecedent label in the i^{th} rule, R_i. By applying the degree of firing of R_i on its consequent part the output of the rule can be calculated. We define

$$z_i=\mu_{C_i}^{-1}(w_i) \tag{5}$$

where, μ^{-1} is a defuzzification method and C_i is the consequent label of the i$^{\text{th}}$ rule. In this paper, we use the LMOM method for defuzzification [34].

Combining the outputs of all rules, a crisp control action in the form of weighted average is obtained using the following equation:

$$F = \frac{\sum_{i=1}^{m} z_i w_i}{\sum_{i=1}^{m} w_i} \tag{6}$$

where, m is the number of the rules and z_i is the output for the i^{th} rule. This can also be extended for multiple output variables.

3 Reinforcement Learning

Initially, Reinforcement Learning (RL) has its roots in psychology. The idea behind RL is learning from experience and through trial-and-error interactions with a dynamic environment as any intelligent creature does during its life time.

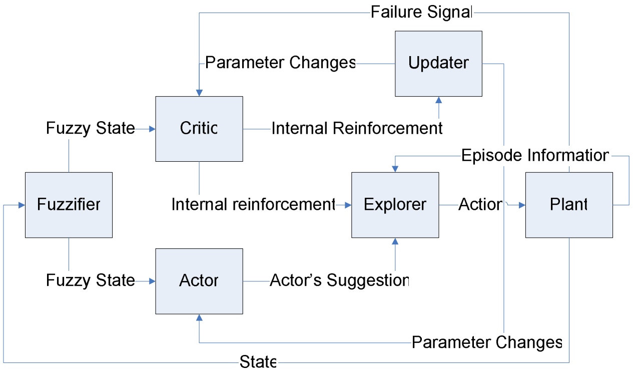


Fig. 1. The architecture of GRLFC

In the most states encountered, there is not any exact former experience. This is almost always the case when the state or action spaces include continuous variables or complex and vague sensations. Therefore, a generalization method is needed. Specifically, the kind of generalization we require is often called *function approximation* because it takes examples from a desired function (e.g., a value function) and attempts to generalize from them to construct an approximation of the entire function.

As a method of generalization, a Fuzzy Inference System is an appropriate tool. Furthermore, the use of FIS rather than global function approximators like Neural Networks has two major advantages: The FIS inherent locality property permits the introduction of human knowledge, and also localizes the learning process to only implicated parameters [5]. However, the process of fine-tuning the fuzzy controller remains a difficult task.

It should be noted that in most real-world applications, training data is often hard to obtain or may not be available. To resolve these important issues, reinforcement learning techniques can be effective for fine-tuning of fuzzy systems, when no training data is available and only a weak reinforcement can be obtained.

Here, several researchers have applied RL techniques to tune a conventional Neural Network as the critic while the actor is an FIS [3], [6]. Others have used Fuzzy Inference Systems to increase the knowledge of the critic about the goodness of the states and consequently enhance the performance of the system [4], [5]. However, barely researchers have considered the vagueness in input states. In this paper we propose a model that captures the uncertainty in state of the system.

4 The Architecture of GRLFC

In the proposed model, a fuzzy controller is implemented by means of an FIS which plays the role of the *Actor*. The Actor implements the knowledge of the expert operator about how to control the system. The *Critic* which evaluates the value of the current state is another FIS and it incorporates the knowledge about the goodness of the states of the plant. Both these components simultaneously learn to improve their performance through interaction by a dynamic environment and receiving a reinforcement signal.

4.1 Fuzzifier

As mentioned before, in many real-world problems, uncertainty exists in the states of a system. This can be caused by many factors like uncertainty in sensor readings or uncertain nature of the states of the system (linguistic input states). The *Fuzzifier* considers the uncertainty in the input variables by constructing a symmetric or asymmetric triangular fuzzy membership function using the crisp input state.

4.2 Actor and Critic

Both the *Actor* and the *Critic* are fuzzy inference systems described in section 2 and have similar architectures depicted in Fig. 2. In this scheme, the first layer is the input layer and accepts triangular fuzzy membership functions as fuzzy states. Layer 2 contains the antecedent labels and determines the degree of matching using MaxMin operator. In layer 3 the degree of firing of the rule is calculated using the Softmin operator and the degrees of matching between the fuzzy input variables and their corresponding antecedent labels. Consequent labels are in layer 4 where defuzzification is performed using LMOM defuzzification method [3] and then the output of the each rule is calculated. Layer 5 is the output layer in which the crisp control action is determined.

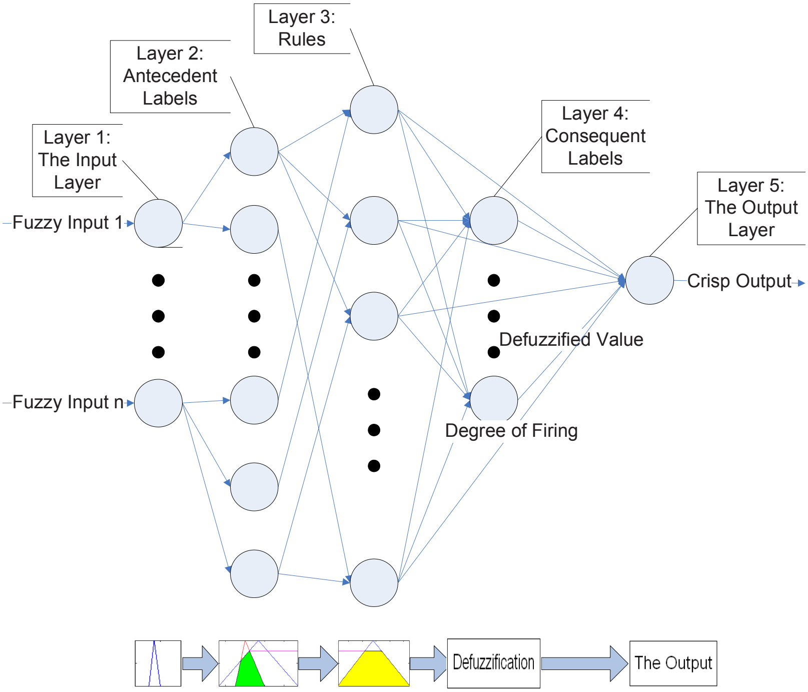


Fig. 2. FIS Scheme

4.3 Explorer

This component makes a trade-off between exploration and exploitation in the state space. Since we assume that the knowledge-bases of the *Actor* and the *Critic* may be rough and out of tune, in the early steps of simulation (or the actual run), the state space must be sufficiently explored. When the time passes and the *Actor* learns to suggest more appropriate actions and the *Critic* learns to correctly predict the state values by the trial-and-error interaction with the environment. Therefore, exploration is smoothly substituted by exploitation.

The *Explorer* accomplishes this task by perturbing the action, F , suggested by the *Actor* using the TD prediction error [7], δ , given by (13), and the length of the previous episode, T . This process is done via the following equation:

$$F' = F + R\sigma(\delta, T) \tag{7}$$

where, R is a standard uniformly distributed random variable on $[-a, a]$ interval and σ is some monotonically decreasing function with respect to T and the magnitude of δ . In this way, when the magnitude of δ is large (small), there will be a large (small) difference between the *Actor's* suggested action, F , and what is actually applied to the plant, F' . This provides the exploration of the state space.

The *Explorer* also provides the perturbation needed by the *Updater* in updating the parameters of the *Actor* and the *Critic*. The perturbation, s , is calculated using the following equation:

$$s = \exp\left(\left((F' - F)\delta\right)^2\right). \tag{8}$$

4.4 Updater

This component tunes the labels in consequent part of the *Actor* and the *Critic* using a decaying learning rate, the TD error and the gradient of each FIS (*Actor* and *Critic*) with respect to parameters of consequent parts of corresponding FIS.

For the *Actor*, the parameters are tuned to reach the objective of maximizing the state value so the systems ends up in a good state and eventually avoids failure. This can be done through (9) in which p is the parameter to be tuned, v is the state value determined by the *Critic*, and α is the learning rate which is a small positive variable.

$$\Delta p = \alpha \frac{\partial v}{\partial p} = \alpha \frac{\partial v}{\partial F} \frac{\partial F}{\partial p} \tag{9}$$

To do this we need the two derivatives on the right hand side of (9). $\partial v / \partial F$ is approximated using (10):

$$\frac{\partial v}{\partial F} \approx \frac{dv}{dF} \approx \frac{v_t - v_{t-1}}{F_t - F_{t-1}} \tag{10}$$

Calculation of $\partial F / \partial p$ is not difficult. V , a label in the consequent part of the *Actor*, is parameterized by p_V which may be center, left spread, or right spread and R_V is the rule that its consequent label is V . In addition, z_{R_V} is the defuzzified output of the rule R_V calculated by (11) using LMOM defuzzification method [4] and w_{R_V} is the degree of firing of rule R_V . Thus,

$$z_{R_V} = C_V + \tfrac{1}{2}\left(RS_V - LS_V\right)\left(1 - w_{R_V}\right) \tag{11}$$

In (11), C_V is the center of the label V and RS_V and LS_V are its right and left spreads, respectively. Thus, the derivative needed for tuning of the *Actor* can be calculated using (10) and (12).

$$\frac{\partial F}{\partial p_V} = \frac{1}{\sum_i w_i} \sum_{V=R_V} w_{R_V} \frac{\partial z_{R_V}}{\partial p_V} \tag{12}$$

In the above equation, $i \in \{1,2,\ldots,m\}$, where m is the number of rules in the knowledge base of the *Actor*.

For the *Critic*, the objective is minimization of the TD prediction error, δ_t , given by the following equation:

$$\delta_t = \begin{cases} 0 & \textit{starting state} \\ r_{t+1} - V_t(s_t) & \textit{failure state} \\ r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t) & \textit{otherwise} \end{cases} \tag{13}$$

where, $V_i(s_j)$ is the *Critic* estimation for the value of state s_j in time step i. The term $r_{t+1} + \gamma V_t(s_{t+1})$ is actually an estimation of $V_t(s_t)$ and, therefore, δ_t is the error of estimation. γ is the reward discount rate which determines the importance of the future time steps in current learning cycle. If γ is set to 0, only the immediate reward is considered and if set to 1, all future steps are taken into account in learning process. In our experiments, we use $\gamma = 0.95$, since this value is usually used by researchers and yields better results. Therefore, the learning rule for the *Critic* is given by (14) in which β is the learning rate and like α is small positive variable. In the beginning of each episode, the learning rates are set to a relatively large value. Then during the episode the learning rates decay quickly and reach a small value and from that point forward, the learning rates decay after each N time steps to provide more exploitation of the good policy found by the controller.

$$\Delta p = -\beta \frac{\partial \delta}{\partial p} = -\beta \frac{\partial \delta}{\partial v} \frac{\partial v}{\partial p} \tag{14}$$

The term $\partial v / \partial p$ is easily calculated similar to the calculation of $\partial F / \partial p_V$ in (12). This is because the *Critic*, like the *Actor*, is an FIS. The other term, $\partial \delta / \partial v$ is approximated by using (15), assuming that the derivative does not depend on r.

$$\frac{\partial \delta}{\partial v} \approx \frac{d\delta}{dv} = (1-\gamma) + \gamma(d^2 v) \tag{15}$$

where, just like the *Actor* learning rules, only the sign of $\partial \delta / \partial v$ is used in calculations. Furthermore, $d^2 v$ is given by the finite difference $v_t - 2v_{t-1} + v_{t-2}$.

5 Experiments

To show the efficiency and superiority of the proposed system, we applied the GRLFC to a well-known control problem called Cart-Pole. In this problem a pole is hinged to a cart which moves along a track. The control objective is to apply an appropriate force, F, to keep the pole vertically and the cart within track boundaries. The state of the system is determined by $(x, \dot{x}, \theta, \dot{\theta})$ in which x and $\dot{x}$ are the displacement and velocity of the cart, and θ and $\dot{\theta}$ are the angular displacement and angular velocity of the pole respectively. A failure occurs when either $|\theta| > 12^\circ$ or $|x| > 2.4m$ or $|F| > 10N$, whereas a success is when the pole stays balanced for $100{,}000$ time steps. In our experiment, δ is calculated using $\gamma = 0.95$. In addition, half-pole length, $l_p = 0.5\,m$, pole mass, $m_p = 0.1\,kg$ and cart mass, $m_c = 1\,kg$. The dynamics of the cart-pole system are modeled by the following nonlinear differential equations:

$$\begin{aligned}\ddot{\theta} &= \frac{g\sin(\theta)+\cos(\theta)\left[\dfrac{-F-m_p l_p\dot{\theta}^2\sin(\theta)+\mu_c\,\mathrm{sgn}(\dot{x})}{m_c+m_p}\right]-\dfrac{\mu_p\dot{\theta}}{m_p l_p}}{l_p\left[\dfrac{4}{3}-\dfrac{m_p\cos^2(\theta)}{m_c+m_p}\right]},\\ \ddot{x} &= \frac{F+m_p l_p\left[\dot{\theta}^2\sin(\theta)-\ddot{\theta}\cos(\theta)\right]-\mu_c\,\mathrm{sgn}(\dot{x})}{m_c+m_p}\end{aligned} \qquad (16)$$

One of the strong points of the proposed model is its capability of capturing the uncertainty in the state of the system. In our particular experiment with the cart-pole problem, vagueness may be caused by uncertainty in sensor readings. This is done by the *Fuzzifier* and through constructing a triangular fuzzy membership function that has the crisp state as its center, C, and $C - LS$ and $C + RS$ as its left spread and right spread respectively. In this experiment we consider symmetric membership functions with $RS = LS = 0.001$.

	NE	ZE	PO	VS
NE	NL	NS		ZE
ZE	NM	ZE		PM
VS				
PO	ZE	PS		PL

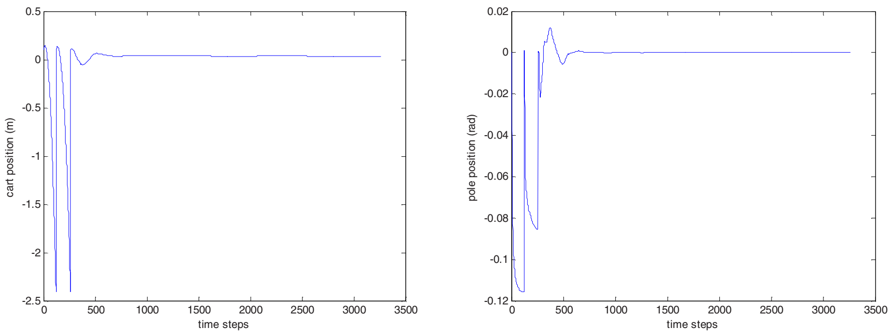
	NE	ZE	PO	VS
NE	NS			
ZE				
VS	NVS			PVS
PO				PS

Fig. 3. 9+4 rules used in Actor

	PO	ZE	NE
PO	BAD		OK
ZE		GOOD	
NE	OK		BAD

	PO	ZE	NE
PO	BAD		OK
ZE		GOOD	
NE	OK		BAD

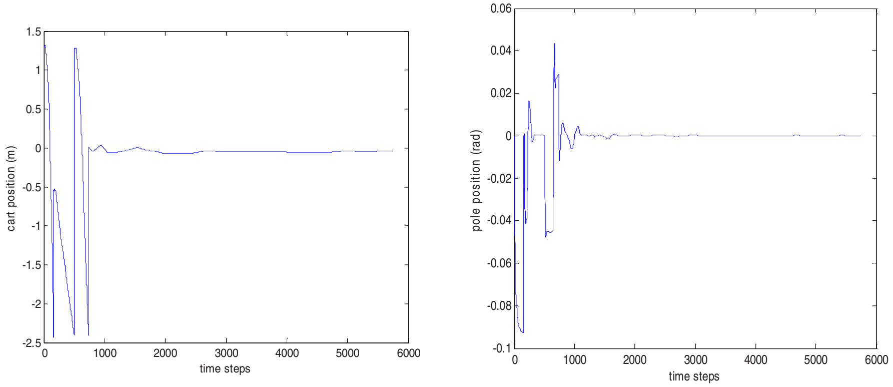
Fig. 4. 5+5 rules used in Critic



(a) Cart position

(b) Pole position

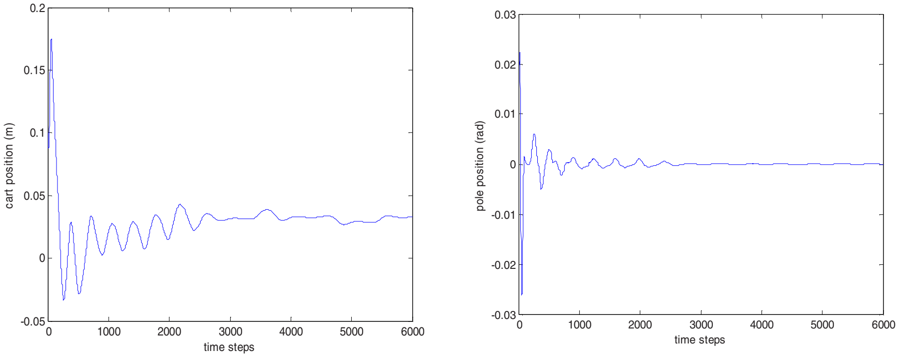
Fig. 5. The center of ZE force label was shifted to +5. The system shifted it back to about 0.



(a) Cart position

(b) Pole position

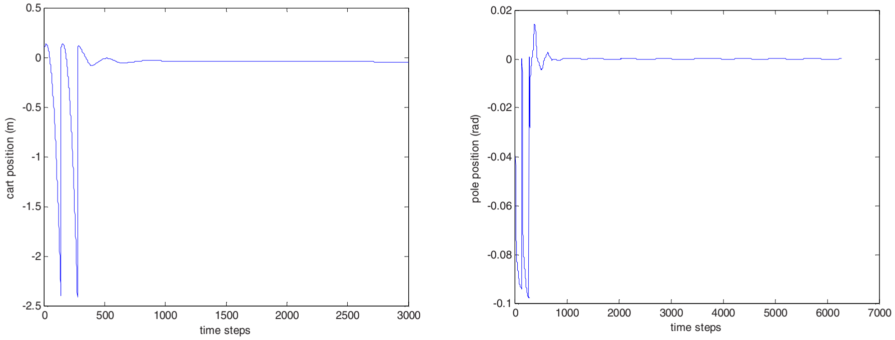
Fig. 6. The center of ZE force label was shifted to +3. The system shifted it back to about 0.



(a) Cart position

(b) Pole position

Fig. 7. Various force label were damaged: ZE-2, PL+5, PM+3, PS+2, NM+1



(a) Cart position

(b) Pole position

Fig. 8. The center of GOOD label was shifted to 0.5. The system shifted it back to about 1.

In our experiment, the *Actor* has the same 9+4 rules of the Action Selection Network (ASN) (Fig. 3) and the Critic has same 5+5 rules of the Action Evaluation Network (AEN) (Fig. 4) in GARIC [4]. Figures 5-8 show the simulation results of experiments in which various labels of the *Actor* or the *Critic* were damaged and the system has managed to tune those parameters to their appropriate values.

6 Conclusion

This research has modified GARIC model [3]. The proposed model extends GARIC in several ways: it is capable to capture the vagueness in input variables. Furthermore, the learning speed is increased and the number of failures before a success is decreased using variable learning rates. Learning strategy for the *Critic* is different from that of Action Evaluation Network (AEN). The *Explorer* component also extends the Stochastic Action Modifier (SAM) in GARIC to provide better exploration of the state space.

References

1. C. Lin, Y. Xu,: A Novel Genetic Reinforcement Learning for nonlinear fuzzy control problem. Neurocomputing, (2005)
2. C. Lin, C. Lee, Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems. IEEE Trans. Fuzzy Syst. 2 (1994) 46–63.
3. H. R. Berenji, P. S. Khedkar, Learning and Tuning Fuzzy Logic Controllers through Reinforcements, IEEE Trans. on Neural Networks, Vol. 3, No. 5, (1992).
4. H. R. Berenji, P. S. Khedkar, Using Fuzzy Logic for Performance Evaluation in Reinforcement Learning, Intl. J. Approximate Reasoning (1997) 131-144
5. L. Jouffe, Actor-Critic Learning based on Fuzzy Inference System, IEEE (1996)
6. C. Chiang, H. Chung and J. Lin, A Self-Learning Fuzzy Logic Controller Using Genetic Algorithms with Reinforcements, IEEE Trans. Fuzzy Syst. Vol. 5, No. 3, (1997)
7. R. S. Sutton, Reinforcement Learning: An Introduction, MIT Press (1998)

New Cluster Validity Index with Fuzzy Functions

Asli Çelikyilmaz[1] and I. Burhan Türkşen[2]

[1] Dept. of Mechanical and Industrial Engineering, University of Toronto, Canada
asli@mie.utoronto.ca
[2] Dept. of Industrial Engineering, TOBB-University of Economics and Technology, Ankara, Turkey
bturksen@etu.edu.tr

Abstract. A new cluster validity index is introduced to validate the results obtained by the recent Improved Fuzzy Clustering (IFC), which combines two different methods, i.e., fuzzy c-means clustering and fuzzy c-regression, in a novel way. Proposed validity measure determines the optimum number of clusters of the IFC based on a ratio of the compactness to separability of the clusters. The compactness is represented with: (i) the sum of the average distances of each object to their cluster centers, and (ii) the error measure of their fuzzy functions, which utilizes membership values as additional input variables. The separability is based on the ratio between: (i) the maximum distance between the cluster representatives, and (ii) the angles between their representative fuzzy functions. The experiments exhibit that the new cluster validity index is a useful function when selecting the parameters of the IFC.

Keywords: Cluster validity, improved fuzzy clustering.

1 Introduction

Fuzzy set-based approaches are developed to identify the uncertainty of the given system by means of linguistic terms represented with membership functions. The most commonly used method to identify the hidden patterns and assign membership values to every object in the given system is the Fuzzy *c*-Means (FCM) [1] clustering algorithm. After that, many variations of the FCM are developed for different purposes, e.g., [4], [6], [9], [11], [12].

Recently, the authors have developed an Improved Fuzzy Clustering (IFC) [4] algorithm, combining the standard fuzzy clustering, i.e., FCM [1] and fuzzy c-regression, i.e., Fuzzy c-regression Models (FCRM) [11], for identification of the Fuzzy System Models (FSM) with Fuzzy Functions (FFs). They have shown that when the new IFC is utilized in FSM, FFs can provide better estimations. Coupling the standard clustering and regression algorithms, the new IFC [4] approach explores the given dataset to find local fuzzy partitions and simultaneously builds *c* regression models (functions). The prominent feature of the IFC [4] approach is that, the membership values and their user defined transformations, where appropriate, are used as the only input variables in the regression models to find the optimum representation of the input-output data. Hence, the objective function of the IFC algorithm is based on the distances of every data point to each cluster center as well as the error of different regression models. Alienation of the original input variables

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 821–830, 2007.
springerlink.com

from the regression functions during the IFC helps to obtain improved membership values to be later used to explain the output variable in each local fuzzy model.

Every fuzzy clustering approach, including the latest algorithm in [4], assumes that some initialization parameters are known. Based on the previous research, many cluster validity index-(cvi)s have been proposed to validate the underlying assumptions of the number of clusters, e.g., [2], [3], [13], [14], [16], [21]. These cvis try to identify the characteristics of a point-wise fuzzy clustering method, i.e., the FCM. These cvi's use either within cluster, i.e., *compactness*, or between cluster distances, i.e., *separability*, or both to assess the clustering schema and they are categorized into two types: *ratio* type and *summation* type [13]. The type of the cvi is determined by the way the within cluster and between cluster distances are coupled.

Nevertheless, most of these cvi's may not be suitable for other variations of fuzzy clustering algorithms, which are designed for different purposes, e.g., FCRM [11]. For these types of FCM variations, different validity measures are introduced [15]. In this paper, a new cvi, which is a ratio type index between the compactness and the separability measures, is proposed to measure the optimum parameters of the previously proposed IFC algorithm [4]. Since IFC joins two different fuzzy clustering approaches, i.e., FCM [1] & FCRM [11], in novel way and utilizes the fuzzy functions, the new validity index will include these concepts in one validity function.

In the next section the IFC [4] algorithm is briefly reviewed. In section 3, the new cvi for IFC algorithm will be introduced. Section 4 will present the simulation results using 4 different datasets to illustrate the performance of the new validity function comparing the results to 3 other well-known cvi's, which are closely related to the proposed validity measure.

2 Improved Fuzzy Clustering Algorithm (IFC)

We will first review the general framework of the fuzzy functions (FFs) approaches [5], [19], [20]. For the given multi-input and single-output system, let the given data matrix be represented as $X=\{\mathbf{x}\in \Re^{nv}, y\in \Re \,|(\boldsymbol{x}_1,y_1), \ldots, (\boldsymbol{x}_{\mathrm{nd}},y_{\mathrm{nd}})\}$, where $\boldsymbol{x_k}=\{x_1,\ldots, x_{nv}\}$ is the kth data vector of nv dimensions, $k=1,\ldots,nd$, nd is the total number of data vectors and y_k is their corresponding output values. In the learning phase of the previous FSM with FFs approaches, briefly, one executes a supervised FCM clustering using the input-output dataset to find the hidden patterns in the data and finds interactive membership values, μ^{xy}_{ik}, of each kth vector in each cluster i, $i=1,..,c$, number of clusters. Then, the input membership values, μ^{x}_{ik}, are calculated and then normalized, γ_{ik}, with an alpha-cut, α-cut, to prevent any harmonics. A different dataset is structured for each cluster i, by using normalized membership values, γ_{ik}, and/or their transformations as additional dimensions. This is same as mapping the original input space, $\Re^{nv}$, onto a higher dimensional feature space $\Re^{nv+nm}$, *i.e.*, $\boldsymbol{x}\rightarrow\Phi_{\mathrm{i}}(\boldsymbol{x}, \gamma_i)$, for each cluster $i,i=1,\ldots,c$. Hence each data vector is represented in $(nv+nm)$ space, where nm is the number of augmented membership values as new dimensions to the original input space. A regression function, $\hat{y}_i=f(\Phi_i(\boldsymbol{x},\gamma_i),\beta_i)$, is fitted for each new dataset of each cluster, $\Phi_{\mathrm{i}}(\boldsymbol{x},\gamma_{ik})$, using any linear or non-linear regression methods, e.g., LSE, support vector regression (SVR) [7] to estimate one output value for every input data vector for each cluster. To defuzzify the result, one single output value of every input data vector is obtained by weighing with their corresponding membership values.

Previously we proposed a new fuzzy clustering method, IFC [4] to be used in FSM with FFs approaches instead of FCM. We modified the FCM algorithm to find the membership values, which can decrease the error of FFs approaches. The IFC not only searches for the best partitions of the data, but also aims at increasing the predictive power of the calculated membership values to model the input-output relations in local fuzzy functions.

The IFC algorithm introduces a new objective function, which carries out two purposes: to find (i) a good representation of the partition matrix; (ii) membership values to minimize the error of the FF models. Hence, we appended the error of the regression functions to the objective function as follows:

$$J_m^{IFC} = \Sigma_i \, \Sigma_k \, (\mu_{ik})^m \, (d_{ik})^2 + \Sigma_i \, \Sigma_k \, (\mu_{ik})^m \, [y_k - f(\boldsymbol{\tau}_{ik} \, ,\delta_i)]^2 \, . \tag{1}$$

where d_{ik}=|| $(\boldsymbol{x},y)_k$-$v_i(\boldsymbol{x},y)$||. The first term of J_m^{IFC} is same as the FCM algorithm, which vanishes when each example is a cluster by itself. The second term is the total squared error (SE) of the fuzzy function $f(\boldsymbol{\tau}_{ik},\delta_i)$ of each cluster, *i*, that are built during the IFC algorithm using the input matrix, $\boldsymbol{\tau}_i(\mu_i)$, which includes the membership values and/or their possible transformations as input variables excluding the original scalar inputs. δ_i are the regression parameters. We want to improve the predictive power of the membership values to be used in system models as additional inputs. Excluding the original input variables from the function estimation, we can improve the prediction power of the membership values.

The approach is novel in comparison to the previous improved FCM optimization approaches, e.g., [11], [12], etc, where they interpreted these membership values as *weights* attached to the models using weighted regression models to estimate the functions with the original input variables. It should also be emphasized that the first term uses input-output dataset $(\boldsymbol{x},y)$, whereas in the earlier approaches only the inputs $(\boldsymbol{x})$ are used during clustering. From the Lagrange transformation of the objective function, which can be found in [4], a new membership functions is extracted as:

$$\underset{\substack{1<i\leq c\\ 1<k<nd}}{\mu_{ik}} = \sum_{j=1}^{c} \left[\frac{d_{ik}^2 + \left(y_k - f_i(\boldsymbol{\tau}_{ik},\boldsymbol{\delta}_i)\right)^2}{d_{jk}^2 + \left(y_k - f_j(\boldsymbol{\tau}_{jk},\boldsymbol{\delta}_j)\right)^2} \right]^{1/(1-m)} \tag{2}$$

The cluster centre, $v_i(\boldsymbol{x}y)$, formulation of the IFC remains as same as the FCM [1]:

$$\forall v_i = [\Sigma_k \, (\mu_{ik})^m \, \boldsymbol{x}_k] \, / \, \Sigma_k \, (\mu_{ik})^m, \quad , k=1,\ldots,nd. \tag{3}$$

A generalized framework of the proposed IFC is shown in Table 1. Let the fuzzy partition at each iteration *t* be denoted with U^t. The algorithm starts with an initial partition matrix, $U^{\,t=0}$, and initial cluster centers, $v^{t=0}$. The $U^{t=0}$ and $v^{t=0}$ are required inputs at the start of the IFC optimization since the new membership function in (2) includes the error terms of the regression models, $[y_k - f(\boldsymbol{\tau}_{ik}^{\;t-1} \, ,\delta_i^{\;t-1})]^2$ from previous iteration to approximate the actual output, y_k, using only the membership values and their transformations as input variables. One may use a crisp clustering such as k-means or a fuzzy clustering such as fuzzy k-means, or standard FCM, to find the $U^{\,t=0}$, and $v^{\,t=0}$.

Table 1. Improved Fuzzy Clustering Algorithm

1. Given the training dataset, $X=\{(\boldsymbol{x}_1,y),\ldots,(\boldsymbol{x}_{nd},y_{nd}))\}$ Set $m>1.1$, $c>1$ and a termination constant, $\varepsilon>0$, maximum number of iterations (max-iter), an initial partition U^0 and $obj^0=0$. Specify the nature of the regression models, τ_i, for each cluster *i*, $i=1,\ldots,c$, $k=1,\ldots,nd$, to create the input matrix structure. For each iteration, $t=1,\ldots max\text{-}iter$;
2. Populate the c number of input matrices, $\boldsymbol{\tau}_i^{t-1}$, one for each cluster *i*, using membership values (U^{t-1}) from the (t-1)th iteration, and their selected user defined transformations.
3. Calculate the values for the *c* number of regression parameters, δ_i^{t-1}, that can minimize the error of user defined function, i.e., special function, $\hat{y}_{ik}=f(\boldsymbol{\tau}_i^{t-1},\delta_i^{t-1})$.
4. Calculate the cluster centers for iteration t: $\forall v^t_i = [\Sigma_k (\mu^{t-1}_{ik})^m \boldsymbol{x}_k] / \Sigma_k (\mu^{t-1}_{ik})^m$
5. Calculate the membership values with the membership function in (2) for iteration t using the regression functions from step 3.
6. Calculate value of the objective function, obj^t where the first and the second terms are defined as: $obj^t = \Sigma_i \Sigma_k \mu^t_{ik} (d^2_{ik})^t + \Sigma_i \Sigma_k \mu^t_{ik} [y_k - (\hat{y}_{ik})^{t-1}]^2$
7. If (obj^t - obj^{t-1})<ε, then terminate, else return to step 3.

One could use any regression function model to find the parameters of the fuzzy functions, e.g., least squares regression (LSE) or support vector regression (SVR) [7] with kernel functions to approximate the output variable. In this paper, we only used LSE method during the IFC to find the relationship between the memberships and the output variable in each cluster. As a result of the IFC algorithm, the optimized membership matrix, $U^*(\boldsymbol{x},y)$, the cluster centers $v^*(\boldsymbol{x},y)$, and the parameters of the functions for each cluster from the last iteration $f(\boldsymbol{\tau}_{ik}^*,\delta_i^*)$ are obtained.

3 A New Cluster Validity Criterion

This paper investigates the validity indexes for two different types of fuzzy clustering algorithms: point-wise clustering, e.g., FCM, and regression type clustering, e.g., FCRM algorithms [11] due to the fact that the new IFC [4] couples the regression and point-wise clustering concepts. We hypothesize that the new validity index should include the concepts from two different types of cvis.

Most cvis are structured combining two different concepts [14]; *(i) compactness,* which measures the similarity between cluster elements within each cluster and *(ii) separability,* which measures the dissimilarity between each individual cluster. A well-known ratio-type cvi (*compactness/separability)* is the XB index [21];

$$XB(nc)=\left(\sum_{i=1}^{c}\sum_{k=1}^{nd}\mu_{ik}^2 d(x_k,v_i)^2\right)\Big/\left(nd\cdot\min_{i,j\neq i} d(v_i,v_j)^2\right),\quad d(x_k,v_i)=\left\|x_k-v_i\right\|^2. \tag{4}$$

where $v_i\in\Re^{nv}$ represents the cluster center as a vector of *nv* dimensions. XB decreases monotonically when *c* is close to the total number of data samples. The compactness increases sharply as *c* decreases from c_{opt} *(optimum number of clusters)* to c_{opt}-1 [14]. A sudden drop in compactness would be the indicator of the c_{opt}.

It is evident that each cluster has different compactness values because some clusters are more (less) dense than the others. In XB index, the compactness of the whole clustering structure is determined (the numerator in (4)) by averaging the compactness of every cluster. However, averaging might suppress the effect of the large changes in the compactness's of some clusters. To exploit these huge shifts in compactness during determination of the c_{opt}, an improved version of XB index is introduced in [14] as:

$$XB^{*}(c)=\left(\max_{i=1,..,c}\sum_{k=1}^{nd}\mu_{ik}^{2}\left\|x_{k}-v_{i}\right\|^{2}\right)\Big/ nd\cdot\min_{i,j\neq i}\left\|v_{i}-v_{j}\right\|^{2}\cdot \tag{5}$$

XB* index is proved [14] to be more effective than the XB index, because with *XB** one can detect the clusters with large compactness, which helps to determine the c_{opt}.

On the other hand, Kung and Lin [15]'s cvi validates the FCRM [11] type clustering approaches. Their compactness measure is based on the XB index; however, it is measured by the regression model fit between the output from each model and actual output. They measure the separability by the inverse dissimilarity between the clusters defined by the absolute value of the standard inner-product of the unit normal vectors, which represent c hyper-planes. The Kung-Lin's cvi is formulized as follows:

$$Kung-Lin(c)=\left(\sum_{i=1}^{c}\sum_{k=1}^{nd}\mu_{ik}^{2}\left\|x^{T}\theta_{i}-y_{k}\right\|^{2}\right)\Big/\left(nd\Big/(\max_{i\neq j}(\left|\left\langle u_{i},u_{j}\right\rangle\right|)+\kappa)\right) \tag{6}$$

where $\theta_i=[\boldsymbol{x}^{\mathrm{T}}\mu_i\,\boldsymbol{x}]^{-1}\,[\boldsymbol{x}^{\mathrm{T}}\mu_i\,\boldsymbol{x}]y$. The u_i's represent the unit normal vector of each c regression functions. The FCRM clusters [11] are represented with the regression equations and therefore their corresponding unit vectors are defined as $u_i=[n_i]/\|n_i\|$, where $n_i=[\theta_{I,1}\ \cdots\ \theta_{i,nv}\ \ -1]\in\Re^{nv+1}$.The n_i indicates the regression function parameters of the i^{th} model in a vector form and $\|\cdot\|$ is the Euclidean norm. Kung and Lin [15] introduced the inner product of the two unit vectors of clusters to measure the separability of the c regression functions.

In this paper, a new ratio-type cvi is proposed for the IFC [4] algorithm to find the optimum number of clusters, c_{opt}. In IFC, the clusters are identified by cluster prototypes (centers) and their corresponding regression functions. The membership values represent the degree to which each object belong to each prototype that is identified. They are also used as candidate input variables, which can help to identify a regression functions along with the original input variables to model the output variable for each prototype.

In order to fully validate the IFC, we need to find a relation between the compactness and separability of clusters by measuring the latter two separate concepts; the clustering structure as well the relationships between their function representations. The compactness of the proposed cvi, **cviFFnew,** will combine two terms. We will use the XB* compactness (numerator in (5)) as the first term of the compactness and a modified version of the compactness of Kung-Lin index (numerator in (6)) as the second term. The second term of the new compactness will represent the error between the regression model and the actual output. The regression models are the "*fuzzy functions*" (FFs), $f:(x,\mu)\rightarrow y$, which will be captured with regression algorithms using the membership values and the original input variables as explanatory variables (predictors) as opposed to being used as *weights* in regression algorithm. The separability of the cviFFnew couples the angle between the regression functions and the distance between each input-output vector and the cluster centers.

IFC algorithm [4] calculates membership values, which are also used together with the original input variables to find FFs for each cluster using any regression algorithm, e.g., LSE, SVR, in system modeling with fuzzy functions. Therefore, we want to validate the optimum number of clusters of the new IFC [4] algorithm by analyzing the behavior of the new membership values when used together with the original input variables in regression functions. Accordingly, we proceed as follows:

(i) A different dataset is structured for each cluster *i*, $\Phi_{\mathrm{i}}(\boldsymbol{x},\mu_i^{\,*}) \in \Re^{nv+nm}$ by mapping the original input space $\boldsymbol{x}\in \mathfrak{R}^{nv}$ onto a new feature space of $\in \Re^{nv+nm}$ using improved membership values ($\mu_i^{\,*}$) from IFC and/or their transformations as additional *nm* dimensions. When appropriate, we use mathematical transformations of the membership values such as $\mu_{ik}^{\,2}$, $\mu_{ik}^{\,m}$, $exp(\mu_{ik})$, $\ln((1\text{-}\mu_{ik})/\mu_{ik})$, etc., where *m* represents the degree of fuzziness.

(ii) We fit linear regression functions for each cluster, $\Phi_{\mathrm{i}}(\boldsymbol{x},\mu_i^{\,*})$. Therefore, the new cluster validity index designed for the IFC clustering is as follows:

$$\left.\begin{aligned} vc^* &= \max_{i=1,..,c} \tfrac{1}{nd} \underline{\sum\nolimits_{k=1}^{nd} \mu_{ik}^m \left(\left\| x_k y_k - v_i \right\|^2 + (y_k - f_i(\Phi_i(x,\mu_i^*),\hat{\beta}_i)^2 \right)} \\ vs^* &= \begin{cases} \min_{i,j\neq i}\left(\left\| v_i - v_j \right\|^2 \Big/ \left| \left\langle \alpha_i,\alpha_j \right\rangle \right| \right), & if \left| \left\langle \alpha_i,\alpha_j \right\rangle \right| \neq 0, i,j=1,..,c \\ \min_{i,j\neq i} \left\| v_i - v_j \right\|^2, & otherwise \end{cases} \end{aligned}\right\} \; cviFFnew = \frac{vc^*}{(c\cdot vs^*)+1} \tag{7}$$

where vc^* represents the compactness and vs^* represents the separability in the new validity measure. Let $n_i^{\,\Phi}$, $n_i^{\,\Phi}=[\beta_{i,1}\ \beta_{i,2} \cdots \beta_{i,nm}\ \beta_{i,(nm+1)} \cdots \beta_{i,(nm+nv)}\ \text{-}1]\in \Re^{nm+nv}$, represent the normal vector of the FFs obtained from the dataset in the feature space, $\Phi_{\mathrm{i}}(\boldsymbol{x},\mu_i^{\,*})\in \Re^{nv+nm}$. The α_i in the $|\langle\alpha_i,\alpha_j\rangle|\in[0,1]$, $i,j=1,\ldots,c$, represent the unit normal vector of each *i*th FF, $\alpha_i=[\,n_i^{\,\Phi}]/\|n_i^{\,\Phi}\|$. The absolute value of the inner product of the two unit vectors of the FF of the two clusters, $|\langle\alpha_i,\alpha_j\rangle|\in[0,1]$, $i,j=1,\ldots,c,\ i\neq j$, equals to the value of the *cosine* of the angle between them: $cos\theta_{i,j} = \langle\, n_i^{\,\Phi},\, n_j^{\,\Phi}\rangle /\ |\, n_i^{\,\Phi}|\ |n_j^{\,\Phi}| = \langle\alpha_i,\alpha_j\rangle$. When two cluster centers are too close to each other due to oversized number of clusters, the distance between them becomes almost (≅0) invisible, then the validity measure goes to infinity. To prevent this, the denominator of cviFFnew in (7) is increased by 1.

The compactness measure, vc^* (numerator of *cviFFnew*) of the new cvi, includes the error term of the FFs. As *c* is increased, within cluster distances will decrease, since clusters will include more similar objects. So the compactness of the clusters will be small. In addition, there is one regression function estimated for each cluster and as the number of functions is increased, the error of the FFs will decrease, because the regression model output will approach to the actual output. On the other hand, when $c< c_{opt}$, the clusters will include dissimilar objects together, which will increase the first term of the compactness in cviFFnew. Since there will be less functions than the actual number of models, the deviation between the actual and estimated error will be high.

The proposed *cviFFnew* also analyzes the maximum compactness values as do the XB^* index instead of average compactness values to identify the approximate c_{opt}. The separability measure of the *cviFFnew* also couples the separability measure from two different structures, i.e., regression and clustering. Between cluster distances are represented as the Euclidean distances between the cluster centers. The absolute value of the cosine of the angle between each FF $|\langle \alpha_i,\alpha_j \rangle| \in [0,1]$ is used as additional separability criteria. If the functions are orthogonal, then they are the most dissimilar functions. The separability, vs^* conditionally combines the between-cluster-distances and angles by taking the ratio between them.

4 Experiments

The validity of the new cvi is illustrated with a numerical example. We used 4 different datasets containing different number of linear models, which are created with varying Gaussian noise, $\varepsilon_{l,m}$=($l=1,..,4,\ m=1,\dots,nf$) of zero means and variance of 0.9 in each function. *nf* is the number of functions used to generate each dataset *l*.

Each dataset is generated randomly using linear functions in Table 2. 400 training input vectors, *x,* which are uniformly distributed in the range of [-5, +5], are generated randomly for dataset1. Then the data is grouped into 4 separate parts, where each group contains 100 observations. Each function from Table 2(A) is applied to one group to obtain the output values to incorporate four separate models in dataset1. We also generated 500, 490 and 450 more training vectors, *x,* separately which are also uniformly distributed in the range of [-5, +5] to generate the datasets with 5, 7 and 9 patterns using the functions in Table 2(B), 2(C), and 2(D). The 5, 7, 9-patterned datasets are generated in the same way as the 4-patterned dataset.

Table 2. Functions used to generate Artificial Datasets

Datasets	Functions ($y_1=x_1^T\beta_1+\varepsilon$)
(A) 4-clustered	y_1=2x+5+$\varepsilon_{1,1}$; y_2=0.2x-5+$\varepsilon_{1,2}$; y_3=-x+1+$\varepsilon_{1,3}$; y_4=x-8+$\varepsilon_{1,4}$.
(B) 5-clustered	y_1=2x+$\varepsilon_{2,1}$; y_2=7x-5+$\varepsilon_{2,2}$; y_3=-x+1+$\varepsilon_{2,4}$; y_4=x+14+$\varepsilon_{2,4}$; y_5=-5x-6+$\varepsilon_{2,5}$.
(C) 7-clustered	y_1=2x+$\varepsilon_{3,1}$; y_2=7x-5+$\varepsilon_{3,2}$; y_3=-x+1+$\varepsilon_{3,3}$; y_4=x-14+$\varepsilon_{3,4}$; y_5=-5x-6+$\varepsilon_{3,5}$; y_6=6$\varepsilon_{3,6}$; y_7=3x-25+$\varepsilon_{3,7}$.
(D) 9-clustered	y_1=x+$\varepsilon_{4,1}$; y_2=3x+2+$\varepsilon_{4,2}$; y_3=0.5x+1+$\varepsilon_{4,3}$; y_4=3x-3+$\varepsilon_{4,4}$; y_5=-x+2+$\varepsilon_{4,5}$; y_6=-2x-9+$\varepsilon_{4,6}$; y_7=-2x+$\varepsilon_{4,7}$; y_8=x-0.5+$\varepsilon_{4,8}$; y_9=-x+0.2$\varepsilon_{4,9}$.

We applied the IFC algorithm [4] to these 4 datasets separately using 2 levels of fuzziness, *m=1.3,* and *m=2.0* and for *c=2,…,15* and obtained membership matrices (partition matrix) for each different *c* and *m* combinations. We calculated the values of the proposed cviFFnew, for each combination as well as the values of XB [21], XB* [14] and Kung-Lin cvi [15]. We plotted these cvis in Fig. 1 from four different datasets for different *c* values and for two fuzziness values, *m=1.3 and 2.0. m=1.3* represents a more crisp model, where overlapping of clusters are negligible, while *m=2* is a model where the local fuzzy clusters overlap to a degree of *2.0.* We expect that the c_{opt} would be 4, 5, 7, and 9 for dataset1, 2, 3 and 4, respectively.

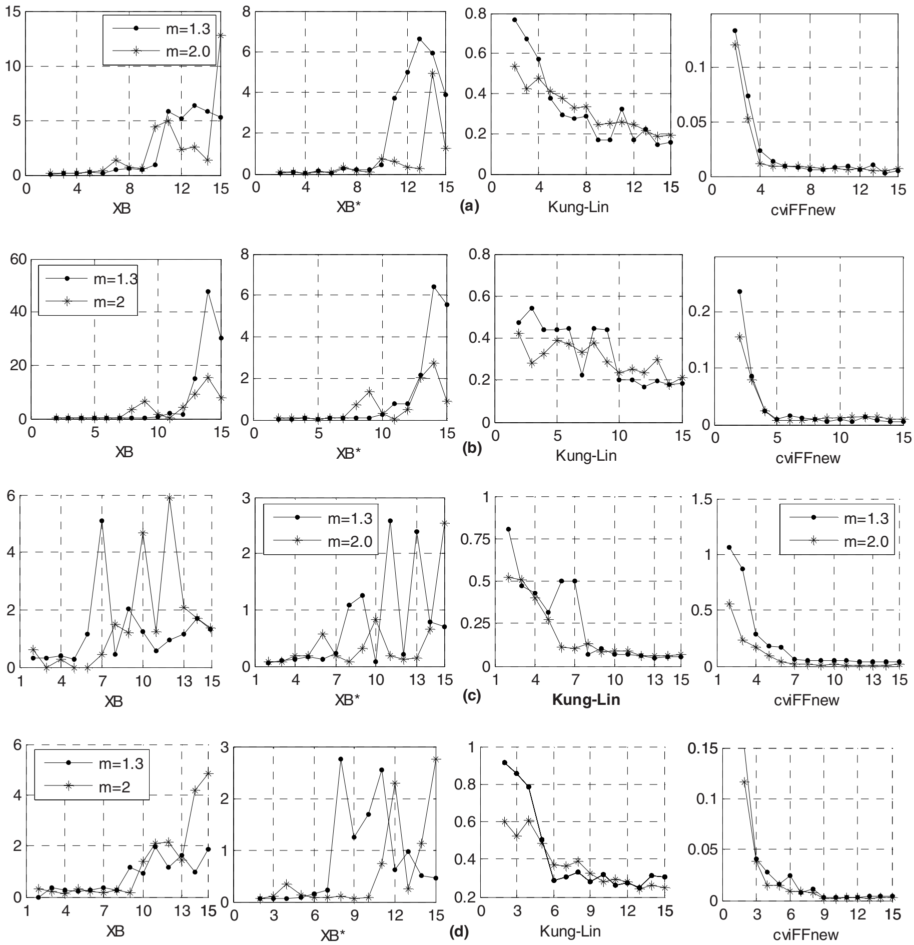


Fig. 1. (a) 4-clustered, (b) 5-clusteredm (c) 6-clustered, (d) 9-clustered Dataset. cvi index versus c for two *m* values; *m=1.3(.-) and m=2.0(*-)*

5 Discussions

Table 2 and Table 3 display the optimum number of clusters indicated by each of cvi measures, i.e., XB, XB*, Kung-Lin and cviFFnew*,* using 4 different datasets for two different fuzziness levels, i.e., *m=1.3* and *m=2.0*, respectively. From Fig. 1, Table 2 and Table 3, one concludes that the cviFFnew can successfully indicate different models in a given dataset. In all of the four datasets, the cviFFnew is not effected from the changing values of the fuzziness criteria, i.e., *m=1.3* and *m=2.0,* of the IFC algorithm and converges at the c_{opt}, and then it asymptotes to zero as the number of clusters goes to infinity, i.e., $c>c_{opt}$.

Table 3. Validity Index Measures for *m*=1.3 and *m*=2.0

	m=1.3				**m=2.0**			
Actual c_{opt}	**4**	**5**	**7**	**9**	**4**	**5**	**7**	**9**
XB	2-6	2-6	5	2, 5,8	2-6	2-6	3, 5, 6	4, 9
XB*	4,5,7	2-9	10	4	6,9	10	8	9-10
Kung-Lin	n/a	n/a	8	6	8, 13	n/a	6 ,7	6
cviFFnew	4	5	7	9	4	5,6	7	9

The XB* [14] validity measure is also used in the experiments to measure the c_{opt} of IFC applications. The XB* is originally designed for FCM clustering algorithms. The similarity between the XB* and the new cviFFnew is that, they both assume that the optimum number of clusters would be at their minimum values. However, their compactness and separability measures are different. Thus, the cviFFnew is expected to show an asymptotical behavior towards the larger number of clusters, whereas the XB* validity index can increase or decrease but the c_{opt} should be where the index is minimum. For dataset1 and dataset2, XB* cannot efficiently validate the IFC results for both of the fuzziness values. For larger *c* values XB* can validate the IFC clustering applications to a certain degree. When the system has less models, XB* is not the best cvi to validate the IFC algorithm.

Kung-Lin[15] index, which is originally designed for FCRM, asymptotes to zero and the number of clusters *c* indicate the c_{opt} at the knee-point when it starts to asymptote to zero. From Fig. 1 one can conclude that, Kung and Lin cannot validate the IFC [4] algorithm for smaller number of clusters and is somewhat capable of identifying them when *c's* are larger.

In [14], the inefficiency of XB index [21] for identification of the c_{opt} of the FCM algorithms was proven with examples. We wanted to test if it can identify the c_{opt} for IFC applications. XB index was unable to identify the c_{opt} in most of the datasets.

6 Conclusion

In this paper, a new cvi is proposed for the validation of a previously proposed Improved Fuzzy Clustering (IFC) algorithm [4], which combines two different clustering concepts, e.g., fuzzy c-means clustering and the fuzzy c-regression clustering with fuzzy functions (FFs). Given fuzzy partitions with different input-output relations, the proposed validity index, cviFFnew, computes two different clustering (dis)similarities. The compactness measures the within cluster similarities based on the within cluster distances and the error of the FF model fit, which represents the input-output relations of the fuzzy partition. The separability concept is measured with the between cluster dissimilarities based on the minimum ratio of the cluster center distances to the value of the cosine of the angle between the FFs. The best cluster validity index is obtained by the ratio between the maximum compactness and the minimum separability measures. The new index gradually asymptotes to its minimum after it reaches an elbow, which is the indicator of the optimum number of clusters. The performance of the proposed index was tested on various data sets demonstrating its validity, effectiveness and reliability. The simulation results indicate that for different *m* values, the new validity index was more robust than the rest of the validity indices.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
2. Bezdek, J.C.: Cluster Validity with Fuzzy Sets. J Cbernit **3** (1976) 58-72
3. Bouguessa, M., Wang, S., Sun, H.: An objective approach to cluster validation. Pattern Recognition Letters **27** (2006) 1419-1430
4. Çelikyılmaz, A., Türkşen, I.B: A new Fuzzy System Modeling with Improved Fuzzy Clustering Approach. IEEE Transaction on Fuzzy Systems, under revision (2006)
5. Çelikyılmaz, A., Türkşen, I.B.: Fuzzy Functions with Support Vector Machines. Information Sciences Special Issue, accepted for publication (2007)
6. Chen J-Q., Xi Y-G., Zhang Z-J.: A clustering algorithm for fuzzy model identification. Fuzzy Sets and Systems **98** (1998) 319-329
7. Gunn, S.: Support Vector Machines for Classification and Regression. ISIS Technical Report, University of Southampton (1998)
8. Demirci, M.: Fuzzy functions and their fundamental properties. Fuzzy Sets and Systems **106** (1999) 239-246
9. Dorado, A., Pedrycz, W., Izquierdo, E.: User Driven Fuzzy Clustering: On the Road to Semantic Classification. In: LNCS Rough Sets, Fuzzy Sets, Data Mining Vol. 3641. (2005) 421-430
10. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for the fuzzy c-means method, in Proc. 5th Fuzzy systems Symposium. (1989) 247-250
11. Hathaway, R.J., and Bezdek, J.C.: 1993. Switching regression models and fuzzy clustering. In IEEE Transactions on Fuzzy Systems. **1** (3), 195-203
12. Höppner, F., Klawonn, F.: Improved fuzzy partitions for fuzzy regression models, Int Jrnl of Approximate Reasoning **32** (2003) 85-102
13. Kim, D-W., Lee, K.H., Lee, D.: Fuzzy cluster validation index based on inter-cluster proximity. Pattern Recognition Letters **24** (2003) 2561-74
14. Kim, M., Ramakrishna, R.S.: New indices for cluster validity assessment. Pattern Recognition Letters **26** (2005) 2353-2363
15. Kung, C-C., Lin, C-C: A new Cluster Validity Criterion for Fuzzy C-Regression Model and Its Application to T-S Fuzzy Model Identification, Proc. On IEEE Intern. Conf. on Fuzzy Systems **3** (2004) 1673-78
16. Mitra, S., Banka, H., Pedrycz, W.: Collaborating Rough Clustering. PReMMI 2005, LNSC 3776 (2005) 768-773
17. Pal, N.K., Bezdek, J.C.: On Cluster Validity for Fuzzy C-Means Model. IEEE Trans. Fuzzy Systems **3** (1995), 370-379
18. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modelling and Control. IEEE Trans Syst., Man, Cyberm SMC-15 (1985) 116-132
19. Türk en, I.B.: Fuzzy Functions with LSE. Applied Soft Computing (2006)
20. Türkşen, I.B., Çelikyılmaz, A.: Comparison of Fuzzy Functions with Fuzzy Rule Base Approaches. International Journal of Fuzzy Systems (IJFS) **8** (2006) 137-149
21. Xie, X.L., Beni, G.A.: Validity Measure for Fuzzy Clustering. IEEE Trans on Pattern and Machine Intelligence **3** (1991) 841-846

A Fuzzy Model for Supplier Selection and Development

M.H. Fazel Zarandi[1], Mahmood Rezaei Sadrabadi[2], and I. Burhan Turksen[3]

[1] Department of Industrial Eng., Amirkabir University of Technology, Tehran, Iran
zarandi@aut.ac.ir
[2] Department of Industrial Eng., Iran University of Science & Technology, Tehran, Iran
rezaei@ind.iust.ac.ir
[3] Department of Mechanical and Industrial Eng., University of Toronto, Toronto, Canada
turksen@mie.utoronto.ca

Abstract. In this paper, a fuzzy model is proposed for supplier selection and to identify proper Supplier Development Programs (SDPs). A fuzzy rule base is used to calculate the utility value of a firm's managers to perform a special SDP. Implementation of the model is demonstrated by a numerical example.

Keywords: Fuzzy rule base, supplier development, supplier selection.

1 Introduction

In an attempt to gain greater competitive advantages, firms are shifting away from dealing with greater number of suppliers to forging closer and having more collaborative relationships with a smaller number of them [1], [2]. It should be noted that before initiating cooperation with suppliers, the firms should evaluate them to determine whether a supplier is capable to meet its current and future requirements. Initial supplier evaluation not only helps the firm's managers to adopt the best supplier among the existent candidates, but also would inform the firm's managers about the supplier's situation and adjusting terms of the contract beneficially.

In the literature, various criteria have been introduced for supplier selection such as strategic and operational factors [3], transportation costs and on-time delivery [4], price and service quality [4], [5], and product quality and supplier's profile [5]. Among the proposed models in the literature, fuzzy technologies such as Fuzzy Extended AHP [5] and Fuzzy TOPSIS [6] has been used in some supplier selection models to tackle the different decision criteria.

Although based on supplier selection models a firm can adopt the best suppliers among the existent ones, but the adopted suppliers might not be in a desirable situation from the firms' point of view. Nowadays, firms are increasingly implementing Supplier Development Programs (SDPs) to maintain capable and high performance supply bases [7], [8]. SDPs are defined as activities undertaken by the firms in their efforts to measure and improve the products or services they receive from their suppliers [1].

Krause [9] identified important factors that influence a firm's involvement in SDPs and interrelation of the factors. Krause and Ellram [10] discussed the importance of

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 831–841, 2007.
springerlink.com

SDP and reviewed literature to identify potential critical elements of SDPs. Also, the role of SDPs on the supplier's product and delivery performance, the firm-supplier relationship, and the firm's competitive advantage has been discussed in many case studies [1], [8], [11], [12].

Supplier selection is usually a multi-criterion decision making problem which includes both qualitative and quantitative factors [2], [6]. Thus, neither pure mathematical models and data nor pure conceptual models are appropriate to model the real problem. Since supplier selection problems is usually involved with human judgments and preferences which are often vague and imprecise, fuzzy models are good tools to resolve such decision making problems. In this paper a new fuzzy decision making model is proposed and its construction is discussed in details.

2 The Proposed Model for Supplier Selection and Development

2.1 Effective Parameters on Decision Making

Results of interviews with experts have revealed four important parameters as the effective parameters of their utility to perform SDPs. These are as follows:

1) Result of Supplier Evaluation (M): It is necessary that a firm has an adequate awareness about the suppliers before initiating cooperation with them. By initial evaluation of the suppliers, a firm can ensure about their capabilities to resolve its requirements. Moreover, *M* is a base to identify proper SDPs in the proposed model. Fig. 1 represents criteria and attributes of *M* hierarchically.

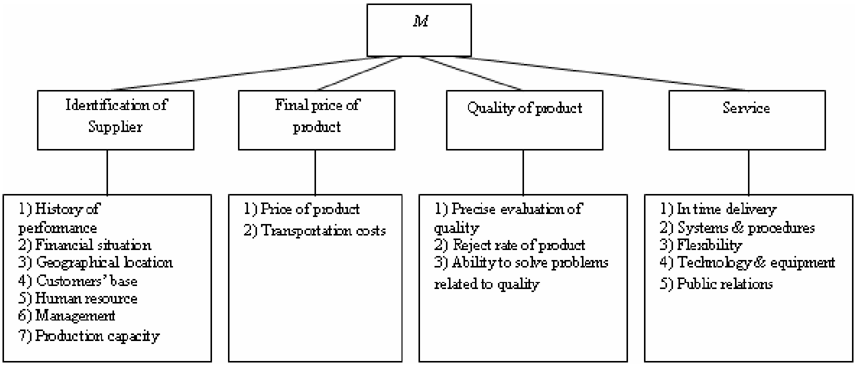


Fig. 1. Criteria and attributes of the parameter *M*

2) Suitable environmental circumstances (E): Environmental circumstances consists of motivating (or preventing) environmental conditions to perform SPDs. Suitability (or unsuitability) of *E* can increase (or decrease) the utility of a firm's managers to perform SDPs. Fig. 2 represents the criteria related to the parameter *E*.

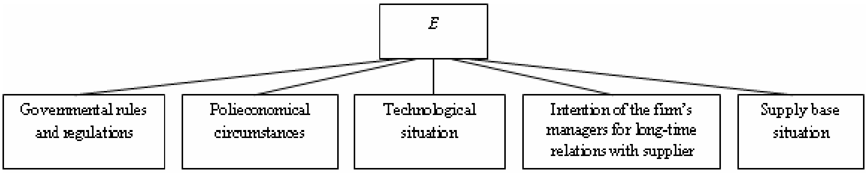

E
Governmental rules and regulations
Polieconomical circumstances
Technological situation
Intention of the firm's managers for long-time relations with supplier
Supply base situation

Fig. 2. Criteria of the parameter *E*

3) Readiness to perform SDP (R): Three candidates are considered to perform a SDP: the firm, the supplier, and a private corporation. Nevertheless, it is supposed that the firm takes on the expenses of any SDP. Generally, readiness of the firm and the supplier are important when the firm or the supplier is performer of a SDP. Also, readiness of the firm, the supplier, and the private corporation are important when a private corporation is performer of a SDP. Unlike *M* and *E*, there is not fixed attributes for *R*. Depending on the performer and the type of the SDP, attributes of *R* and their importance may change. Here, five SDPs are considered: 1) education; 2) consultant; 3) support in purchasing; 4) systems and procedures; 5) technology. Let $R_{ij}\ ; i=1,2,\ldots,5;\ j=1,2,3$ be the total readiness when the *i*th SDP is performed by the *j*th performer, where *j*=1,2,3 mention to the firm, the supplier, and the private corporation, respectively. For each pair *ij* there is a hierarchical diagram that represents criteria and attributes of the total readiness R_{ij}. Fig. 3 represents criteria and attributes of R_{11}, i.e., when the firm is performer of education as the SDP.

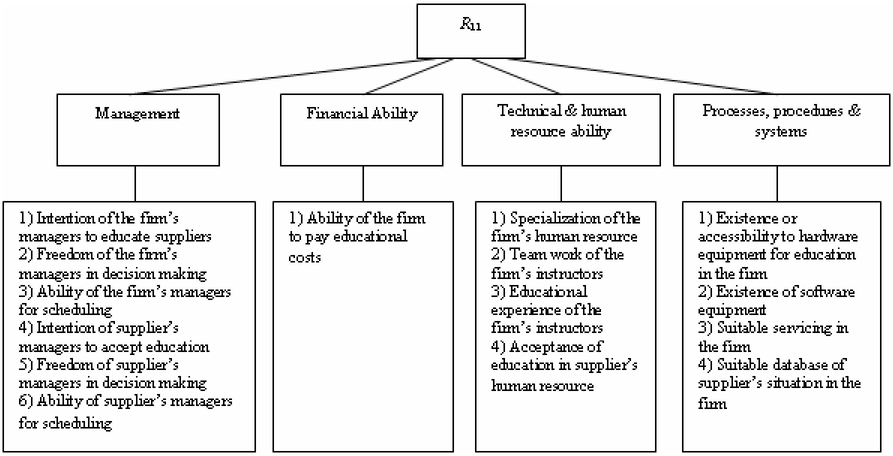

R_{11}
Management
Financial Ability
Technical & human resource ability
Processes, procedures & systems
1) Intention of the firm's managers to educate suppliers
2) Freedom of the firm's managers in decision making
3) Ability of the firm's managers for scheduling
4) Intention of supplier's managers to accept education
5) Freedom of supplier's managers in decision making
6) Ability of supplier's managers for scheduling
1) Ability of the firm to pay educational costs
1) Specialization of the firm's human resource
2) Team work of the firm's instructors
3) Educational experience of the firm's instructors
4) Acceptance of education in supplier's human resource
1) Existence or accessibility to hardware equipment for education in the firm
2) Existence of software equipment
3) Suitable servicing in the firm
4) Suitable database of supplier's situation in the firm

Fig. 3. Criteria and attributes of the parameter R_{11}

4) Rate of return of the SDP (ROR): Rate of return can be the most important parameter to perform a project from managerial point of view. So, before performing a SDP, its economical feasibility should be investigated because a firm's managers usually look at the SDPs as long-time investments. Let $ROR_{ij}\ ; i=1,2,\ldots,5;\ j=1,2,3$ be

ROR of the SDP when the *i*th SDP is performed by the *j*th performer. The calculation of ROR_{ij} is presented in section 3.

2.2 Determination of Potential SDPs

Since a supplier is evaluated by attributes of *M*, the supplier's strengths and weaknesses is revealed via the values of these attributes and the SDPs can be regarded as some programs to improve the supplier's weaknesses. As mentioned previously, five SDPs are considered where each of them can improve some attributes of *M* more or less. We define a parameter which represents the distance between the present status of the supplier and the status of a desirable hypothetical supplier from the firm's managers point of view. This distance is the gap of utility $\Delta\tilde{U}$. Let $\tilde{A}_i^M$ and $\tilde{A}_i^U$ be the measured value and the desirable value of the attribute $\tilde{A}_i$, respectively, where both are fuzzy numbers. Thus, the distance between $\tilde{A}_i^M$ and $\tilde{A}_i^U$ is defined as:

$$\tilde{g}_i = \begin{cases} \tilde{A}_i^U - \tilde{A}_i^M & ; \, if \;\; \pi(\tilde{A}_i^U > \tilde{A}_i^M) = 1 \\ 0 & ; \, otherwise \end{cases} \tag{1}$$

$\pi(\tilde{A}_i^U > \tilde{A}_i^M) = 1$ if and only if there exist elements $a^U \in \tilde{A}_i^U$ and $a^M \in \tilde{A}_i^M$ where $a^U > a^M$ and $\mu_{\tilde{A}_i^U}(a^U) = \mu_{\tilde{A}_i^M}(a^M) = 1$ and π indicates the possibility measure. We say $\tilde{g}_i > 0$ if and only if $\pi(\tilde{A}_i^U > \tilde{A}_i^M) = 1$. Now, we can calculate $\Delta\tilde{U}$ as the weighted average of $\tilde{g}_i$'s according to:

$$\Delta\tilde{U} = \left(\sum_{i=1}^{n} w_i' \tilde{g}_i\right) \otimes \left(\sum_{i=1}^{n} w_i'\right)^{-1} \tag{2}$$

where, w_i' is the modified weight of the *i*th attribute of *M*. In other words, first the attributes with $\tilde{g}_i > 0$ are specified and then the modified weight of the *i*th attribute w_i' is calculated by dividing its initial weight w_i in *M* by sum of the weights of attributes with $\tilde{g}_i > 0$. Therefore, we have:

$$w_i' = w_i \times \left(\sum_{\substack{j=1 \\ \tilde{g}_j > 0}}^{n} w_j\right)^{-1} \tag{3}$$

It should be noted that $w_i' = 0$ for attributes with $\tilde{g}_i = 0$. Portion of the *i*th attribute in the gap of utility $\Delta\tilde{U}$ is $\tilde{d}_i = w_i' \tilde{g}_i$. Thus, we can rewrite (2) as follows:

$$\Delta\tilde{U} = \left(\sum_{i=1}^{n} \tilde{d}_i\right) \otimes \left(\sum_{i=1}^{n} w_i'\right)^{-1} \tag{4}$$

Since $\sum_{i=1}^{n} w_i' = 1$, (4) leads to:

$$\Delta\tilde{U} = \sum_{i=1}^{n} \tilde{d}_i \tag{5}$$

As mentioned before, each SDP can improve some attributes of *M* and therefore can decrease somewhat the amount of $\Delta\tilde{U}$. Now, the ability of each SDP to decrease $\Delta\tilde{U}$ should be determined. Let $\tilde{Z}_k \ ;k=1,2,...,5$ be the ability of the *k*th SDP to decrease $\Delta\tilde{U}$ and $w_{ki} \ ;k=1,2,...,5;i=1,2,...,n$ be the weight of the *k*th SDP to improve the *i*th attribute. Thus:

$$\tilde{Z}_k = \sum_{i=1}^{n} w_{ki}\tilde{d}_i \tag{6}$$

In order to calculate w_{ki}'s, a pairwise comparison matrix is generated for each attribute and $w_{ki};k=1,2,...,5$ are obtained from the matrix. Roughly speaking, we consider the first attribute with $\tilde{g}_i > 0$ and eliminate the SDPs which do have no impact to improve the considered attribute. For the remaining SDPs, a pairwise comparison matrix is generated where the element in the *k*th row and *l*th column $\tilde{v}_{kl}$ shows the ability of the *k*th SDP in contrast to the *l*th SDP to improve the considered attribute and $\tilde{v}_{kl}$ is a fuzzy number. Let S_i be the set of all SDPs which can improve the *i*th attribute and n_i be number of elements in S_i. By presenting a pairwise comparison matrix for each S_i, $W_i = (w_{1i}, w_{2i},..., w_{5i})$ is obtained for the *i*th attribute. Obviously, $w_{ki} = 0$ if the *k*th SDP does not belong to S_i. The values of $\Delta\tilde{U}$ and $\tilde{Z}_k \ ;k=1,2,...,5$ are defuzzified and then the greatest Z_k's are select sequentially until $0.6\Delta U$ is covered. The resultant SDPs are considered for further investigation.

2.3 Final Decision Making

Till now, the values of the attributes of *M* are evaluated based on which the set of potential SDPs are specified for further investigation. Let S_{SDP} be the set of potential SDPs. Based on the values of attributes of *M*, we are able to calculate the value of *M*. Similarly, we can evaluate the values of the criteria of *E* and calculate its value. These two values (*M* and *E*) are used for all SDPs regardless to the performer. However, parameters *R* and *ROR* are depend on the SDP and the performer, so three values for *R* and *ROR* are calculated for each SDP in S_{SDP}, i.e.; R_{i1}, R_{i2}, and R_{i3}, and also ROR_{i1}, ROR_{i2}, and ROR_{i3}. When the *i*th SDP can not be performed by the *j*th performer, $R_{ij} = ROR_{ij} = 0$ is considered. SDPs in S_{SDP} are just a preliminary suggestion and we should investigate them more deliberately. For example, the firm's managers may not intend to develop a supplier when ROR_{ij} or R_{ij} is very low, even though its related SDP exists in S_{SDP}. Generally, each vector (M, E, R_{ij}, ROR_{ij}) causes a utility, U_{ij}, for the firm's managers which shows their tendency to perform the *i*th SDP by the *j*th performer. When $\max\{U_{ij}; j=1,2,3\} > U_T$, the firm's managers may decide to perform the *i*th SDP by the performer which causes the maximum utility among the three ones; U_T is a threshold for the utility. Therefore, we can consider a FRB in which (M, E, R_{ij}, ROR_{ij}) is input vector and U_{ij} is the output.

FRBs can be constructed based on the expertise knowledge or input-output data of the system. Even though the second approach is more precise, it is not applicable for our model, since no numerical data of the decision making system are available. Hence, we use the first approach to construct the required FRB. To this end, the following steps are taken: 1) specification the output and the effective variables on 2) specification of the universe of discourse for all input and output variables 3) covering the universes of discourse by an odd number of fuzzy numbers 4) formation of all combinations of Membership Functions (MFs) for input variables 5) asking the experts to assign a MF out of the output's MFs to each combination. As mentioned before, the output is U_{ij}, utility of the firm's managers to perform the *i*th SDP by the *j*th performer. Also, *M*, *E*, R_{ij}, and ROR_{ij} are considered as the effective variables of the output. Obviously, parameters *M*, *E*, R_{ij}, and U_{ij} are qualitative, so we consider interval [1,9] as their universes of discourse. We cover universes of discourse of parameters *M*, *E*, R_{ij}, by three MFs and U_{ij} by five MFs.

Specifying the universe of discourse for ROR_{ij} needs more attention, because it is a quantitative parameter. Here, we transform ROR_{ij} to a qualitative parameter by a heuristic method in order to conform it to the other parameters. We first determine two points ROR^{-} and ROR^{*} where the firm's managers consider all $ROR_{ij} < ROR^{-}$ as *ROR*'s which are absolutely very low and all $ROR_{ij} > ROR^{*}$ as *ROR*'s which are absolutely very high. ROR^{-} and ROR^{*} are determined based on *MARR* of the firm's managers. For instance, $ROR^{-} = 0.75MARR$ and $ROR^{*} = 1.75MARR$ are admissible. Therefore, we can define a new parameter named $ROR_{ij} / MARR$ instead of ROR_{ij} and cover its universe of discourse by five MFs according to Fig. 4.

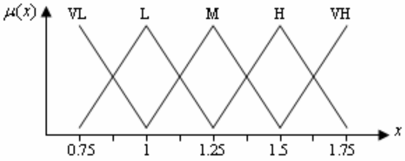


Fig. 4. Linguistic terms for parameter *ROR/MARR*

Regarding to number of MFs for the input variables, the experts determined 109 rules that some of them are as follows:

If *M* is High and *E* is High and *R* is Low and *ROR/MARR* is Very High Then *U* is High
If *M* is High and *E* is Low and *R* is Low and *ROR/MARR* is High Then *U* is Low
If *M* is High and *E* is Medium and *R* is Low and *ROR/MARR* is Medium Then *U* is Medium

Here, we briefly describe implementation of the proposed model. We first ask the firm's managers to evaluate values of the attributes of *M* and also the desirable value for each attribute of M by a fuzzy number. The values of M and also E are calculated by a fuzzy AHP method. Then the attributes with $\tilde{g}_i > 0$ are specified and the value of $\Delta\tilde{U}$ is calculated according to (1)-(6). The firm's managers then are asked to eliminate the SDPs which have no impact to improve the ith attribute of M with $\tilde{g}_i > 0$ and fill in a pairwise comparison matrix for the remaining SDPs in which abilities of the remaining SDPs to improve the ith attribute of M are compared using fuzzy numbers. Based on these pairwise comparison matrices, we are able to calculate $w_{ki}\ ; k = 1,2,...,5; i = 1,2,...,n$, the weight of the kth SDP to improve the ith attribute.

Then, the values $\tilde{Z}_k\ ; k = 1,2,...,5$, ability of the kth SDP to decrease $\Delta\tilde{U}$, are calculated and defuzzified along with $\Delta\tilde{U}$. The greatest Z_k 's are adopted until they cover $0.6\Delta U$. Accordingly, the set of potential SDPs, S_{SDP} , is determined. Regarding to the SDPs in S_{SDP} and each performer, proper attributes for R_{ij} are specified and their values are calculated. Then, for each SDP in S_{SDP} , by considering each performer, cash flows are estimated via fuzzy numbers. Given these fuzzy cash flows, values of ROR_{ij} are obtained. Eventually, for each SDP in S_{SDP} we have a fixed value for M, a fixed value for E, three values for R_{ij} , and three values for ROR_{ij} . Given each vector $(\tilde{M}, \tilde{E}, \tilde{R}_{ij}, R\tilde{O}R_{ij})$, U_{ij} can be calculated from the FRB. Since the estimations of M, E, R_{ij} , and ROR_{ij} as the inputs of the FRB are fuzzy numbers, we use a similarity measure to calculate the degree of matching of each input vector with antecedents of the rules. The similarity between two fuzzy numbers can be calculated as follows:

$$S(\tilde{A}, \tilde{B}) = \frac{\tilde{A} \cap \tilde{B}}{\tilde{A} \cup \tilde{B}} \tag{7}$$

The remaining of the inference procedure is similar to the conventional procedures. Eventually, for each SDP in S_{SDP} three values for U_{ij} are obtained, i.e., U_{i1}, U_{i2}, U_{i3} . Now, a threshold should be determined for the utility, for example $U_T = 7$, and make the final decision for each SDP in S_{SDP} as follows:

If $\max\{U_{ij}; j = 1,2,3\} > U_T$ *then develop the supplier in the ith SDP by performer which causes the maximum utility.*

3 Calculation of the Decision Making Parameters

The values of the parameters $\tilde{M}, \tilde{E}$, and $\tilde{R}_{ij}$ can be calculated based on a fuzzy AHP method. Here, the calculation of $R\tilde{O}R$ is discussed. We consider triangular fuzzy numbers $\tilde{F}_j = (f_j', f_j, f_j'')$ as cash flows. In order to obtain $R\tilde{O}R$, we must solve:

$$\tilde{P} = \sum_{j=1}^{n} \tilde{F}_j \left(\frac{1}{(1+i)^{n_j}}\right) = 0 \tag{8}$$

where, $\tilde{F}_j$ is the fuzzy cash flow at the end of the n_j th year.

(8) shows that $\tilde{P}$ would be a triangular fuzzy number that is represented as:

$$\tilde{P} = \left(\sum_{j=1}^{n} f_j'(\frac{1}{(1+i)^{n_j}}), \sum_{j=1}^{n} f_j(\frac{1}{(1+i)^{n_j}}), \sum_{j=1}^{n} f_j''(\frac{1}{(1+i)^{n_j}}) \right) \tag{9}$$

Given the values of *n*, n_j, f_j', f_j, and f_j'', $\tilde{P}$ is just a function of variable *i*. By using the extension principle and α -levels, different values for *i* are obtained, where, each *i* has a grade of membership equal to $\mu(\alpha)$. In $\alpha = 0$ we have:

$$P_0'(i) = \sum_{j=1}^{n} f_j'(\frac{1}{(1+i)^{n_j}}) = 0 \Rightarrow ROR_0' = i \tag{10}$$

$$P_0''(i) = \sum_{j=1}^{n} f_j''(\frac{1}{(1+i)^{n_j}}) = 0 \Rightarrow ROR_0'' = i \tag{11}$$

and in $\alpha = 1$ we have:

$$P_1(i) = \sum_{j=1}^{n} f_j(\frac{1}{(1+i)^{n_j}}) = 0 \Rightarrow ROR_1 = i \tag{12}$$

Therefore, $R\tilde{O}R = (ROR_0', ROR_1, ROR_0'')$ as a triangular fuzzy number is obtained.

4 Numerical Example

Suppose that based on the pairwise comparison matrix for criteria of *M*, their weights are obtained as follows:

$W_C = (0.04,\ 0.44,\ 0.37,\ 0.15)$

Similarly, the weight vector for attributes of each criterion are:

$W_{C_1} = (0.48,\ 0.17,\ 0.06,\ 0.03,\ 0.10,\ 0.10,\ 0.06)$, $W_{C_2} = (0.90,\ 0.10)$,
$W_{C_3} = (0.25,\ 0.67,\ 0.08)$, $W_{C_4} = (0.39,\ 0.26,\ 0.18,\ 0.15,\ 0.02)$

Table 1 shows the evaluated and the desirable values of *M* and its respective gap.

Table 1. Evaluated and Desirable Values of Attributes of *M*

	C1							C2		C3			C4				
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A^M	H	M	H	VH	VH	M	H	H	H	H	H	VH	H	H	M	M	VH
A^U	M	M	H	H	VH	VH	H	VH	H	H	VH	VH	H	H	M	H	H

According to values of the attributes of *M* and the weights, the value of *M* is $\tilde{M} = (4.96, 6.96, 8.88)$. Similarly, $\tilde{E} = (5.01, 6.99, 8.26)$ is obtained.

As it is observed, the attributes 6, 8, 11, and 16 have a positive gap, so we modify their weights. The initial weights and the modified weights of these attributes are:

$(w_6, w_8, w_{11}, w_{16}) = (.004,.396,.248,.023)$, $(w'_6, w'_8, w'_{11}, w'_{16}) = (.006,.59,.37,.034)$

Now, we can calculate the value of ΔU as follows:

$$\tilde{d}_6 = w'_6 \tilde{g}_6 = .006 \otimes (0,4,6) = (0,.024,.036),$$
$$\tilde{d}_8 = .59 \otimes (-2,2,4) = (-1.18,1.18,2.36)$$
$$\tilde{d}_{11} = .37 \otimes (-2,2,4) = (-.74,.74,1.48), \ \tilde{d}_{16} = .034 \otimes (-2,2,6) = (-.068,.068,.204)$$
$$\Delta\tilde{U} = (-1.988,2.012,4.080), \ \Delta U = defuzz(\Delta\tilde{U}) = 1.762$$

Now, we must generate four pairwise comparison matrices for the attributes with positive gaps, where SDPs are compared to improve the related attribute. However, as mentioned before, we first omit the SDPs which have no impact on the attribute. These matrices end up in the below weight vectors:

$$W_6 = (w_{16}, w_{26}, w_{36}, w_{46}, w_{56}) = (.889,.101,0,0,0)$$
$$W_8 = (w_{18}, w_{28}, w_{38}, w_{48}, w_{58}) = (.461,0,.084,.25,.205)$$
$$W_{11} = (w_{111}, w_{211}, w_{311}, w_{411}, w_{511}) = (.061,.273,.215,.039,.412)$$
$$W_{16} = (w_{116}, w_{216}, w_{316}, w_{416}, w_{516}) = (.695,.189,.116,0,0)$$

Now, we can calculate portion of each SDP to decrease the gap of utility:

$$\tilde{Z}_1 = (-.636,.658,1.352), \tilde{Z}_2 = (-.215,.217,.446), \tilde{Z}_3 = (-.266,.266,.54)$$
$$\tilde{Z}_4 = (-.324,.324,.648), \tilde{Z}_5 = (-.547,.547,1.094)$$
$$Z_1 = defuzz(\tilde{Z}_1) = .581, Z_2 = .19, Z_3 = .231, Z_4 = .279, Z_5 = .472$$

We must have $\Delta U = \sum_{k=1}^{5} Z_k$ because:

$$\Delta U = \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} (\sum_{k=1}^{5} w_{ki}) d_i = \sum_{k=1}^{5} (\sum_{i=1}^{n} w_{ki} d_i) = \sum_{k=1}^{5} Z_k$$

However, as is seen $\Delta U = 1.762$ and $\sum_{k=1}^{5} Z_k = 1.753$. The difference between these two values is derived from computational approximations. So, by considering the common value $\Delta U = \sum_{k=1}^{5} Z_k = 1.753$, the first and the fifth SDPs are selected for further investigation, since $Z_1 + Z_5 = 1.053 > 1.052 = 0.6\Delta U$.

For each SDP three values are calculated for *R* and *ROR* as follows:

$$\tilde{R}_{11} = (4.89,6.89,8.43), \tilde{R}_{12} = 0, \tilde{R}_{13} = (5.47,7.47,8.74)$$
$$\tilde{R}_{51} = (4.52,6.52,8.24), \tilde{R}_{52} = (4.78,6.78,8.45), \ \tilde{R}_{53} = (5.34,7.34,8.56)$$
$$R\tilde{O}R_{11} = (.3,.31,.34), R\tilde{O}R_{12} = 0, R\tilde{O}R_{13} = (.21,.22,.26)$$
$$R\tilde{O}R_{51} = R\tilde{O}R_{52} = R\tilde{O}R_{53} = (.31,.32,.35)$$

By considering $MARR = 0.2$, we would have six different input vectors as follows:

$$(\tilde{M}, \tilde{E}, \tilde{R}_{11}, \widetilde{ROR}_{11} / MARR) = ((4.96,6.96,8.88), (5.01,6.99,8.26), (4.89,6.89,8.43), (1.5,1.55,1.7))$$

$$(\tilde{M}, \tilde{E}, \tilde{R}_{12}, \widetilde{ROR}_{12} / MARR) = ((4.96,6.96,8.88), (5.01,6.99,8.26),0,0)$$

$$(\tilde{M}, \tilde{E}, \tilde{R}_{13}, \widetilde{ROR}_{13} / MARR) = ((4.96,6.96,8.88), (5.01,6.99,8.26), (5.47,7.47,8.74), (1.05,1.1,1.3))$$

$$(\tilde{M}, \tilde{E}, \tilde{R}_{51}, \widetilde{ROR}_{51} / MARR) = ((4.96,6.96,8.88), (5.01,6.99,8.26), (4.52,6.52,8.24), (1.55,1.6,1.75))$$

$$(\tilde{M}, \tilde{E}, \tilde{R}_{52}, \widetilde{ROR}_{52} / MARR) = ((4.96,6.96,8.88), (5.01,6.99,8.26), (4.78,6.78,8.45), (1.55,1.6,1.75))$$

$$(\tilde{M}, \tilde{E}, \tilde{R}_{53}, \widetilde{ROR}_{53} / MARR) = ((4.96,6.96,8.88), (5.01,6.99,8.26), (5.34,7.34,8.56), (1.55,1.6,1.75))$$

By the above inputs and given the FRB, we would have:

$$U_{11} = 7.16 \,,\; U_{12} = 0 \,, U_{13} = 5.16 \,,\; U_{51} = 7.16 \,,\; U_{52} = 7.16 \,,\; U_{53} = 7.28$$

$$\max\{U_{11}, U_{12}, U_{13}\} = U_{11} = 7.16 \,,\; \max\{U_{51}, U_{52}, U_{53}\} = U_{53} = 7.28$$

By considering $U_T = 7$, firm and private corporation are selected as performers of the first and the fifth SDP, respectively.

5 Conclusion

In this paper, a fuzzy decision making model have been proposed for supplier selection and development. In the proposed model, some relations have been used to specify the potential SDPs. Then they have been further investigated through a FRB in which *M*, *E*, *R*, and *ROR* are considered as input variables and the utility of the firm's managers to perform the specified SDP as the output variable. Results of implementation of the proposed model show high adaptability of the model with the experts' opinions and can be used as a decision support system to contribute the experts in the decision making process.

References

1. Prahinksi, C., Benton, W. C.: Supplier Evaluation: Communication Strategies to Improve Supplier Performance. Journal of Operations Management. 22(1) (2004) 39–62
2. Bevilacqua, M., Ciarapica, E. F., Giacchetta, G.: A Fuzzy-QFD Approach to Supplier Selection. Journal of Purchasing and Supply Management. 12 (2006) 14–27
3. Talluri, S., Narasimhan, R.: A Methodology for Strategic Sourcing. European Journal of Operational Research. 154(1) (2004) 236–250
4. Germain, R., Droge, C.: Wholesale Operations and Vendor Evaluation. Journal of Business Research. 21(2) (1990) 119–129
5. Chan, F.T., Kumar, N.: Global Supplier Development Considering Risk Factors Using Fuzzy Extended AHP-Based Approach. The Int. J. of Man. Sci. (2006), Article in Press
6. Chen, C. T., Lin, C. T., Huang, S. F.: A Fuzzy Approach for Supplier Evaluation and Selection in Supply Chain Management. Int. J. of Prod. Economics. 102 (2006) 289–301
7. Modi, S. B., Mabert, V. A.: Supplier Development: Improving Supplier Performance through Knowledge Transfer. J. of Operations Management. xx (2006) xxx–xxx, Article in Press

8. Wagner, S. M.: A Firm's Responses to Deficient Suppliers and Competitive Advantage. Journal of Business Research. 59 (2006) 686–695
9. Krause, D. R.: The Antecedents of Buying Firms' Efforts to Improve Suppliers. Journal of Operations Management. 17(2) (1999) 205–224
10. Krause, D.R., Ellram, L.M.: Critical Elements in Supplier Development. European Journal of Purchasing & Supply Management. 3(1) (1997) 21–31
11. Li, L.W., Humphreys, P., Chan, L.Y., Kumar., M.: Predicting Purchasing Performance: The Role of Supplier Development Programs. J. of Mat. Proc. Tech. 138 (2003) 243–249
12. De Toni, A., Nassimbeni, G.: Just-in-time Purchasing: An Empirical Study of Operational Practices, Supplier Development and Performance. Omega. 28(6) (2000) 631–651

A Neuro-fuzzy Multi-objective Design of Shewhart Control Charts

Mohammad Hossein Fazel Zarandi[1], Adel Alaeddini[2], I. Burhan Turksen[3], and Mahdi Ghazanfari[4]

[1] Department of Industrial Engineering, AmirKabir University of Technology, Tehran, Iran, P.o.Box:15875-13144
zarandi@aut.ac.ir
[2] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran, P.o.Box:16846-13114
adel@ind.iust.ac.ir
[3] Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada, P.o.Box:M5S2H8
turksen@mie.utoronto.ca
[4] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran, P.o.Box:16846-13114
mehdi@iust.ac.ir

Abstract. Control charts are widely used to establish statistical process control (SPC) and maintain manufacturing processes in desired operating conditions. Effective use of a control chart requires finding the best parameters for the optimal operation of the chart, which is called design of control chart. The design of control chart involves the selection of three parameters namely, the sample size, the sampling interval, and the control limits coefficients. Conventional approaches to design control charts include complex mathematical models and associated optimization schemes. Also, Conventional approaches usually use point estimators of the input parameters which are not able to represent the true parameters sufficiently. The input parameters are suffered from ambiguity uncertainty, which can be effectively modeled by using fuzzy set theory. In this paper a fuzzy multi-objective model for economic-statistical design of X-bar control chart, in which the input parameters are expressed by fuzzy membership functions, is proposed. Moreover, an ANFIS approach is used to transform the complex fuzzy multi-objective model into a fuzzy rule-base. Next, a genetic algorithm (GA) has been developed to find the best design (input) parameters of control charts. Finally, the proposed approach is tested and validated by using a numerical example.

Keywords: Statistical Process Control, Control Charts Design, Fuzzy Set Theory, ANFIS, Genetic Algorithm.

1 Introduction

Control Charts are one of the most widely used statistical process control (SPC) tools. They are based on a graphical display of data series versus sample number or time. The use of control charts is based on determination of some parameters such as the sample size, the sampling interval and the control limits coefficients that are usually

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 842–852, 2007.
springerlink.com

called the design of the control charts. Traditionally, control charts have been designed with respect to statistical criteria and practical experiences. The main objective of these criteria is fast detection of particular shifts in the quality characteristic of a process. This usually involves selecting the sample size and control limits such that the average run length of the chart to detect a particular shift in the quality characteristic and the average run length of the procedure when the process is in-control are equal to specified values (Montgomery [1]). In addition to the statistical aspects, the design of the control charts has economic consequences, such as sampling and testing costs, investigating and correcting assignable cause costs, poor quality costs, false alarm costs, etc. Therefore, in recent years, the economic design of control charts has been the main concern of many researchers. The earliest use of economic modelling for determination of statistical control procedures refers to the work of Girshick and Rubin [2]. Later, Duncan [3] proposed an economic model for determining the parameters of control charts. Woodall [4] criticized the economic design of control charts for its poor statistical performance. Saniga [5] proposed a multi-objective optimization model for joint economical statistical design of $\overline{X}$ and R charts. Montgomery et al. [6] proposed a similar constrained model for the design of EWMA[1] charts. Del Castillo et al. [7] studied semi economical design of control charts. They proposed a multi-objective model for the design of charts. Their approach was followed by Del Castillo [8]. Celano and Fichera [9] proposed an evolutionary algorithm, to solve control chart design problems considering the optimization of cost of a chart and at the same time the statistical properties. Chen and Liao [10] formulated the optimal design of control charts as a multi-criteria decision-making (MCDM) problem. They also proposed a solution procedure on the basis of data envelopment analysis to solve the MCDM problem. Pignatiello and Tsai [11] discussed the robust design of control chart when cost and process parameters are not precisely known. Vommi and Seetala [12] presented a risk-based approach which the input parameters were expressed as ranges. They also used Genetic algorithm as a search tool to find the best design (input) parameters with which the control chart had to be designed. Linderman and Choo [13] discussed the robust economic design of charts when a single process assumes different scenarios. Vommi and Seetala [14] suggested a design approach for the robust economic design problem of a process with multiple scenarios which used simple genetic algorithm for the aim of optimization. Saniga and Shirlan [15] reported very few applications of economic and statistical-economic design of control chart. There are at least two reasons for the lack of practical implementation of this methodology: First, complex mathematical model and optimization schemes and, second, difficulty in estimation of the costs and other parameters of the model.

This research presents two main contributions: first, adding fuzzy uncertainty to the economic-statistical design of control charts and second, presenting a new approach in solving the fuzzy multi-objective model by transforming it to a fuzzy rule-base through an ANFIS[2] and using genetic algorithm as a search tool to find the input values which yield optimal output value in the generated fuzzy rule-base. The rest of the paper is organized as follows: Section 2 introduces the basic notations required for the model. Section 3 explains the process of development of the proposed model. In

[1] Exponentially Weighted Moving Average.

[2] Adaptive Neuro-Fuzzy Inference System.

section 4 the proposed optimization method is presented. Section 5 provides a numerical example to validate the proposed method. Finally, conclusions and further works are appeared in Section 6.

2 Notations

a_1	Fixed component of sampling cost
a_2	Variable component of sampling cost
a_3	Cost of finding and fixing assignable cause
a'_3	Cost of investigating false alarms
a_4	Hourly penalty cost associated with the out of control state
λ_i	Process failure rate for shift of kind i
λ	Total process failure rate
v	Hourly penalty cost associated with production in the out-of-control state
s_i	Shift size for shift of kind i
g	Time to test and interpret the result per sample unit
d	Time to search and fix process
μ	Mean of the process
σ	Standard deviation of the process
p_l	Constraint on Type II error
a_u	Constraint on Type I error
V_0	Net income per hour of operation in the in-control state
V_1	Net income per hour of operation in the out-of-control state
α	Type I error
β	Type II error
p	Detection power of the chart
$\phi(z)$	Density Function of Standard Normal Distribution
n	Sample Size
h	Sampling Interval
k	Control limits coefficient
ARL_0	Average Run Length to in-control condition
ARL_1	Average Run Length to out-of-control condition

3 The Proposed Model

The proposed model consists of three fuzzy economic-statistical objective functions, two constraints and three decision variables. The first objective functions is for maximization of in-control ARL which is equivalent to minimization of the type I error or, to reduce false alarms of the control charts. The second objective function can minimize out-of-control ARL which is equivalent to minimizing type II error or speed of detecting assignable causes. Finally, the third objective function minimizes the expected loss per hour incurred by the process or equivalently maximizes the expected net income per hour. The constraints of the model limit probability of the type I error, and the detection power of the chart, according to the pre-selected upper and lower bounds.

3.1 Crisp Representation of the Components of the Model

The First Objective Function. The first objective function minimizes the false alarm rate of the chart, ARL_0, or equivalently, maximizes the in-control average run length:

$$\alpha = 2\int_{-\infty}^{-k} \phi(z)dz \,. \tag{1}$$

$$ARL_0 = \frac{1}{\alpha} = \frac{1}{2\int_{-\infty}^{-k} \phi(z)dz} \,. \tag{2}$$

The Second Objective Function. The second objective function maximizes the detection power of the chart against different shifts, or equivalently, minimizes the out-of-control average run length, ARL_1:

$$ARL_1 = \frac{1}{1-\beta} = \frac{1}{\int_{-\infty}^{-k-s.\sqrt{n}} \phi(z)dz + \int_{k-s\sqrt{n}}^{\infty} \phi(z)dz} \,. \tag{3}$$

$$1-\beta = \int_{-\infty}^{-k-s.\sqrt{n}} \phi(z)dz + \int_{k-s\sqrt{n}}^{\infty} \phi(z)dz \,. \tag{4}$$

The third objective function. The third objective function which is an extension to Duncan's [3] model, minimizes, expected cost per unit time $E(A)$:

$$E(A) = \frac{E(C)}{E(T)} \,. \tag{5}$$

where, $E(C)$ is expected total cost incurred during a cycle, and $E(T)$ is expected length of a cycle. $E(T)$, consists of four periods: (1) The expected in-control period, (2) The expected time to signal, (3) The expected time to sample and interpret the results, and (4) the expected time to find the assignable cause. The length of in-control period follows an exponential distribution with mean of $1/\lambda$.The expected time to signal is equal to expected number of samples to recognize the shift time between samples minus expected least in-control time between consecutive samples in the cycle. The number of samples required producing a true out-of -control signal is a geometric random variable with the mean $1/(1-\beta)$. Given the occurrence of an assignable cause between j th to $(j+1)$ th samples, the expected least in-control time within this interval is:

$$\tau = \frac{\int_{jh}^{(j+1)h} e^{-\lambda t}\lambda(t-jh)dt}{\int_{jh}^{(j+1)h} e^{-\lambda t}\lambda dt} = \frac{1-(1+\lambda h)e^{\lambda h}}{\lambda\left(1-e^{\lambda h}\right)}. \tag{6}$$

Therefore, the expected time to signal is: $h/(1-\beta)-\tau$. The expected time to sample and interpret each unit of sample which consists of n observations is g. Hence, the expected time to sample and interpret the results is: $g.n$. Finally, the expected time to find the assignable cause is d. Therefore, the expected length of a cycle is:

$$E(T) = \frac{1}{\lambda} + \frac{h}{1-\beta} - \tau + g.n + d. \tag{7}$$

$E(C)$, expected total cost incurred during a cycle consists of five parts: (1) The expected income during in-control state period, (2) The expected income during out-of-control state period, (3) The expected cost of sampling, (4) The expected cost of false alarms, and (5) The expected cost of repairs. The profit per hour during the in-control period is V_0, and the expected time of in-control period is $1/\lambda$. Hence, the expected income during in-control state period is: $V_0(1/\lambda)$. The profit per hour, during the out-of-control period is V_1, and the expected time of out-of-control period is: $\frac{1}{\lambda}+\frac{h}{1-\beta}-\tau+g.n+d$. Therefore, the expected income during out-of-control state period is:

$$V_1.\left(\frac{1}{\lambda} + \frac{h}{1-\beta} - \tau + g.n + d\right). \tag{8}$$

The expected cost of sampling is the product of expected sampling cost per sample and the expected number of samples in a cycle. The expected cost per sample consists of fixed and variable components: $a_1 + a_2 n$. The expected number of sample per cycle is also $E(T)/h$. Consequently, the expected cost of sampling per cycle is: $(a_1 + a_2 n)E(T)/h$. The expected cost of false alarms is the product of the expected number of false alarms per cycle and the cost of a false alarm. The expected number of false alarms per cycle can be calculated as the product of probability of false alarms and the expected number of samples taken before the process goes out of-control. The expected number of samples taken before the process goes out-of-control is: $e^{-\lambda h}/1-e^{-\lambda h}$. Then, the expected number of false alarms per cycle is: $\alpha.e^{-\lambda h}/1-e^{-\lambda h}$. The cost of investigating a false alarm is a_3'. Accordingly, the expected cost of false alarms is: $\alpha.a_3'.e^{-\lambda h}/1-e^{-\lambda h}$. Finally the expected cost of finding an assignable cause is a_3. Therefore, the expected net income per cycle can be calculated as follows:

$$E(C) = V_0\frac{1}{\lambda} + V_1.\left(\frac{h}{1-\beta} - \tau + g.n + d\right) - a_3 - \frac{\alpha.a'_3.e^{-\lambda h}}{1-e^{-\lambda h}} - (a_1 + a_2 n)\frac{E(T)}{h}. \quad (9)$$

The expected net income per hour is calculated by dividing the expected net income per cycle by the expected cycle length, resulting in:

$$E(A) = \frac{V_0(1/\lambda) + V_1.(h/(1-\beta) - \tau + g.n + d) - a_3 - \alpha.a'_3.e^{-\lambda h}/(1-e^{-\lambda h})}{1/\lambda + h/(1-\beta) - \tau + g.n + d} - \frac{(a_1 + a_2 n)}{h}. \quad (10)$$

substituting $a_4 = V_0 - V_1$ in (10), where a_4 is the hourly penalty cost associated with production in the out-of- control state, $E(A)$ can be rearranged as:

$$E(A) = V_0 - \frac{(a_1 + a_2 n)}{h} - \frac{a_4(h/(1-\beta) - \tau + g.n + d) + a_3 + \alpha.a'_3.e^{-\lambda h}/(1-e^{-\lambda h})}{1/\lambda + h/(1-\beta) - \tau + g.n + d}. \quad (11)$$

Instead of using a constant cost of poor quality against different shift sizes, we propose Taguchi [18] cost function for a_4, which calculates cost of poor quality based on squared distance of quality characteristic from the target as follows:

$$a_4 = \sum_i v.s_i^2.\frac{\lambda_i}{\sum_i \lambda_i} \quad . \quad (12)$$

Hence, $E(A)$ can also be rewritten as:

$$E(A) = \frac{E(C)}{E(T)} = V_0 - E(L). \quad (13)$$

In (13), V_0 has not any contribution in the optimization process, so, it can be removed from the objective function. Hence, $E(L)$ can be rewritten as:

$$E(L) = \frac{(a_1 + a_2 n)}{h} + \frac{a_4(h/(1-\beta) - \tau + g.n + d) + a_3 + \alpha.a'_3.e^{-\lambda h}/(1-e^{-\lambda h})}{1/\lambda + h/(1-\beta) - \tau + g.n + d}. \quad (14)$$

Model Constraints. The proposed model includes two statistical constraints borrowed from Saniga [6]. The first constraint put an upper bound to the type I error. The second constraint put a lower bound to the detection power of the chart:

$$\begin{gathered} \alpha \le a_u \\ p \ge p_l \end{gathered} \quad (15)$$

3.2 Fuzzy Representation of the Components the Model

As mentioned previously, Saniga and Shirlan [16] reported very few implementation of economic model for the design of control charts. One of the main bottlenecks of the practical implementation of the control charts design is a lack of knowledge about the values of the optimization model parameters. These values are seldom known by users, and there exist, at least, two reasons for such a situation. First, they may randomly vary in time. In such a case, in the optimization procedure, we may use their average values, as it is frequently used in practice. The second reason is that the input parameters and their related values are usually defined imprecisely. In such cases, fuzzy set theory is a powerful tool to resolve these main problems. Here, imprecise information about the parameters of the model is presented by fuzzy numbers. Hence, The fuzzy equivalents of the input parameters a_1, a_2, a_3, a'_3, v, g, d, s are represented by fuzzy sets $\tilde{a}_1$, $\tilde{a}_2$, $\tilde{a}_3$, $\tilde{a}'_3$, $\tilde{v}$, $\tilde{g}$, $\tilde{d}$, $\tilde{s}$. Substitutions of crisp parameters with fuzzy parameters change the second and third objective functions as follows:

$$ARL_1 = \frac{1}{1-\beta} = \frac{1}{\int_{-\infty}^{-k-\tilde{s}.\sqrt{n}} \phi(z)dz + \int_{k-\tilde{s}\sqrt{n}}^{\infty} \phi(z)dz} . \tag{16}$$

$$E(L) = \frac{(\tilde{a} + \tilde{a}_2 n)}{h} - \frac{\tilde{a}_4 \left(h/(1-\beta) - \tau + \tilde{g}.n + \tilde{d} \right) + \tilde{a}_3 + \alpha.\tilde{a}'_3.e^{-\lambda h} / \left(1 - e^{-\lambda h}\right)}{1/\lambda + h/(1-\beta) - \tau + \tilde{g}.n + \tilde{d}} . \tag{17}$$

The final structure of the model could be summarized as follows:

$$\begin{aligned} & MAX\ ARL_0 \\ & MIN\ ARL_1 \\ & MIN\ E(L) \\ & s.t. \\ & \alpha \le a_u \\ & p \ge p_l \qquad \forall\, Decision\ Variables\,(n, h, k) \end{aligned} \tag{18}$$

4 Optimization Procedure

The solution procedure of the above problem consists of three steps: aggregating objective functions into single objective function problem, transforming the model to a fuzzy rule-base, and finding input values which produce optimal output value in the generated rule-base.

4.1 Objective Functions Aggregation

To simplify the solution procedure of the above problem three objective functions of the model can be aggregated into a single objective problem by the use of MODM[3] methods. For this purpose, objective functions should be ranked and dimension-lessened before aggregation. Then, the weighted sum of objective functions which is widely used in MODM can be used to transform the above multi objectives problem to a single objective problem as follows:

[3] Multiple objective decision making.

$$\min \sum_{j=1}^{k} \omega_j . f_j(x) \quad j = 1,2,\cdots,m \tag{19}$$

where, $f_j(x)$ is the jth objective function and ω_j is the rank of jth objective function.

4.2 Transformation of the Model into Fuzzy Rule-Base

Based on the proposed method, twelve input parameters are needed to model control charts execution in a process, nine of which include fuzzy uncertainty. Thus, the complexity of the model is very high. In this paper, ANFIS methodology has been implemented to come up with the complexity of the above complex multi-objective model and to transfer it into a fuzzy rule-base format. The inputs of the fuzzy rule-base are the decision variables of the optimization model, and the output of the fuzzy rule-base is the weighted sum of objective functions. Thus, every feasible solution of the MODM problem is equivalent to an input vector of the fuzzy rule-base, and the optimal solution of the MODM problem is equivalent to a input vector which produce optimal output value in the generated rule-base. To develop such a system using ANFIS, requires a large number of input-output data and training procedure which would be explained in details in section 5.

4.3 Rule-Base optimization

In this research, genetic algorithm is used as a search tool to find rule-base inputs in the premises part which generate optimal outputs value in the consequent part. For this purpose, the algorithm generates a population of input values at each generation, and examines the outputs of them through the rule-base. Here, inferential mechanism of the rule-base is used as the fitness function of the evolutionary algorithms. Of each generation the input values that produce the least output values are nominated for participation in the next generation until the optimal solution is found.

5 Numerical Example

In this section, the proposed model is tested and validated by a numerical example. Since there is not any investigated problem under different shifts and uncertain parameters in the literature, the proposed method has been tested by using a modified example of Montgomery [1].

Without loosing of generality, it is assumed that process runs under normal distribution with $\mu = 0$ and $\sigma = 1$. It can be also assumed that other parameters of the above process take values as: $\lambda = .05$, $\tilde{a}_1 = (0.5,1,1.5)$, $\tilde{a}_2 = (0.1,0.2,0.3)$, $\tilde{a}_3 = (10,25,35)$, $\tilde{a}_3' = (35,50,60)$, $\tilde{d} = (1,2,3)$, $\tilde{v} = (90,100,110)$, $\tilde{s} = (0,2,3)$, $\tilde{g} = (0.05,0.1,.15)$. These values are an extent to Montgomery [1] original example.

To provide sufficient data for rule-base generation through ANFIS a large amount of input-output data is needed, which is generated by simulations as follows:

The first part of each inputs-output generated data consists of a triple vector (n, h, k) produced from possible values of n, k, h, and in order to limit the scope of the solutions to practical applications, input values are forced to get values within range: $2 \le n \le 20$, $0.1 \le h \le 1$, $2.5 \le k \le 3.5$.

The input vectors should be put into the three objective functions of the main model to produce three output values: $E(L), ARL_0, ARL_1$. Then, these quantities have to be dimension lessened, weighted and summed in order to produce the output part of generated input-output data. In the investigated example, the output values $E(L), ARL_0, ARL_1$ are weighted by the vector $(1.3,1,0.5)$after getting dimension less. It should be noticed that in real world problems the elements of this vectors are determined based on the process properties and preferences.

Based on above procedure 10,450 input-output data have been generated for the investigated problem. By then, with regards to two constraints of the main model, non feasible data should be omitted before manipulating by the ANFIS. For this example, the right hand sides the constraints have been chosen as: $\alpha \le 0.01$, $p \ge 0.9$ which are common in Statistical Process Control. After filtering the data through above constraints, 5,737 of 10,450 inputs-output data have been selected for training process.

As a result a 5,737×4 matrix of generated Input-output data (5,737 input-output data consist of 3 inputs and 1 out-put) have been generated to feed into the ANFIS. To develop the rule-base through the ANFIS several types of membership functions and training methods have been investigate. Finally, Grid partition method and triangular fuzzy number has been used for rule generation. The elected ANFIS has been trained under 60 epochs with an error tolerance of 0.06778 which has been the least error in all of the methods which has been investigated. The application of the elected ANFIS led to 27, three input-one output Sugeno-type rules that have been able to simulate the behavior of the main multi-objective model precisely (See Figs. 1- 3 in the Appendix).

The final step of the proposed approach includes finding the best input vector which yields the optimal output value. For this case, a genetic algorithm has been used. A few sets of parameters have been investigated to tune the search algorithm.At last, Following parameters have been chosen to optimize generated fuzzy rule-base; population size=20, Crossover fraction=0.8, Mutation function=Gausssian with scale and shrink parameters=1, Selection function=stochastic uniform, Generations=200. The algorithm reached to solution (10.798, 3.4996, 0.57749) with total fitness 0.68165 (See Fig. 2. in Appendix).

6 Conclusions

In this paper, a fuzzy economic-statistical design methodology for $\overline{X}$ control chart has been presented. The design simultaneously takes into consideration the uncertainty of different shift sizes and input parameters which hasn't been practiced by any of the available methods in literature. In this regard, an aggregated Adaptive neuro-fuzzy system(ANFIS), as a data modeling tool, fuzzy set theory, as an imprecision modeling tool, and Genetic algorithm, as an efficient search tool for finding the best design parameter set has been developed. The method has also takes into account the statistical constraints in the design, which is absolutely essential in the control chart design. Finally, the proposed approach has been tested and validated with a numerical example. The results show the applicability and robustness of the system in the complex control chart design.

References

1. Montgomery D.C..: Introduction to statistical quality control. 5th, Wiley, New York, NY. (2005)
2. Girshick MA, Rubin H A.: baysian to a quality control model. Vol. 23(1). Annals of Mathematical Statistics, (1952) 114-125.
3. Duncan AJ.: The economic design of _X charts to maintain current control of a process. Vol. 51. Journal of the American Statistical Association, (1956) 228–242.
4. Lorenzen TJ, Vance LC.: The economic design of control charts: a unified approach. Vol.28. Technometrics (1986) 3–10.
5. Woodall WH.: Weaknesses of the economic designof control charts. Vol.28 Technometrics. (1986) 408–409.
6. Saniga EM: Economic Statistical Control Chart Designs with an Application to X and R Charts. Vol.31. Technometrics. (1989) 313–320.
7. Montgomery DC, Torng JCC, Cochran JK, Lawrence FP.: Statistically constrained economic design of the ewma control chart. Vol.27 Journal of Quality Technology. (1995) 250–256.
8. Del Castillo E: A multivariate self-tuning controller for run-to-run process control under shift and trend disturbances. Vol.28. IIE Transactions. (1996) 1011–1021.
9. Del Castillo E, Mackin P, Montgomery D.C.: Multiple-criteria optimal design of x-bar control charts. Vol 28. IIE Transactions. (1996) 467-474.
10. Celano G, Fichera S.: Multiobjective economic design of an x control chart. Vol.37. Computers & Industrial Engineering.(1999) 129-132
11. Chena YK, Liaob HC.: Multi-criteria design of an X-bar control chart. Vol.46. Computers & Industrial Engineering (2004) 877–891
12. Pignatiello JJ Jr, Tsai.: A Optimal economic design of control charts when cost model parameters are not precisely known. Vol.20.IIE Transactions. (1988)103–10.
13. Vommi VB, Seetala MSN.: A new approach to robust economic design of control charts. Applied Soft Computing. (2006) ARITCLE IN PRESS
14. Linderman K, Choo AS. Robust economic control chart design. Vol.34. IIE Transactions. (2002)1069–78.
15. Vommi VB, Seetala MSN.: A simple approach for robust economic design of control charts. Computers & Operations Research. (2006) ARITCLE IN PRESS
16. Saniga EM, Shirland LE: Quality control in practice-a survey. Vol. 10(5) Quality Progress. (1977) 30–33.
17. Keats JB, Castillo ED, Collani EV, Saniga EM.: Economic modeling for statistical process control, Vol. 29 (2). J. Qual. Technol. (1997)144–147.
18. Taguchi G.: Introduction to Quality Engineering. Asian Productivity Organization, UNIPUB, White Plains, NY. (1986).

Appendix

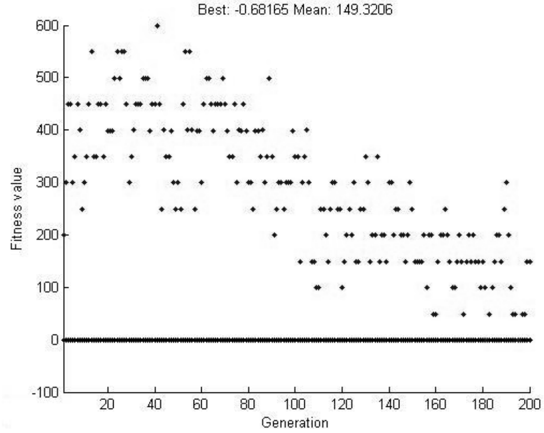
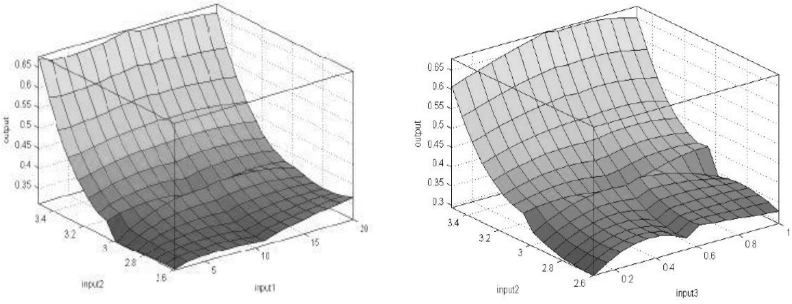
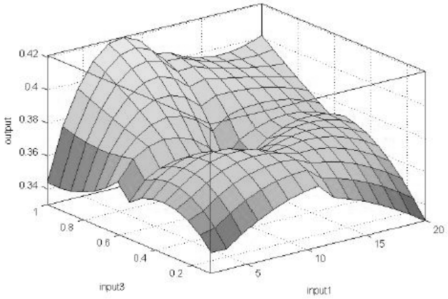


Fig. 1. The fuzzy rule-base output of elected ANFIS with triangular input membership functions an linear output membership functions containing triple inputs and single output with 27 rules and 0.06778 error



(a) (b)

Fig. 2. Genetic Algorithm Fitness values versus generation



(c)

Fig. 3. Fuzzy Rule base output surface Versus: (a) inputs 1,2 (b)inputs 2,3 (c)inputs 3,1

Author Index