ADVANCES IN SOFT COMPUTING

Ashutosh Tiwari · Joshua Knowles
Erel Avineri · Keshav Dahal · Rajkumar Roy
Editors

Applications of Soft Computing

Recent Trends

Springer

Ashutosh Tiwari, Joshua Knowles, Erel Avineri, Keshav Dahal, Rajkumar Roy (Eds.)

Applications of Soft Computing

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

Mieczysław Kłopotek, Maciej Michalewicz and Sławomir T. Wierzchoń (Eds.)
Intelligent Information Systems 2002, 2002
ISBN 3-7908-1509-8

Andrea Bonarini, Francesco Masulli and Gabriella Pasi (Eds.)
Soft Computing Applications, 2002
ISBN 3-7908-1544-6

Leszek Rutkowski, Janusz Kacprzyk (Eds.)
Neural Networks and Soft Computing, 2003
ISBN 3-7908-0005-8

Jürgen Franke, Gholamreza Nakhaeizadeh, Ingrid Renz (Eds.)
Text Mining, 2003
ISBN 3-7908-0041-4

Tetsuzo Tanino, Tamaki Tanaka, Masahiro Inuiguchi
Multi-Objective Programming and Goal Programming, 2003
ISBN 3-540-00653-2

Mieczysław Kłopotek, Sławomir T. Wierzchoń, Krzysztof Trojanowski (Eds.)
Intelligent Information Processing and Web Mining, 2003
ISBN 3-540-00843-8

Ajith Abraham, Katrin Franke, Mario Köppen (Eds.)
Intelligent Systems Design and Applications, 2003
ISBN 3-540-40426-0

Ahmad Lotfi, Jonathan M. Garibaldi (Eds.)
Applications and Science in Soft-Computing, 2004
ISBN 3-540-40856-8

Mieczysław Kłopotek, Sławomir T. Wierzchoń, Krzysztof Trojanowski (Eds.)
Intelligent Information Processing and Web Mining, 2004
ISBN 3-540-21331-7

Miguel López-Díaz, Maríaç. Gil, Przemysław Grzegorzewski, Olgierd Hryniewicz, Jonathan Lawry
Soft Methodology and Random Information Systems, 2004
ISBN 3-540-22264-2

Kwang H. Lee
First Course on Fuzzy Theory and Applications, 2005
ISBN 3-540-22988-4

Barbara Dunin-Keplicz, Andrzej Jankowski, Andrzej Skowron, Marcin Szczuka
Monitoring, Security, and Rescue Techniques in Multiagent Systems, 2005
ISBN 3-540-23245-1

Bernd Reusch (Ed.)
Computational Intelligence, Theory and Applications: International Conference 8th Fuzzy Days in Dortmund, Germany, Sept. 29 – Oct. 01, 2004 Proceedings, 2005
ISBN 3-540-2280-1

Frank Hoffmann, Mario Köppen, Frank Klawonn, Rajkumar Roy (Eds.)
Soft Computing: Methodologies and Applications, 2005
ISBN 3-540-25726-8

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

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

Ashutosh Tiwari
Joshua Knowles
Erel Avineri
Keshav Dahal
Rajkumar Roy
(Eds.)

Applications of Soft Computing

Recent Trends

Springer

Dr. Ashutosh Tiwari
Dr. Rajkumar Roy
Enterprise Integration
School of Industrial and
Manufacturing Science (SIMS)
Cranfield University
Cranfield, Bedfordshire, MK43 0AL, UK
E-mail: a.tiwari@cranfield.ac.uk
E-mail: r.roy@cranfield.ac.uk

Dr. Joshua Knowles
School of Chemistry
University of Manchester
Faraday Building, Sackville Street
PO Box 88, Manchester, M60 1QD, UK
E-mail: j.knowles@manchester.ac.uk

Dr. Erel Avineri
Centre for Transport & Society
Faculty of the Built Environment
University of the West of England
Frenchay Campus
Coldharbour Lane, Bristol, BS16 1QY, UK
E-mail: erel.avineri@uwe.ac.uk

Dr. Keshav Dahal
School of Informatics
University of Bradford
Richmond Road, Bradford, West Yorkshire,
BD7 1DP, UK
E-mail: k.p.dahal@bradford.ac.uk

Library of Congress Control Number: 2006923228

ISSN print edition: 1615-3871
ISSN electronic edition: 1860-0794
ISBN-10 3-540-29123-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29123-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Printed in The Netherlands

Typesetting: by the authors and techbooks using a Springer LaTeX macro package
Cover design: *Erich Kirchner*, Heidelberg

Printed on acid-free paper SPIN: 11561101 89/techbooks 5 4 3 2 1 0

Preface

Soft computing is the study of computational methods that involve — or are tolerant of — approximation, imprecision, uncertainty and/or partial truth. Principal amongst these methods are the complementary techniques of evolutionary computing, neural networks and fuzzy systems. This volume contains a collection of papers that were presented at the 10th Online World Conference on Soft Computing in Industrial Applications (WSC10), held in September 2005 on the World Wide Web. It provides a comprehensive overview of the recent advances in the industrial uses of soft computing and covers a wide range of application areas, including optimisation, data analysis and data mining, computer graphics and vision, prediction and diagnosis, design, intelligent control, and traffic and transportation systems.

This volume is divided into nine sections, which correspond to the nine sessions in the conference program. Eight of these sections consist of refereed scientific papers organized by application type or area, while the ninth contains three invited tutorial papers.

More information about the WSC10 Conference can be found in the welcoming message from the General Chair and Technical Chair included overleaf. A further message about this series of conferences and about the World Federation on Soft Computing follows that.

We would like to express our appreciation to all the authors and members of the program and organising committee who have made this volume possible, and hope that you, the reader, find it a useful and enjoyable addition to your library.

Ashutosh Tiwari and Joshua Knowles
WSC10 – General Chair, WSC10 – Technical Chair

Cranfield University, UK
University of Manchester, UK

January 11, 2006

Message from WFSC Chair

Dear WSC10 participants,

Warm welcome to this year's World Conference on Soft Computing (WSC10) in Industrial Applications! This is the 10th year of the conference running successfully. From a simple start, the conference has achieved good recognition from other professional bodies and you researchers. Due to the 'inexpensive' nature of the conference, it has attracted research papers from around the World and has become a truly "World Conference". During this period the World Federation on Soft Computing (WFSC) has also developed a very successful Journal on Applied Soft Computing, published by Elsevier. Time is now right to go to the next stage and formalise the structure of the Virtual Organisation and expand its activities. In the coming year we shall seek volunteers to take responsibilities for different types of WFSC activities.

WSC10 has selected only 38 papers for the publication, this is the first step to improve the quality of the conference and motivate publishers to publish the papers within a well recognised publication. Thanks to Springer for their support in developing a longer term publication plan for the WSC series of conferences. Long term sustainability of WSC Series quality will depend on the quality of the publication. Finally, it is also important that we have an active discussion of papers in the conference. Due to the on-line nature of the conference, it should allow you to discuss technical papers effectively. May I take this opportunity to encourage you to take active part in the WSC10 paper discussions? Your feedback is very important for the authors.

On behalf of WFSC, I thank the authors, reviewers, Cranfield University, University of Manchester, other organisers and publisher Springer for their hard work and support in organising the conference. Special thank goes to Dr. Ashutosh Tiwari and Dr. Joshua Knowles for their leadership and good organisation. With your support and participation, all this hard work will be successful!

Professor Rajkumar Roy
Chairperson
WFSC

Cyberspace
19th September 2005

Message from WSC10 General Chair and Technical Chair

On behalf of the organising committee of the 10th *Online World Conference on Soft Computing in Industrial Applications (WSC10)*, we wish to extend a very warm welcome to you. WSC10 will be held on the World Wide Web from September 19 to October 7 2005. It is organised by the World Federation of Soft Computing (WFSC), and is co-sponsored by Springer, BT, Elsevier, European Neural Network Society (ENNS), North American Fuzzy Information Processing Society (NAFIPS), European Society for Fuzzy Logic and Technology (EUSFLAT), and International Fuzzy Systems Association (IFSA)

Soft computing is the study of computational methods that involve – or are tolerant of – approximation, imprecision, uncertainty, and/or partial truth. Principal among these methods are the complementary techniques of evolutionary computing, neural networks and fuzzy systems. WSC10 aims at bringing together outstanding research and developments in the field of soft computing and its applications, from across the world. Papers are published on-line with the dual benefits of rapid dissemination and reduced cost. WSC10 has received an excellent response this year. About 100 papers were submitted from around 30 countries, covering all the continents. This reflects the popularity and international nature of this event.

38 refereed technical papers are presented in WSC10, providing a comprehensive overview of the recent advances in the industrial applications of soft computing. The papers presented in WSC10 will appear in a Springer volume entitled, *Applications of Soft Computing: Recent Trends*, which can be ordered online. Extended versions of selected papers will also be invited for a "fast track" submission to the Elsevier Science Applied Soft Computing Journal. WSC10 covers a wide range of application areas, including optimisation, data analysis and data mining, computer graphics and vision, prediction and diagnosis, design, intelligent control, and traffic and transportation systems. Papers are separated into a number of sessions, organised by application area. The papers, together with accompanying slide presentations and multimedia demonstrations, are available to view on the Internet throughout the

conference. Authors of papers will also be available to answer questions and provide further information via conference website.

We would like to express our sincere thanks to all the authors and members of the program and organising committee who have made this event a success. We are also grateful to the sponsors of this event for their support.

We hope that you will enjoy the conference. We look forward to meet you all during the conference!

Ashutosh Tiwari and Joshua Knowles
WSC10 – General Chair, WSC10 – Technical Chair

Cranfield University, UK
University of Manchester, UK

September 19, 2005

WSC10 Organization and Program Committee

General Chair

Ashutosh Tiwari, Cranfield University, UK

Technical and Program Chair

Joshua Knowles, University of Manchester, UK

Virtual Exhibition Chair

Rajkumar Roy, Cranfield University, UK

Special Session Chair

Erel Avineri, University of the West of England, UK

Publicity Chair

Keshav Dahal, Bradford University, UK

Conference Coordinator

John Hadden, Cranfield University, UK

Advisory Board

Rajkumar Roy, Cranfield University, UK (Chair)
Ajith Abraham, Chung-Ang University, Korea
Mario Köppen, Fraunhofer IPK Berlin, Germany
Frank Hoffmann, University of Dortmund, Germany
Takeshi Furuhashi, Mie University, Japan

International Co-Chairs

Nadia Nedjah, State University of Rio de Janeiro, Brazil
Esam Shehab, Cranfield University, UK
Shengxiang Yang, University of Leicester, UK
Ruhul A. Sarker, University of New South Wales, Australia
Serge Popov, Kharkiv University of Radio Electronics, Ukraine
Ashraf Saad, Georgia Institute of Technology, USA
Muhammad Sarfraz, King Fahd University of Petroleum and Minerals, Saudi Arabia

Local Organising Committee

Esam Shehab, Cranfield University, UK (Chair)
John Hadden, Cranfield University, UK
Abiola Oduguwa, Cranfield University, UK

International Technical Program Committee

Janos Abonyi, University of Veszprem, Hungary
Ajith Abraham, Chung-Ang University, Korea
Erel Avineri, University of the West of England, UK
Bart Baesens, Catholic University of Leuven, Belgium
Soumya Banerjee, Institute of Management Studies Dehradun, India
Sigal Berman, Weizmann Institute, Israel
Christian Blum, Polytechnic of Catalunya, Spain
Ulrich Bodenhofer, University of Linz, Austria
Andrea Bonarini, Poytechnic of Milan, Italy
Brian Carse, University of the West of England, UK
Oscar Castillo, Technical Institute of Tijuana, Mexico
Carlos Coello, CINVESTAV-IPN, Mexico
Oscar Cordon, University of Granada, Spain
David Corne, University of Exeter, UK
Mauro Dell'Orco, University of Bari, Italy
Guy De Tré, Ghent University, Belgium
Giuseppe Di Fatta, University of Konstanz, Germany
Dimiter Driankov, Orebro University, Sweden
Katrin Franke, Fraunhofer IPK, Germany
Takeshi Furuhashi, Mie University, Japan
Xiao-Zhi Gao, Helsinki University of Technology, Finland
Antonio Gaspar-Cunha, University of Minho, Portugal
Ashish Ghosh, Indian Statistical Institute, India
Jens Gottlieb, SAP AG, Germany
Röderich Gross, Free University of Brussels, Belgium
Hani Hagras, University of Essex, UK
Julia Handl, University of Manchester, UK

Ioannis K. Hatzilygeroudis, University of Patras, Greece
Francisco Herrera, University of Granada, Spain
Frank Hoffmann, University of Dortmund, Germany
Hisao Ishibuchi, Osaka University, Japan
Yaochu Jin, Honda Research Institute Europe, Germany
Frank Klawonn, Wolfenbüttel University of Applied Sciences, Germany
Andreas König, University of Kaiserslautern, Germany
Mario Köppen, Fraunhofer IPK, Germany
Andras Kornai, Metacarta, USA
Natalio Krasnogor, University of Nottingham, UK
Vladik Kreinovich, University of Texas at El Paso, USA
Renato Krohling, University of York, UK
Halva Labella, Free University of Brussels, Belgium
Marie-Jeanne Lesot, Laboratoire Informatique de Paris VI, France
Luis Magdalena, Technical University of Madrid, Spain
Max Manfrin, Free University of Brussels, Belgium
Patricia Melin, Technical Institute of Tijuana, Mexico
Sanaz Mostaghim, ETH Zurich, Switzerland
Nadia Nadjah, State University of Rio de Janeiro, Brazil
Detlef Nauck, British Telecommunications plc, UK
Shervin Nouyan, Free University of Brussels, Belgium
Andreas Nürnberger, University of Magdeburg, Germany
Jae C. Oh, Syracuse University, USA
Sankar K. Pal, Indian Statistical Institute, India
Luis Paquete, Technical University of Darmstadt, Germany
Calos Andres Pena-Reyes, EPFL, Italy
Serge Popov, Kharkiv National University of Radio Electronics, Ukraine
Günther Raidl, Technical University of Vienna, Austria
Andrea Roli, Universita degli Studi G. d'Annunzio, Italy
Olivia Rossi-Doria, University of Padova, Italy
Javier Ruiz del Solar, University of Chile, Chile
Ashraf Saad, Georgia Institute of Technology, USA
Michael Sampels, Free University of Brussels, Belgium
Muhammad Sarfraz, King Fahd University of Petroleum and Minerals, Saudi Arabia
Jim Smith, University of the West of England, UK
Thomas A. Sudkamp, Wright State University Ohio, USA
Yukinori Suzuki, Muroran Institute of Technology, Japan
Vicenc Torra, University of Barcelona, Spain
Vito Trianni, Free University of Brussels, Belgium
Edward Tunstel, NASA Jet Propulsion Lab, USA
Berend Jan van der Zwaag, University of Twente, The Netherlands
Marley Vellasco, University College London, UK
Christian Wöhler, Daimler Chrysler AG, Germany

Contents

Part I Computer Graphics, Imaging and Vision

Part II Control and Robotics

Part III Design

Part IV Pattern Recognition

Part V Classification

Part VI Identification and Forecasting

Part VII Optimization Problems: Assignment, Partitioning and Ordering

Part VIII Optimization Methods: Development and Analysis

Part IX Tutorials

Part I

Computer Graphics, Imaging and Vision

Efficient Genetic Algorithms for Arabic Handwritten Characters Recognition

Dr. Laheeb M. Al-zoubaidy
Associative Proff.[a]
Department of Computer Science, College of Computer Sciences & Math., Mosul University Iraq

Abstract. The main challenge in handwritten character recognition involves the development of a method that can generate descriptions of the handwritten objects in a short period of time. Genetic algorithm is probably the most efficient method available for character recognition. In this paper a methodology for feature selection in unsupervised learning is proposed. It makes use of a multiobjective genetic algorithm where the minimization of the number of features and a validity index that measures the quality of clusters have been used to guide the search towards the more discriminate features and the best number of clusters.

The proposed strategy is evaluated synthetic data sets and then it is applied to Arabic handwritten characters recognition. Comprehensive experiments demonstrate the feasibility and efficiency of the proposed methodology, and show that Genetic Algorithm (GA) are applied here to improve the recognition speed as well as the recognition accuracy.

Keywords. Arabic handwritten characters recognition, Genetic Algorithm (GA), Feature Extracted, Feature Selection

1 Introduction

In areas of pattern recognition, features can be characterized as a way to distinguish one class of objects from another in a more concise and meaningful manner than is offered by the raw representations. Therefore, it is of crucial importance to define meaningful features when we plan to develop a good recognizer, although it has been known that a general solution has not been found. In many cases, features are generally defined by hand based on the experience and intuition of the designer.

Depending on problems given, there are a number and variety of features can be defined in terms of extracting methods and ways of representation. In many practical applications, it is not unusual to encounter

[a] Corresponding Author.
E-mail addresses: lahmzub@yahoo.com

problems involving hundreds of features. The designer usually believes that every feature is meaningful for at least some of the discriminations. However, it has been observed in practice that, beyond certain point, the inclusion of additional features leads to worse rather than better performance. Furthermore, including more features means simply increasing processing time. This apparent paradox presents a genuine and serious problem for classifier design [5].

Arabic is a major world language spoken by 186 million people [6]. Very little research has gone into character recognition in Arabic due to the difficulty of the task and lack of researchers interested in this field. As the Arab world becomes increasingly computerized and mobile, and technology becomes increasingly ubiquitous, the need for a natural interface becomes apparent.

Classification of Arabic handwritten characters, which has been a typical example of pattern recognition, contains the same problem. Due to diversity in ones written by a single person. In order to deal with such a wide range of diversity existing in handwritings, recognizers often employ several hundreds of features.

Approaches to circumvent the feature selection problem found in the literature are: linear combination of feature vectors, principal component analysis, simple selection based on the discrimination power of features and sequential forward/backward selection. Because feature dimensions are large enough, and the solution space has the characteristics of the multi-modal function, the feature selection process takes a lot of time when most of the above mentioned approaches are adopted, and a local optimum can be chosen as the solution, instead of the global one [10,12]

In this paper, we introduce a feature selection method, which can minimize most of the problems can be found in the conventional approaches, by applying genetic algorithms(GA) which recently received considerable attention regarding their potential as an optimization technique for complex problems. Genetic algorithms are stochastic search technique based on the mechanism of natural selection and natural genetics.

In this paper we propose a methodology for feature selection in unsupervised learning for handwritten Arabic characters recognition. It makes use of the Genetic Algorithm (GA) proposed by Kim [4] which deals with multi-objective optimization. The objective is to find a set of non dominant solutions which contain the more discriminate features and the more pertinent number of clusters. We have used two criteria to guide the search: minimization of the number of features and minimization of a validity index that measures the quality of clusters. A standard K-Means algorithm is applied to form the given number of clusters based on the selected features.

Afterwards, it is applied to handwritten Arabic characters recognition in order to optimize the character classifiers. Experimental results show the efficiency of the proposed methodology

2 Genetic Algorithms

In this section we present a brief introduction about genetic algorithms. [8].

The genetic algorithm is a model of machine learning which derives its behavior from a metaphor of some of the mechanisms of evolution in nature. This is done by the creation within a machine of a population of individuals represented by chromosomes, in essence a set of character strings that are analogous to the base-4 chromosomes that we see in our own DNA.

The individuals represent candidate solutions to the optimization problem being solved. In genetic algorithms, the individuals are typically represented by *n*-bit binary vectors. The resulting search space corresponds to an *n*–dimensional boolean space. It is assumed that the quality of each candidate solution can be evaluated using a fitness function.

Genetic algorithms use some form of fitness-dependent probabilistic selection of individuals from the current population to produce individuals for the next generation. The selected individuals are submitted to the action of genetic operators to obtain new individuals that constitute the next generation. Mutation and crossover are two of the most commonly used operators that are used with genetic algorithms that represent individuals as binary strings. Mutation operates on a single string and generally changes a bit at random while crossover operates on two parent strings to produce two offsprings. Other genetic representations require the use of appropriate genetic operators.

The process of fitness-dependent selection and application of genetic operators to generate successive generations of individuals is repeated many times until a satisfactory solution is found. In practice, the performance of genetic algorithm depends on a number of factors including: the choice of genetic representation and operators, the fitness function, the details of the fitness-dependent selection procedure, and the various user-determined parameters such as population size, probability of application of different genetic operators, etc. The specific choices made in the experiments reported in this paper are summarized in Table 2, (see Figure 1) depicts a GA cycle.

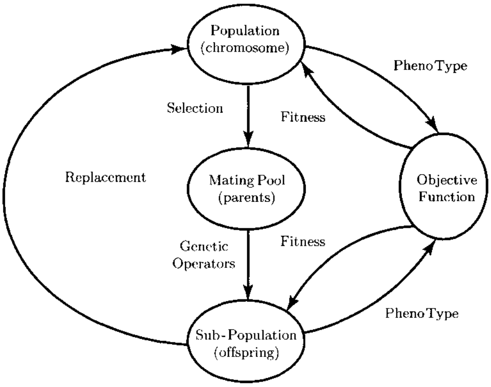


Fig. 1. Genetic algorithm (GA) Cycle

The basic operation of the genetic algorithm is outlined as follows [1]:

```
begin
 t <- 0
 initialize P(t)
 while (not termination condition)
   t <- t + 1
   select P(t) from p(t - 1)
   crossover P(t)
   mutate P(t)
   evaluate P(t)
 end
end
```

Since genetic algorithms were designed to efficiently search large spaces, they have been used for a number of different application areas, and genetic algorithms mainly deal with optimization problems. The applications include job scheduling, TSP (Traveling Salesman Problem), communication network design image restoration, data clustering, and feature selection for speaker identification, camera calibration [9], signature verification [9], medical diagnosis [11], facial modeling [2] and handwritten recognition [4].

3 Recognition System

Figure 2 briefly shows the system flow of a general handwriting recognizer. An input image is encoded for saving space and easier manipulation in the subsequent steps. Several steps of preprocessing, such as noise removal, slant correction, and smoothing are applied to the encoded image. Defined features are extracted from the processed image and the recognition process is performed using the features [7].

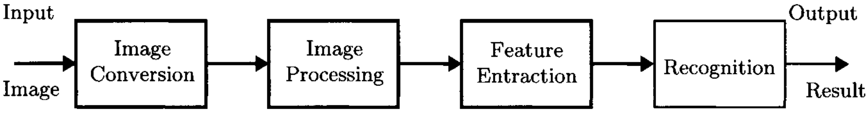


Fig. 2. Flow diagram of recognition system

Arabic handwritten character recognizer as shown in Figure 3 is used to evaluate the effect of the proposed feature selection algorithm. The recognizer has two phases (see Figure 3) training and testing.

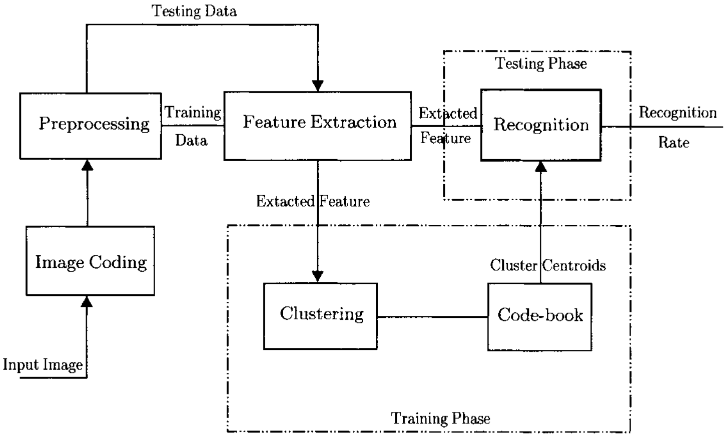


Fig. 3. Arabic handwritten character recognition system

3.1. Feature Extracted Using Genetic Algorithm (GA)

In this subsection we present the choice of a representation for encoding candidate solutions to be manipulated by the genetic algorithm. Each individual in the population represents a candidate solution to the feature subset selection problem. Let *m* be the total number of features available to choose from to represent the patterns to be classifier ($m = 74$ in our case). The individual (chromosome) is represented by a binary vector of dimension *m*. If a bit is a 1, it means that the corresponding feature is selected; otherwise the feature is not selected. This is the simplest and most straightforward representation scheme [4].

Feature, dimensions of 74, are defined and extracted so to see how the proposed feature extracted algorithm deals with features. The smaller set of 74 consists of 2 global features – aspect ratio and stroke ratio of the entire template, and 72 local features – distribution of the eight directional slopes for each of 9 (3*3) sub images [3]. (See Figure 4).

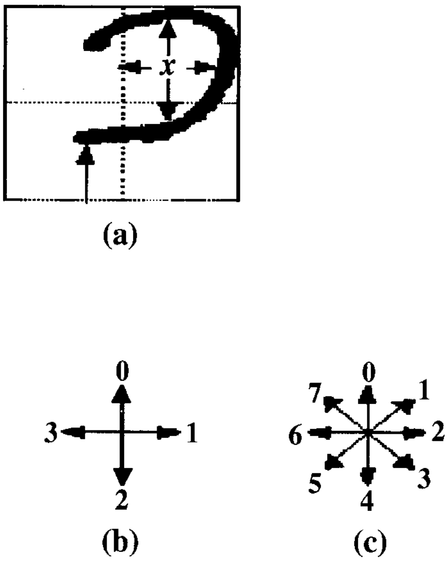


Fig. 4. Feature set for Arabic character Dal (د): (a) Concavities, (b) 4-Freeman directions (c) 8-Freeman directions

The feature extracted procedure based on Genetic Algorithm (GA). Since we are representing a chromosome through a binary string, the operator's mutation and crossover operates in the following way: Mutation operates on a single string and generally changes a bit at random. Thus, a string 11010 may, as a consequence of random mutation gets changed to 11110. Crossover on two parent strings to produce two offsprings. With a randomly chosen crossover position 4, the two strings 01101 and 11000 yield the offspring 01100 and 11001 as a result of crossover.

3.2. Selection Mechanism

The selection mechanism is responsible for selecting the parent chromosome from the population and forming the mating pool. The selection mechanism emulates the survival of the fittest mechanism in nature. It is expected that a fitter chromosome receives a higher number of offsprings and thus has a higher chance of surviving on the subsequent evolution while the weaker chromosomes will eventually die.

In this work we are using the roulette wheel selection [4] which is one of the most common and easy-to implement selection mechanism. Basically it works as follows: each chromosome in the population is associated with a sector in a virtual wheel. According to the fitness value of the chromosome, the sector will have a larger area when the corresponding chromosome has a better fitness value while a lower fitness value will lead to a smaller sector.

3.3. Feature Selection Using Genetic Algorithm (GA)

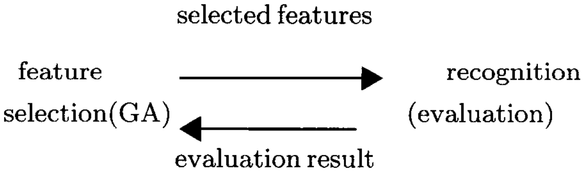
The recognizer evaluates the selected features and returns the results to the feature selection algorithm. The genetic algorithms evaluate the solution based on the recognition result. The process is repeated until the termination condition is satisfied.

The feature selection procedure, based on the simple genetic algorithm (SGA), is presented in this section. A chromosome is represented by a bit string, (see Figure 5(a)), in the Figure 5(a), '1's in the string represent the corresponding features are selected and '0's represent the corresponding ones are not selected. Figure 5(b) shows the feature selection process. A new generation is formed by selection some of the parents and offspring according to the fitness value, and rejecting others so as to keep the population size constant. Offspring are formed by either merging two chromosomes using a crossover process and modifying a chromosome using a mutation process. Fitter chromosome have higher probability of being selected. The recognizer evaluate the selected features and returns the results to the feature selection algorithm. The genetic algorithms evaluate the solution based on the recognition result. The process is repeated until the termination condition is satisfied.

1010101011110........10101101011

length of string = feature dimension

(a)



(b)

Fig. 5. (a) A chromosome (b) feature selection process

Distance between the centroid of a cluster and the feature is computed using Eqn. (1), and it is modified to Eqn. (2) in order to consider the features selected only:

$$Dist = \sum_{i=0}^{p} (x_i - y_i)^2 \qquad (1)$$

$$Dist = \sum_{i=0}^{p} (x_i - y_i)^2 \text{ when } s_i = 1 \qquad (2)$$

where, x_i is the i-th feature extracted from testing data, y_i is the i-th feature of any cluster in the codebook, p is the number of features before the selection is performed, and s_i indicates whether the i-th feature is selected (1) or not (0).

The computation complexity of the matching process in the recognition module, is reduced as much as the number of features reduced, (p - q), when q is the number of features selected. Reducing the feature dimension is important in terms of improving recognition speed because it has been known that the matching is the most time consuming process among the modules in a recognizer and the number of matching is proportional to the number of features being compared [11]. In addition, storage space required for the codebook is reduced (see Figure 6), where k is the number of clusters.

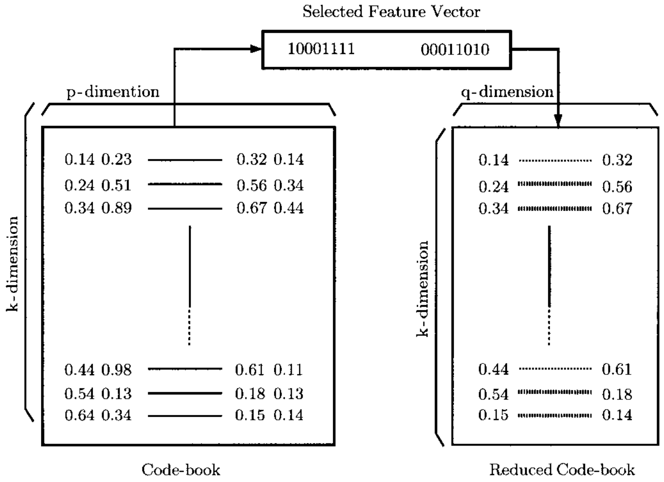


Fig. 6. Reducing Coodebook space

3.3.1 Introducing Variable Weights

As a result of the feature selection process, the number of features is reduced and slight performance degradation could be expected. Therefore, the feature selection method can be used for applications in which efficiency in terms of both speed and space is important in spite of some degree of degradation in recognition accuracy.

In this section, we use an approach to improve the recognition accuracy using the information collected while the feature selection is being performed. In the approach we do not aim for reducing the feature dimension, but we build a weight matrix by observing bits in the chromosomes. (see Figure 7) to show how to build the matrix using this approach. If, while the genetic algorithms are being performed, a bit in a chromosome is '1' (selected), the corresponding element in the matrix is increased by 1, otherwise no change occurs. After the feature selection process is completed, the matrix is normalized and used in recognition module. The equation used for computing distance between features of an input image and the centroid of a cluster is changed from Eqn. (1) to Eqn. (3),

$$Dist = \sum_{i=0}^{p}(x_i - y_i)^2 \times \alpha_i \tag{3}$$

Where α_i is i-th normalized weight. Even though there is no reduction in the feature dimension in the approach, the weights reflect how often the corresponding features are selected during the feature selection process, and the weight is considered during the matching because more frequently selected features can be regarded as more important ones than others.

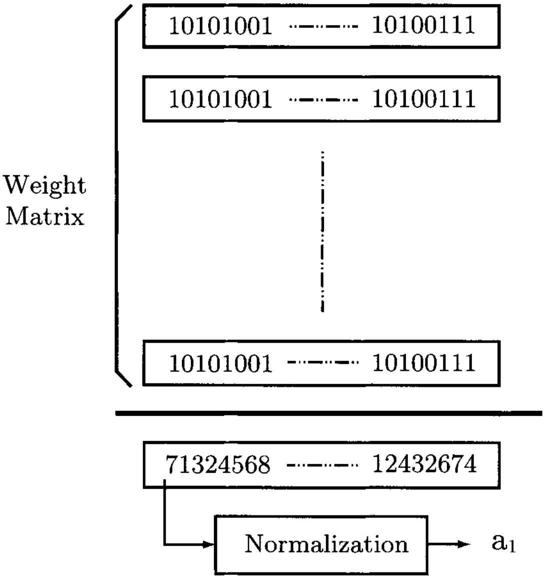


Fig. 7. Building the weight matrix

4 Experimental Results

Arabic Character handwritten Recognizer working with features extracted from image representation is used in the experiment. Handwritten Arabic characters recognizer conducted on the Arabic handwriting of 5 independent writers images, 28 images (represent 28 Arabic characters) are selected for codebook generation, evaluation of the feature selection algorithm, and performance measure of the recognizer with the selected features in the testing phase, respectively.

4.1 Feature Selection Using GA

Table 1 shows recognition accuracy of the Arabic Character handwritten recognizer with selected features using the GA. In the table, feature dimension of 74 represents the case that whole features are included without reduction.

According to the results in the table, significant reduction in the feature dimension, with trivial drop in the recognition accuracy, is observed. Parameters chosen for the GA are summarized in Table 2.

Table 1. Recognition performance with various number of selected features

Feature Dim.	74	74(Var. weight)
Recognition Rate	95%	95.3%

Tab. 2. Parameters used in the GA

Parameters	**Value**
Population Size	30
String length	74
Probability of Crossover	0.8
Probability of mutation	0.007
Selection method	Roulette wheel
Number of generations	1000

Also The last column in Table 1 represents the recognition rate with the variable weights obtained during the feature selection process. Variable weights are assigned to all features when the distance is computed using Eqn. (3). Comparing the result with the original, assigning different weight to each feature, depending on how often the feature is selected during the feature selection process, improves the recognition rate.

5. Compare Genetic Algorithm Approach with Previous Works

Recognition approach (genetic approach) using in this work compared with previous works according to the number of writers and the nature of training samples used by the system (see Table 3). The system is required to deal with as much writers as possible, training samples of each handwriting. The work presented by klassen [6] reports correct characters recognition using genetic Algorithm is an evolutionary machine learning strategy that uses cross-overs and mutations to create a program of mathematical operations on a data population to produce the "fittest" population. Genetic Programming had a positive example average of 92% for the training set, 77% for the validation set, and 72% for the test set.

Table 3. Comparison of performance and test conditions for recognition system in recent studies.

	Klassen,2001	Genetic algorithm approach
Training Rate	92%	95%
Validation rate	77%	89%
Test rate	72%	85%
Writers	25	25
Classes	15	28

The results obtained in this research were very promising and identification accuracy as high as 95% for the training set, 89% for the validation set, and 85.3% for the test set. classifier has shown a good performance.

6 Conclusion

In this paper, a feature selection method based on genetic algorithms is presented. The experimental results prove that feature selection using the GA reduces 30-50% of features with trivial drop in recognition accuracy, applying the variable weight obtained during the feature selection process improves the recognition accuracy, and the feature selection is working more effectively for features with larger dimension. The results obtained in this research were very promising and identification accuracy as high as 95% for the training set, 89% for the validation set, and 85.3% for the test set.

References

[1] Davis L. (1991), Handbook on Genetic Algorithms, Van Nostrand, Reinhold.

[2] Ho S.Y. & Huang H.L. (2001), Facial Modeling From an Uncalibrated Face Image Using a Coarse-to-Fine Genetic Algorithm, Pattern Recognition, 34(5).

[3] Kim G. & Govindaraju V. (1997), A Lexicon Driven Approach to Handwritten Word Recognition for Real-Time Applications, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4).

[4] Kim G. &, Kim S. (2000), Feature Selection Using Genetic Algorithms for Handwritten Character Recognition. In 7th IWFHR, Amsterdam-Netherlands.

[5] Kim B.S. & Song H.-J. (1998), An Efficient Preprocessing and Feature Extraction Method Based on Chain Code for Recognizing Handwritten Characters (in Korean), Journal of KISS(B): Software and Applications, 25(12).

[6] Klassen T. (2001), Towards Neural Network Recognition Of Handwritten Arabic Letters. Masters Thesis, Dalhousie University, Halifax, U.S.A.

[7] Madhvanath S., Kim G., & Govindaraju V. (1999), Chaincode Contour Processing for Handwritten Word Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(9).

[8] Mitchell M. (1996), An Introduction to Genetic Algorithms. MIT Press, Cambridge – MA.

[9] Ramesh V.E. & Murty N.(1999), Off-line Signature Verification Using Genetically Optimized Weighted Features, Pattern Recognition, 32(2).

[10] Shi D. (1998), Feature Selection for Handwritten Chinese Character Recognition Based on Genetic Algorithms, International Conference of Systems, Man and Cybernetics, 5.

[11] Yang J. & Honavar V. (1998), Feature Subset Selection Using a Genetic Algorithm, IEEE Intelligent Systems, 13(1).

[12] Wang Y. K. & Fan K. C. (1996), Applying Genetic Algorithms on Pattern Recognition: An Analysis and Survey. In Proceedings of ICPR'96.

Generic Black Box Optimisation Algorithms for Colour Quantisation

Gerald Schaefer and Lars Nolle

School of Computing and Informatics, Nottingham Trent University, U.K.
{Gerald.Schaefer,Lars.Nolle}@ntu.ac.uk

Abstract. We apply a variant of Simulated Annealing (SA) as a standard black-box optimisation algorithm to the colour quantisation problem. The main advantage of black-box optimisation algorithms is that they do not require any domain specific knowledge yet are able to provide a near optimal solution. We evaluate the effectiveness of our approach by comparing its performance with several specialised colour quantisation algorithms. The results obtained show that even without any domain specific knowledge our SA based algorithm is able to outperform standard quantisation algorithms and hence to provide images with superior image quality.

Keywords: Black box optimisation, simulated annealing, colour quantisation

1 Introduction

Colour quantisation is a common image processing technique that allows the representation of true colour images using only a small number of colours and is useful for displaying images on limited hardware such as mobile devices, for image compression, and for other applications such as image retrieval [11]. True colour images typically use 24 bits per pixel which results in an overall gamut of 2^{24} i.e. more than 16.8 million different colours. Colour quantisation uses a colour palette that contains only a small number of colours (usually between 8 and 256) and pixel data are then stored as indices to this palette. Clearly the choice of the colours that make up the palette has a crucial influence on the image quality of the quantised image However, the selection of the optimal colour palette is known to be an np-hard problem [4]. In the image processing literature many different algorithms have been introduced that aim to find a palette that allows for good image quality of the quantised image [4, 3, 2].

In this paper we apply a variant of Simulated Annealing (SA) as a standard black-box optimisation algorithm to the colour quantisation problem. The main advantage of black-box optimisation algorithms is that they do not require any domain specific knowledge yet are able to provide a near optimal solution. We evaluate the effectiveness of our approach by comparing its performance to the results obtained by several purpose built colour quantisation algorithms [4, 3, 2]. The results obtained show that even without any domain specific knowledge our SA based algorithm is able to outperform standard quantisation algorithms and hence to provide palettised images with superior image quality.

The rest of the paper is organised as follows: Section 2 provides the background for optimisation based on Simulated Annealing. Section 3 explains our application of SWASA, a modified SA algorithm, to the colour quantisation problem. Section 4 provides experimental results based on a set of standard test images while 5 concludes the paper.

2 Simulated Annealing

Simulated annealing (SA) was first introduced as a general optimisation method by Kirkpatrick *et al.* [6], based on the work of Metropolis *et al.* [7]. It simulates the annealing of metal, in which the metal is heated-up to a temperature near its melting point and then slowly cooled down. This allows the particles to move towards a minimum energy state, with a more uniform crystalline structure. The process therefore permits some control over the microstructure.

Simulated annealing is a variation of the hill-climbing algorithm. Both start from a randomly selected point within the search space of all the possible solutions. Each point in the search space has a measurable error value, E, associated with it, which indicates the quality of the solution. From the current point in search space, new trial solutions are selected for testing from the neighborhood of the current solution. This is usually done by moving a small step in a random direction. In this application, small and equally distributed random numbers from the interval $[-s_{max}, s_{max}]$ are added to each component of the current solution vector, where s_{max} is called the 'maximum step width'. The values for s_{max} need to be chosen from the interval between 0 and the upper limit of the search space dimension. The decrease in error values is ΔE. If ΔE is negative, i.e. the error of a trial solution is less than the error of the current one, the trial solution is accepted as the current solution.

Unlike hill-climbing SA does not automatically reject a new candidate solution if ΔE is positive. Instead it becomes the current solution with probability $p(T)$ which is usually determined using

$$p(T) = e^{-\Delta E/T} \tag{1}$$

where T is referred to as 'temperature', an abstract control parameter for the cooling schedule. For a given temperature and positive values of ΔE the probability function shown in Equation 1 has a defined upper limit of one, and tends towards zero for large positive values of ΔE. That means, in a practical computer application, the probability $p(T)$ has to be calculated for each candidate solution and to be compared with an equally distributed random number from the interval $[0,1]$. If the probability $p(T)$ is greater than the random number the candidate solution is accepted as the current solution, otherwise it is rejected.

The algorithm starts with a high temperature i.e. with a high transition probability. The temperature is then reduced towards zero, usually in steps, according to a cooling schedule such as

$$T_{n+1} = \alpha T_n \tag{2}$$

where T_n is the temperature at step n and α is the cooling coefficient (usually between 0.8 and 0.99).

During each step the temperature must be held constant for an appropriate number of iterations in order to allow the algorithm to settle into a 'thermal equilibrium' i.e. a balanced state. If the number of iterations is too small the algorithm is likely to converge to a local minimum.

Step width adapting simulated annealing (SWASA) [8] overcomes the problems associated with constant values for s_{max} by using a scaling function [10] to adapt the maximum step width to the current iteration by

$$s_{max}(n) = \frac{2s_0}{1 + e^{\beta n / n_{max}}} \tag{3}$$

where $s_{max}(n)$ is the maximum step width at iteration n, s_0 is the initial maximum step width, n_{max} the maximum number of iterations and β is an adaptation constant.

3 Simulated Annealing for Colour Quantisation

In this paper we apply the SWASA algorithm described in Section 2 as a black box optimisation algorithm to the colour quantisation problem. For colour quantisation the objective is to minimise the total error introduced through the application of a colour palette. The colour palette C for an image I, a codebook of k colour vectors, should then be chosen so as to minimise the error function

$$\text{error}(C, I) = \frac{1}{\sum_{j=1}^{k} l_j} \sum_{i=1}^{k} \sum_{j=1}^{l_i} ||C_i - I_j|| + p(C, I) \tag{4}$$

with

$$p(C, I) = \sum_{i=1}^{k} \delta a_i, \quad a_i = \begin{cases} 1 & \text{if } l_i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where l_i is the number of pixels I_j represented by colour C_i of the palette, $||.||$ is the Euclidean distance in RGB space, and δ is a constant ($\delta = 10$ in our experiments). The objective function error(C, I) used is hence a combination of the mean Euclidean distance and a penalty function. The penalty function $p(C, I)$ was integrated in order to avoid unused palette colours by adding a constant penalty value to the error for each entry in the codebook that is not used in the resulting picture.

As can be seen from Equation 4 the objective function is highly non-linear, i.e. it has a high degree of epistasis [1]. Past experience [9] has shown that for this kind of optimisation problems simulated annealing outperforms other generic optimisation algorithms like genetic algorithms [5].

For our colour quantisation algorithm we employ a population based version of the SWASA algorithm with a population size of 10. The start temperature was chosen to be 100 and the cooling coefficient was set to 0.9. The temperature was kept constant over 20 iterations and the maximum number of iterations was set to 10000.

4 Experimental Results

In order to evaluate our new method for colour quantisation we have taken a set of six standard images commonly used in the colour quantisation literature (*Lenna, Peppers, Mandrill, Sailboat, Airplane, and Pool* - see Figure 1) and applied our optimisation scheme to generate quantised images with a palette of 16 colours.

To put the results we obtain into context we have also implemented four popular colour quantisation algorithms to generate corresponding quantised images with palette size 16. The algorithms we have tested were:

- Popularity algorithm [4]: Following a uniform quantisation to 5 bits per channel the n colours that are represented most often form the colour palette.
- Median cut quantisation [4]: An iterative algorithm that repeatedly splits (by a plane through the median point) colour cells into sub-cells..
- Octree quantisation [3]: The colour space is represented as an octree where sub-branches are successively merged to form the palette.
- Neuquant [2]: A one-dimensional self-organising Kohonen neural network is applied to generate the colour map.

For all algorithms, pixels in the quantised images were assigned to their nearest neighbours in the colour palette to provide the best possible image quality.

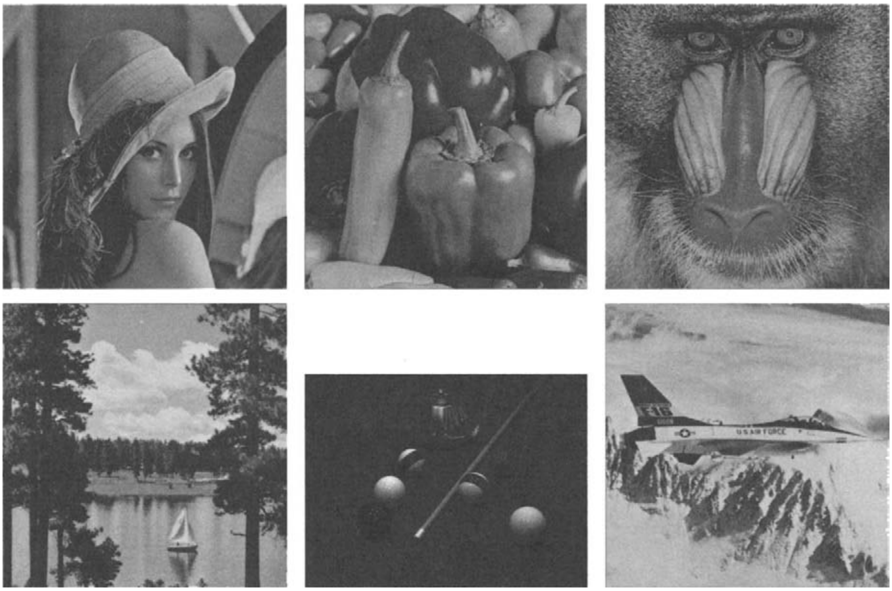


Fig. 1. The six test images used in the experiments: (*Lenna, Peppers, Mandrill, Sailboat, Pool, and Airplane.* (from left to right, top to bottom).

The results are listed in Table 1, expressed in terms of peak-signal-to-noise-ratio (PSNR) defined as

$$\mathrm{PSNR}(I_1, I_2) = 10 \log_{10} \frac{255^2}{MSE(I_1, I_2)} \tag{6}$$

with MSE (the mean-squared error) given as

$$\mathrm{MSE}(I_1, I_2) = \frac{1}{3nm} \sum_{i=1}^{n} \sum_{j=1}^{m} [(R_1(i,j) - R_2(i,j))^2 + (G_1(i,j) - G_2(i,j))^2 + (B_1(i,j) - B_2(i,j))^2] \tag{7}$$

Table 1. Quantisation results, given in terms of PSNR [dB].

Image	Popularity alg.	Median cut	Octree	Neuquant	Sim. Annealing
Lenna	22.24	23.79	27.45	**27.82**	27.79
Peppers	18.56	24.10	25.80	26.04	**26.16**
Mandrill	18.00	21.52	24.21	**24.59**	24.46
Sailboat	8.73	22.01	26.04	**26.81**	26.69
Airplane	15.91	24.32	28.77	28.24	**29.43**
Pool	19.87	24.57	29.39	27.08	**29.84**
mean	17.22	23.39	26.94	26.73	**27.40**

where $R(i,j)$, $G(i,j)$, and $B(i,j)$ are the red, green, and blue pixel values at location (i,j) and n and m are the dimensions of the images.

From Table 1 we can see our Simulated Annealing approach to colour quantisation obtains the best results for three of the six images and comes close 2-nd for the other three. Overall a mean PSNR of 27.40 dB is achieved which is significantly better than the 26.94 and 26.73 dB obtained by Octree and Neuquant, the two next best algorithms.

An example of the performance of the different algorithms is provided in Figure 2 which shows part of the *Pool* image together with the same part extracted from the images colour quantised by all algorithms. It is clear that the popularity algorithm performs poorly on this image and assigns virtually all of the colours in the palette to green and achromatic colours. Median cut is better but still provides fairly poor colour reproduction; most of the colours in the quantised image are fairly different from the original. The same holds true for the images produced by Neuquant. Here the most obvious artefact is the absence of an appropriate red colour in the colour palette. A far better result is achieved by the Octree algorithm, although here also the red is not very accurate and the colour of the cue is greenish instead of brown. Clearly the best image quality is maintained by applying our Simulated annealing technique. Although the colour palette has only 16 entries all colours of the original image are accurately presented including the red ball and the colour of the billiard cue.

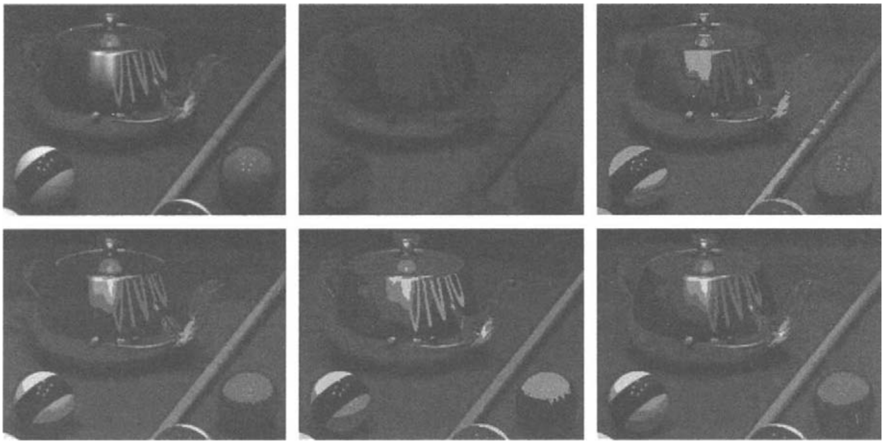


Fig. 2. Part of original *Pool* image (top-left) and corresponding images quantised with (from left to right, top to bottom): Popularity algorithm, Median cut, Octree quantisation, Neuquant, and our Simulated annealing approach.

5 Conclusions

In this work we have applied a variant of Simulated Annealing as a standard black-box optimisation algorithm to the colour quantisation problem. Experimental results obtained on a set of standard test images have demonstrated that this type of optimisation techniques cannot only be effectively employed but is even able to outperform standard purpose built colour quantisation algorithms.

References

1. Y. Davidor. Epistasis variance: Suitability of a representation to genetic algorithms. *Complex Systems*, 4:369–383, 1990.
2. A.H. Dekker. Kohonen neural networks for optimal colour quantization. *Network: Computation in Neural Systems*, 5:351–367, 1994.
3. M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. In A.S. Glassner, editor, *Graphics Gems*, pages 287–293. 1990.
4. P. S. Heckbert. Color image quantization for frame buffer display. *ACM Computer Graphics (ACM SIGGRAPH '82 Proceedings)*, 16(3):297–307, 1982.
5. J.H. Holland. *Adaptation in Natural and Artificial Systems.* University of Mitchigan Press, 1975.
6. S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
7. A. Metropolis, W. Rosenbluth, M.N. Rosenbluth, H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
8. L. Nolle. On the effect of step width selection schemes on the performance of stochastic local search strategies. In *18th European Simulation Multi-Conference*, pages 149–153, 2004.
9. L. Nolle, D.A. Armstrong, A.A. Hopgood, and J.A. Ware. Simulated annealing and genetic algorithms applied to finishing mill optimisation for hot rolling of wide steel strip. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 6(2):104–111, 2002.
10. L. Nolle, A. Goodyear, A.A. Hopgood, P.D. Picton, and N. Braithwaite. On step width adaptation in simulated annealing for continuous parameter optimisation. In *Computational Intelligence - Theory and Applications*, volume 2206 of *Lecture Notes in Computer Science*, pages 589–598. Springer, 2001.
11. G. Schaefer, G. Qiu, and G. Finlayson. Retrieval of palettised colour images. In *Storage and Retrieval for Image and Video Databases VIII*, volume 3972 of *Proceedings of SPIE*, pages 483–493, 2000.

Neural Network Combined with Fuzzy Logic to Remove Salt and Pepper Noise in Digital Images

A. Faro, D. Giordano, C. Spampinato
Dipartimento di Ingegneria e Informatica e delle telecomunicazioni
Università degli studi di Catania, viale A. Doria 6, Catania, 95125 Italy

Abstract. Image denoising is an important step in the pre-processing of images. Aim of the paper is to remove the salt and pepper noise on images by using a novel filter based on neural network and fuzzy logic. By this filter it is possible to remove only the pixels that are really affected by noise, thus avoiding image distortion due to the removal of good pixels. A comparison between the proposed filter and the classical median filter shows an increase of about 20% of the peak signal to noise ratio and a better capacity of preserving the details of the images. The proposed approach outperforms the existing algorithms and does not depend on the noise level.

1 Introduction

Digital images are affected by different kinds of noise such as Gaussian noise, random noise with normal distribution, salt and pepper noise, and black and white impulses on the digital image. Gaussian noise is usually the result of a camera electronic noise, whereas salt and pepper noise is the result of inoperative or 'dead' pixel within the camera sensor. Linear filters are most frequently used to suppress the noise because they may be easily designed and implemented. However, linear filters are suitable to eliminate linear noises. For other kinds of noise, the performance of such filters highly decreases and specific denoising methods have to be adopted.

This paper deals with how to manage the salt and pepper noise, that is an impulsive noise with uniform spatial distribution in the image where each noisy dot can be either an isolated pixel or composed of more than one pixel. Median filters [1] have been widely used for removing such impulsive noise, since they are quite effective for the noise removal and easy to implement. However, a median filter assigns to each pixel P the median value of the pixels belonging to a small window around P, and therefore the filter removes image details, being little sensible to the extreme values of the scale (e.g., the black and white of the grey tone scale), and also causes image distortion because it is applied to all the pixels of the image. The former problem has been widely investigated and may be solved by assigning to any pixel the value computed by an algorithm, let us say algorithm I, that aims at minimizing a suitable objective function [2]. During the minimization procedure the luminosity of most of the uncorrupted pixels will converge to the uncorrupted values whereas the other pixels will be restored preserving the edges and the local features. Since not all the uncorrupted pixels remain unchanged, algorithm I may cause out of focus and distorted images too. For this reason other algorithms, let us say algorithms II, have been proposed (e.g., [3]) to identify the noisy pixels so that the previous algorithm I is applied only to restore the corrupted pixels. In [4] another algorithm, let us say algorithm III, is proposed that shows better performances. It consists in the application of the algorithm II followed by the algorithm I in which the objective function takes into account only the uncorrupted pixels around the corrupted pixel to be restored. However, some performance limitations of the proposed algorithms I, II and III have been pointed out too. These limitations deal with the high processing time for image denoising and the rapid decrease of performance at increasing levels of the salt and pepper noise.

Aim of the paper is to show that by a soft computing approach it is possible to restore the images corrupted by salt and pepper noise with performances better than the ones obtainable by the available algorithms. The proposed approach makes use of neural networks to identify the set S of the pixels that are potentially affected by the salt and pepper noise and of fuzzy logic to avoid considering affected by noise pixels that belong to S but that have not been corrupted. The restoration of the noisy pixels is carried out by the median filter that takes into account only the uncorrupted pixels around the corrupted pixel to be restored. The use of objective functions, such as the one proposed in [2], instead of the median filter for better preserving the edges and the local features of the images is for further study. The paper is organized as follows: Sect. 2 addresses the use of neural network to understand what are the potentially noisy pixels. Section. 3 focuses on the use of fuzzy logic to choose the threshold for denoising the image. Section. 4 describes the results of processing real images and finally Sect. 5 reports the conclusions.

2 Neural Network Approach for Noisy Pixel Identification

Multi-layer perceptron networks trained by back propagation are among the most popular and versatile form of neural networks. The input vector representing the pattern to recognize is passed to the input layer and distributed to the subsequent hidden layer and to the output layer via weighted connections.

Each neuron in the network operates by taking the sum of its weighted inputs and passing the result through a nonlinear activation function.

Multi-layer neural networks are well known for their capability of approximating non linear bi-dimensional functions such as the one that represents the luminosity $L(x,y)$ of an image whose pixels (x,y) are possibly corrupted by a salt and pepper noise. In our approach the original image is approximated by an image extracted from the original one by using a neural network having two input neurons related to the coordinates (x,y), some intermediate neurons and an output neuron related to the luminosity L_a of the pixel of coordinates (x,y), *i.e.*, $L_a(x,y)$. The network tries during the training phase of reproducing $L(x,y)$. However, due to the non-linearity of the function $L(x,y)$ it is not possible for the neural network to reproduce exactly this function. The best that can done is to modify its synaptic weights by a generalized delta rule that aims at minimizing the distance between $L_a(x,y)$ and $L(x,y)$.

Interestingly, due to the high variations of the luminosity of the pixels affected by the salt and pepper noise with respect to the neighboring pixels, the distance between the luminosity of the pixels corrupted by the noise and the one learned during the testing phase remains very high since the neural network is not able to follow the high luminosity variations produced by the noise. This feature allows us to use the image learnt by the neural network as a filter, being possible to isolate the potentially noisy pixels as the ones of the learnt image that are characterized by a high distance from the corresponding pixels belonging to the original image.

To illustrate how the method works, we show in Fig. 1a the original image I whose pixel luminosity is given by the luminosity $L(x,y)$ affected by the salt and pepper noise, whereas Fig.1c points out the approximated image I_A whose pixel luminosity is given by the function $L_a(x,y)$ extracted from image I by using the neural network drawn in Fig.1.b. The hidden layer consists of ten neurons which have proved adequate for our scope. By subtracting the approximated image from the original one we obtain the error image I_E, drawn in Fig. 2a, whose pixel luminosity is given by $\Delta L(x,y) = L(x,y) - L_a(x,y)$. Let $P_N(x,y)$ be the pixels that are potentially covered by noise, to identify them we have to decide a threshold T so that such pixels are the ones for which the following relation holds :

$$\Delta L\,(x,y) > T$$

The final result of the method consists of a mainly black image in which the noisy pixels $P_N(x,y)$ are pointed out by white points (Fig. 2b). By applying the outlined median filter to only the pixels $P_N(x,y)$ we would have a better performance, in terms of signal to noise ratio, than the one reachable with the conventional median filter.

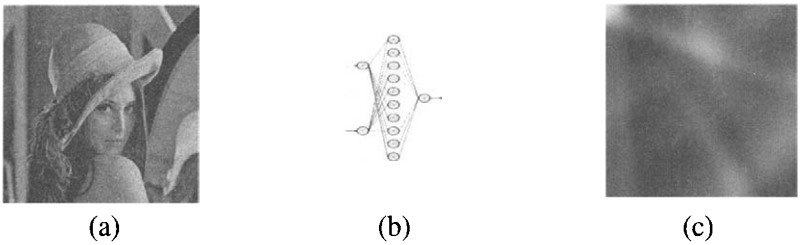


Fig. 1. (a) Corrupted image, (b) Neural Network used, (c) Image approximated by the neural network

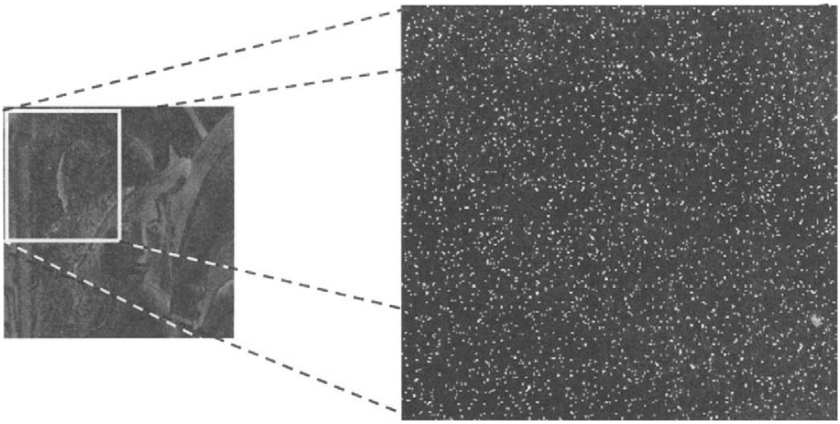


Fig. 2. (a) Error image, (b) Thresholded image related to a part of the error image

Since the value of the threshold *T* may influence significantly the performance of the proposed method, such decision is taken empirically by taking into account the mean value μ and the variance σ of the pixel luminosity, i.e.,:

$$\mu = \Sigma_{i,j}\ L(xi,yi) / N \qquad \sigma = (\Sigma_{i,j}\ (L(xi,yi) - \mu)^2 / N)^{1/2}$$

where N is the total number of the image pixels. In the next section we will propose a method to choose such threshold by a fuzzy logic approach, thus increasing the performance of the neural networks based filter.

3 Fuzzy Thresholding

To find the rules that may help in finding the optimal threshold for extract-ing the noisy pixels from a set of pixels potentially affected by noise, the Mamdani fuzzy controller may be used [5]. In fact, such a system works very well

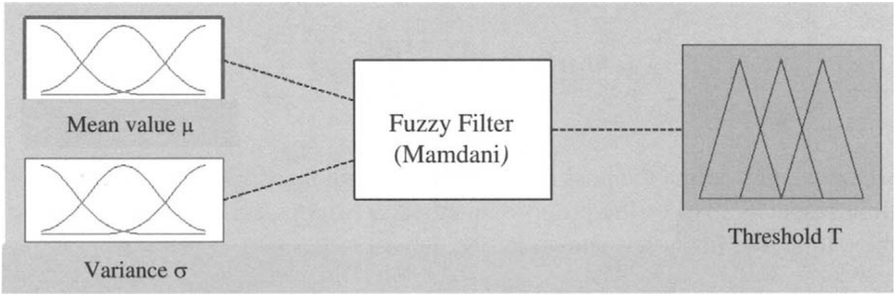


Fig. 3. Mamdani System to threshold error image

for identifying the fuzzy rules able to interrelate a set of known input-output pairs. In our case, the fuzzy controller (see Fig. 3) will receive, as inputs, the previously defined average value μ and the variance σ of the error image pixel luminosities, whereas the output variable has to be the value T to threshold the error image.

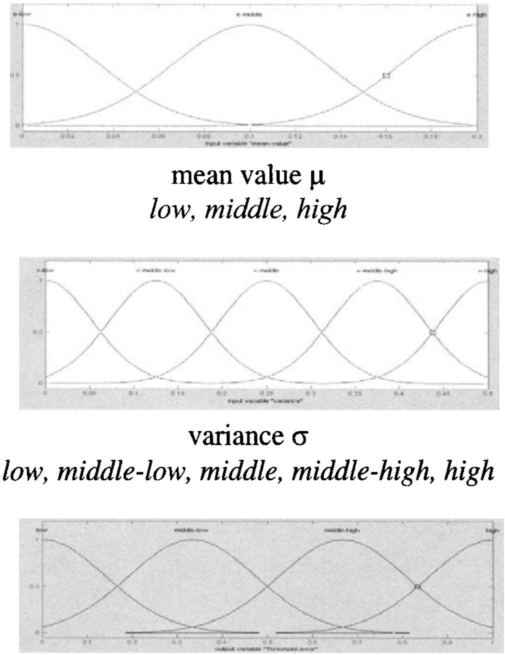
However, whereas $\mu(i)$ and $\sigma(i)$ may be easily computed from the images, to find the threshold $T(i)$ we should proceed according to the following procedure:

- take a set S of different images $I(i)$ characterized by different mean luminosities $\mu(i)$ and variances $\sigma(i)$
- add a salt and pepper noise to the previous images thus obtaining the images $I_N(i)$
- apply the method outlined in Sect. 2 thus obtaining from images $I_N(i)$ the image $M(i)$ consisting, as the one drawn in Fig. 2.b, of a wide black area containing some white pixels representing the pixels potentially affected by the noise
- find the threshold $T(i)$ that allow us for each image $I_N(i)$ to identify more or less all the noisy pixels

The above {input-output} pairs $\{\mu(i), \sigma(i) - T(i)\}$, are useful to train the Mamdani fuzzy controller in order to automatically find the set of fuzzy rules able to identify the unknown threshold for any image that has not been processed during the training phase. In order to implement the fuzzy thresholding, the following two-parameters Gaussian type membership function is used:

$$m(u)=\exp\left(-\left(\left(\frac{u-c}{s}\right)^{2}\right)\right)$$

where c and s define the position of the centre and the slope of Gaussian function, respectively. For the proposed method, we adopt, for the two inputs and the output, the fuzzy sets shown in Fig. 4.



mean value μ
low, middle, high

variance σ
low, middle-low, middle, middle-high, high

threshold T
low, middle-low, middle high, high

Fig. 4 Fuzzy sets for the mean value, the variance, and the threshold.

After having trained the Mamdani fuzzy controller with a set of training images, the following rules to threshold the error image have been obtained:
IF (μ is low) AND (σ is low) THEN (threshold is low)
IF (μ is low) AND (σ is middle-low) THEN (threshold is low)
IF (μ is low) AND (σ is middle-high) THEN (threshold is middle-high)
IF (μ is middle) AND (σ is middle) THEN (threshold is middle-low)
IF (μ is middle) AND (σ is middle-high) THEN (threshold is middle-high)
IF (μ is middle) AND (σ is high) THEN (threshold is high)

IF (μ is high) AND (σ is middle-low) THEN (threshold is middle-high)
IF (μ is high) AND (variance is middle-high) THEN (threshold is high)

Figure 5 points out how these rules allows us to compute the threshold starting from mean-value and variance. It is easy for the user to slightly modify these rules or to add new ones depending on her/his experience. This could be done, for example, modifying the previous rules by using adjectives such as *quite*, *most*, *essentially* and *little*, e.g.: *IF (mean-value is low or quite medium) AND (variance is essentially low) THEN (threshold is little low).* To compute the membership functions of the modified rules, the interested reader may consult [6].

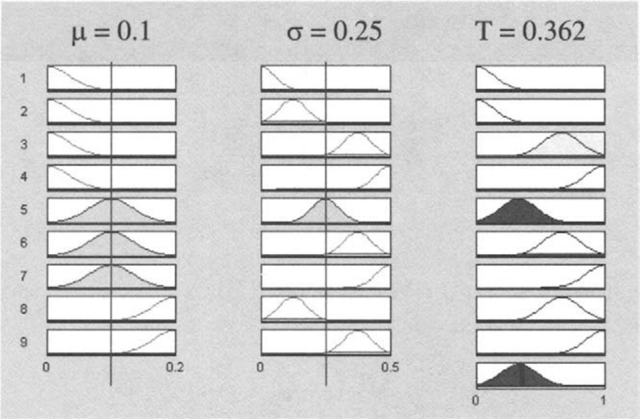


Fig. 5 Fuzzy rules dealing with mean value, variance, and the output.

To speed up the process of finding the threshold it is suitable to compute in advance the function T(μ(i), σ(i)). Figure 6 shows the function we have obtained according to the outlined procedure.

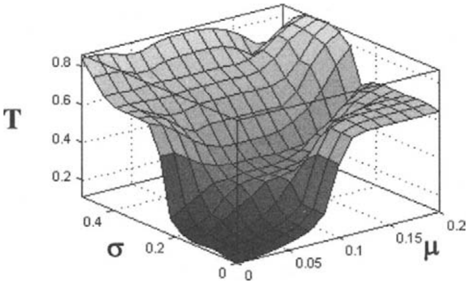


Fig. 6. Fuzzy control surface

4 Results of the Image Pre-Processing

The noise dealt with our algorithms is only the salt and pepper impulse noise, where the corrupted pixels take on the value of either 0 or 255 with equal probability. The percentage of noisy pixels R is used to indicate how much an image is corrupted. For example, if an image is corrupted by R=10% impulse noise, then 10% of the pixels in the image are corrupted randomly by positive and negative impulses. The algorithm has been tested on the Lena Image shown in Fig. 7.

Fig. 7. Original Lena image

Peak signal-to-noise ratio (PSNR) is used to evaluate the restoration performance. PSNR is defined as:

$$PSNR = 10\log_{10}\left\{\frac{255^2}{\frac{1}{MN}\sum(x_{i,j} - y_{i,j})^2}\right\}$$

where $x_{i,j}$ e $y_{i,j}$ are the coordinates of the pixels P(i,j) of the source image and the restored image respectively. The image size is MxN and 8 bit/pixel. Figure 8 shows the results using the median filter, the iterative (10 iterations) median filter which, as is known, increases the performance of the median filter and the proposed filter, for an image corrupted with salt and pepper noise with R=10%. The simple median filter can preserve the image details but many noisy pixels remain in the image. The iterative median filter removes most of the impulses, but many good pixels are also removed. The proposed filter removes almost all the pixels that are really affected by noise.

The PSNR performances provided in Table 1 show that by our approach we obtain a significant increase of performances and that the decrease of PSRN is little influenced by the increase of the noise level. The advantage of using our approach is particularly evident if one considers the performances of a sophisticated algorithm of type III, such as the one presented in [4]. In fact, in this algorithm PSRN(30% of noise) = 36.5 and PSRN(50% of noise) = 33, whereas in our algorithm PSNR is about 38.3 and 37.2 respectively.

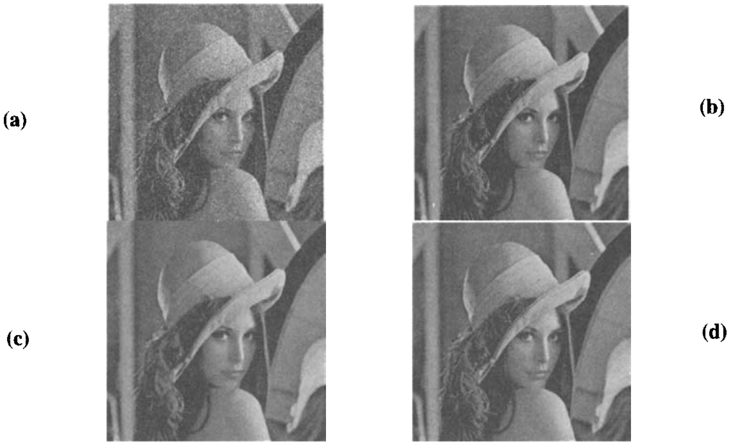


Fig. 8. Corrupted Lena image (a), and outputs of: median filter with 3x3 window (b), ten iterations of median filter (c), and proposed algorithm (d)

Table 1. PSNR of the restored images for different noise levels by the median filter and by our algorithm. The last column refers to a double iterations of our algorithm.

% Noise	Median Filter	Iterative Median Filter	Proposed Algorithm	Iterative version of the proposed Algorithm
10	35.497	34.551	37.881	38.984
30	32.155	31.811	35.537	38.342
50	30.708	30.456	33.410	37.191

To evaluate the performances of the algorithm we also use the SAD (sum of absolute difference) between the original image and the filtered image. Such parameter gives an idea on how much the details of the original image are preserved. SAD is defined as follows:

$$SAD = \frac{1}{NxM}\sum\sum \left|u(i,j) - y(i,j)\right|$$

where $u_{i,j}$ e $y_{i,j}$ are the coordinates of the pixels P(i,j) of the original image and the restored image respectively. For the image Lena the results shown in Table 2 demonstrate the good performances achieved by our algorithm also for preserving the details of the image. Figure 9 confirms qualitatively this result by showing an image in which edges and local features play a role more relevant than the one they play in the Lena image denoised by our algorithm. PSRN and SAD related to this case show the same figures of the previous ones, thus it could not be necessary to use objective functions (e.g., the one

Table 2. SAD of the restored images for different noise levels by the median filter and our algorithm. The last column refers to a double iterations of our algorithm.

% Noise	Median Filter	Iterative Median Filter	Proposed Algorithm	Iterative version of the proposed Algorithm
10	1.004	1.580	0.407	0.180
30	1.741	1.816	1.289	0.340
50	4.003	2.075	1.975	0.410

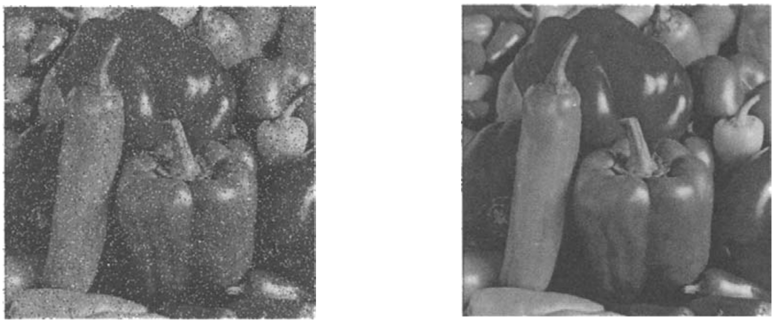


Fig. 9. Corrupted peppers image with 10% noise on the left and output of the iterative proposed algorithm on the right

proposed in [2]) instead of the median filter for preserving the details of the images. In this way we can avoid to significantly increase the processing time being the objective function approaches more time consuming than the ones based on the median filter [4].

A complete discussion on the processing time is for further study. However, let us note that the CPU time needed for processing, by a PC equipped with a 2 GHz CPU, the Lena and the peppers images covered by 50% of noise is about 1000 seconds even if the noise increases to 70–90%. This execution time is less than the time (i.e., 2009 seconds) needed by the algorithm III at 70% of noise (see [4]). Moreover, the execution time needed by our algorithm does not depend on the noise levels neither on the image complexity, whereas the one required by the algorithm III is highly influenced by these factors (e.g. it passes from 2009 to 6917 seconds when the noise increases from 70% to 90%, and from 2009 seconds for denoising the Lena image to 4263 seconds for denoising the image of a bridge). We don't have information about the time needed by the algorithm III at lower levels of noise. Presumably its execution time should decrease. However, it is likely that the execution time of our algorithm will still be better because it is considerably less than the one of algorithm III at 70% of noise, and for lower noise levels it could further decrease at the expenses of a PSRN lower than the one currently attained by using a lower number of epochs during the learning phase of the neural net.

5 Concluding Remarks

A novel type of filter based on neural networks and fuzzy logic has been proposed and applied to noise reduction in digital images. In this filter, we use the neural network to approximate the function *L(x,y)* that represents the image luminosity. Next, we use fuzzy logic to threshold the error image, given by the difference between the original image and the approximated one, thus obtaining the pixels that are really affected by the salt and pepper noise. Experiments indicate that our method provides a significant improvement over the median filter. A comparison with other algorithms proposed in literature has shown that our approach outperforms the available algorithms in both PSRN and SAD, thus obtaining images that are very close to the original ones even if the images show relevant edges. The time performance of our approach is better than the one of the existing algorithms at the same noise levels. Its time performance does not depend on the noise level neither on the image complexity, whereas the one of the other algorithms are highly influenced by these factors. Better PSRN and SAD could be achieved by a repetitive use of the proposed filter (see last column of tables 1 and 2). Since this increases the processing time, we are evaluating if and when this is acceptable or if a GRID based implementation may be more suitable for the problem at hand.

References

[1] Gonzales R.C.and Woods R.E., (2002): *Digital image processing.* Prentice Hall

[2] Nikolova, M. (2004): *A variational approach to remove outliers and impulse noise*, Journal of Mathematical Imaging and Vision 20 (2004), pp. 99-120

[3] Hwang H., Haddad, R.A., (1995): *Adaptive median filters: anew algorithms and results*, IEEE Transactions on Image Processing 4 (1995) pp. 499-502

[4] Chan R.H., Ho C.W. Nikolova M. (2005): *Salt and pepper noise removal by median type noise detectors and detail-preserving regularization* – www.math.cuhk.edu.hk/~rchan/paper/impulse/impulse.pdf to appear on IEEE Transactions on Image processing

[5] Mamdani, E. H., Assilian, S. (1975): *An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller.* Int. Journal of Man-Machine Studies, 7 (1) pp. 1-13

[6] Wang P.P. (Ed.) (2001): *Computing With Words.* Wiley

Computing Optimized NURBS Curves using Simulated Evolution on Control Parameters

Muhammad Sarfraz, Sadiq M. Sait, Mohmmed Balah, and M. Humayun Baig

College of Computer Sciences and Engineering
King Fahd University of Petroleum and Minerals
KFUPM#1510, Dhahran 31261, Saudi Arabia
{sarfraz, sadiq, mbalah, humayun}@cccse.kfupm.edu.sa

Abstract. In curve fitting problems, the selection of appropriate parameters in order to get an optimized curve for a shape design, is well-known. For large data, this problem needs to be dealt with optimization algorithms avoiding possible local optima and at the same time getting to the desired solution in an iterative fashion. Many evolutionary optimization techniques like genetic algorithm, simulated annealing have already been successfully applied to the problem. This paper presents an application of another evolutionary heuristic technique known as "Simulated Evolution" (SimE) to the curve fitting problem using NURBS. The shape parameters, in the description of NURBS, have been targeted to be optimized in a best possible way. The paper describes the mapping scheme of the problem to SimE followed by the proposed algorithm's outline with the results obtained.

Keywords: curve fitting, NURBS, approximation, simulated evolution, algorithm

1 Introduction

In planar shape design problems, the main objective is to achieve an optimized curve with the least possible computation cost. For complicated shapes with large measurement data, the problem becomes dependent upon the selection of appropriate parameters in the description of the curve model. Algorithms based on heuristic techniques like genetic algorithms, simulated annealing, simulated evolution (SimE) etc., can provide us with an approach in finding optimal parameters with reasonable cost. Since the data in such a problem cannot be approximated with a single polynomial, the application of splines, Bezier curves etc., are well known. Non-uniform Rational B-splines (NURBS) [4], providing more local control on the shape of the curve, gives a better approximation of the underlying data in shape design problems. In [9], knots corresponding to the control points have been optimized using a genetic algorithm. An approach based on Tabu search has been applied in [14]. An algorithm proposed in [12] discusses optimization of knots and weights using Simulated Annealing. In [10] proposed a novel approach for opti-mization of NURBS control points using simulated evolution. The main contribution of this work is to propose a curve fitting algorithm based on SimE using NURBS. Here, the shape parameters in the description of the NURBS have been selected to be computed such that the computed curve provides a best fit to the original data raised from the planar images after the scanning process.

The paper has been designed into various sections. The following section deals with the image contour extraction. Detection of significant points have been reported in Section 3. Section 4 gives a brief description of NURBS whereas the SimE algorithm has been discussed in Section 5. The proposed approached, with details of the evolutionary optimization curve technique SimE, has been explored and designed in Section 6. Demonstration of the executed results is given in Section 7 and Section 8 concludes the paper.

2 Image Contour Extraction

A digitized image is obtained from an electronic device or by scanning an image. The quality of digitized scanned image depends of various factors such as the image on paper, scanner type and the attributes set during scanning. The contour of the digitized image is extracted using the boundary detection algorithms. There are numerous algorithms for detecting boundary. We used the algorithm proposed by [8]. The input to this algorithm is a bitmap file. The algorithm returns a number of boundary points and their values.

3 Detection of Significant Points

Detection of significant points is the next step after finding out contour of the image. The significant points are those points, which partition the outline into various segments. Each segment is considered to be an element in our proposed approach. A number of approaches have been proposed by researchers [2]. In this paper, the detection of corner points has been implemented using the technique presented by Chetverikov and Szabo [2]. In [2] corner point is defined as a point where triangle of specified angle can be inscribed within specified distance from its neighbor points. It is a two pass algorithm. In the first pass the algorithm scans the sequence and selects candidate corner points. The second pass is post-processing to remove superfluous candidates.

4 NURBS

A unified mathematical formulation of NURBS provides free form curves and surfaces. NURBS contains a large number of control variable, because of those variables it is flexible and powerful. NURBS is a rational combination of a set piecewise rational polynomial of basis functions of the form

$$S(u) = \frac{\sum_{i=1}^{n} p_i w_i B_{i,k}(u)}{\sum_{i=1}^{n} w_i B_{i,k}(u)} \tag{1}$$

where p_i are the control points and w_i represent the associated weights. u is the parametric variable and $B_{i,k}(u)$ is B-spline basis function [6]. Assuming basis function of order k (degree $k-1$), a NURBS curve has $n+k$ knots and the number of control points equals to weights. The knot set $\{u_i\}$ is a non-decreasing sequence: $u_1 \le u_2 \le ... \le u_{n+k-1} \le u_{n+k}$. The parametric domain is $u_k \le u \le u_{k+1}$. NURBS include weights as extra degrees of freedom, which are used for geometric design [5–7].

5 Outline of Simulated Evolution (SimE)

SimE is a powerful general iterative heuristic for solving combinatorial optimization problems [11, 13]. The algorithm consists of three basic steps: Evaluation, Selection and Allocation. These three steps are executed sequentially for a prefixed number of iterations or until a desired improvement in goodness is observed. The SimE algorithm starts with an initial assignment, and then seeks to reach better assignments from one generation to the next. SimE assumes that there exists a population P of a set M of n elements. A cost function is used to associate with each assignment of element m a cost C_m. The cost C_m is used to compute the goodness g_m of element m for each $m \in M$.

The selection step partitions the elements into two disjoint sets P_s and P_r based on their goodness. The elements with bad goodness are selected in the set P_s and the rest of elements in the set P_r. The non-deterministic selection operator takes as input the goodness of each element and a parameter B, a selection Bias. Hence the element with high goodness still has a non-zero probability of being assigned to the selected set P_s. The value of the bias is application dependent. In our case the value of B has been taken as -0.5.

The Allocation step takes P_s and P_r and generates a new solution P' which contains all the members of the previous population P. The members of P_s are then worked upon so that their goodness could be enhanced in the subsequent iterations. The choice of a suitable allocation function is problem dependent [13].

6 Proposed Approach

The proposed approach to the problem is described here in detail.

6.1 Problem Mapping

This section describes about the SE formulation of the current problem in detail. In curve fitting problems, the solution space consists of the number of data points on the image boundary. The contour of the digitized image is extracted using the boundary detection algorithms. There are numerous algorithms for detecting boundary. We used the algorithm proposed by Quddus [8]. The input to this algorithm is a bitmap file and output is, number of boundary points and their values.

6.1.1 Initialization

Initialization is the first step in SE. It consists of selecting a starting solution for the problem under consideration. This solution can be generated randomly or the output of any constructive heuristic. In our case from the boundary points, initial solution is created using corner detection algorithm.

Corner detection algorithm divides the given image in to segments. The corner points are the only end points of the segment (S). For example if there is n number of corner points we have n-1 segments. For each segment we need to calculate the parameters t, control points, knot vector and the weight of NURBS. The initial solution of weight vector is randomly selected from the range [0, 1]. The number of elements in the weight vector corresponds to the number of control points. The values of the parameters t for each segment are calculated using centripetal method. The number of control points of the segment are always equals to the order of the NURBS curve. In our proposed approach we have included the two corner points of the segment as the control points and the remaining control points are determined using least square method. Weight corresponding to each control point of a segment is taken randomly between 0 and 1. After calculating required parameters for each segment, curve is fitted using NURBS. This fitted curve for each segment is considered as the initial solution for SE. The other important parameters which are initialized in this step are a stopping condition and selection bias (B).we have taken selection bias (B) in the range of [-.1,1] and fixed number of iterations.

6.1.2 Evaluation

In this step, each individual segment of the curve is evaluated on the basis of goodness. The goodness g_i of each segment S_i is determined by

$$g_i = \frac{\varepsilon}{Q_i + \varepsilon}, \ \varepsilon \geq 1 \tag{2}$$

This criterian is different than that discussed in [10]. In this research, we have taken $\varepsilon = (l_i + k)$ where l_i is the length of the knot vector for each segment, k is the order of the curve and Q_i is sum square error between the target and the fitted curve. Q_i can be defined as

$$Q_i = \sum_{1}^{N} \{(S(u_i) - F_i)^2\}. \tag{3}$$

Here $S(u_i)$ is the approximated curve and $\{F_i\}$ is the target curve data. N is the total number of data points in each segment. The goodness g_i represents a measure of how near each segment is to its optimum curve fit. As is obvious from Eqn. (2), the goodness of an element is between 0 and 1. The value of goodness g_i nearer to 1 means that segment i is nearer to its optimum curve fitting.

6.1.3 Selection

The goodness g_i is used to probabilistically select segments (S_i) in the selection step. On the basis of the goodness g_i, the selection function partitions the segments in to two sets P_r and P_s probabilistically. Selection function is defined as follows:

```
If (Random [0, 1] <= $1 - g_i + B$)    then
                                           $P_s = P_s\ U\{S_i\}$
                                       Else
                                           $P_r = P_r\ U\{S_i\}$
```

Set P_s contain the segments with low or bad goodness and the set P_r contains the rest of the segments. We have taken selection bias (B) in the range of [0, -0.1].

6.1.4 Allocation and Weight Optimization

The purpose of the allocation is to perturb the current solution in such a way that it reaches the optimum solution. In our case optimum solution is achieving smooth curve with least error. After fitting the initial curve, further refinement of curve have to be done to achieve better fitting accuracy. For this purpose, different NURBS parameters have to be changed.

The allocation step for optimizing weights is to perturb the current solution of weights by assigning the selected segments in P_s to new values. In our case we perturb the weights in neighborhood of $[w_i^{cur}, w_i^{cur} + 0.5]$, where w_i^{cur} is the current weight corresponding to the control point i. For each segment 10 to 15 trail allocations for weight vector is made and for each trail error between the fitted curve and target is calculated. After performing all trails of perturbing weight vector, the trail with least error is made permanent and corresponding segment is removed from P_s. This process is repeated until all the segments in P_s are perturbed.

6.2 Algorithm Outline

The algorithm of the proposed scheme is contained on various steps as follows:

```
Begin
Step1: Input the digitized image.
Step2: Find the image contour.
Step3: Find the corner points
Step4: For each segment $S_i$
        a) Find control points from data points using the method least square technique.
        b) Find knot vector.
        c) Find weight vector corresponding to control points.
        d) Fit the curve with NURBS
    End for
Step5: Initialize population (Weights generated at step 4(c)) Pi, Bias value and number of iterations for SE
Step6: for $j=1$ to number of iterations (say 100)
        a) Evaluation
            For each segment $S_i$
                Find goodness ( $g_i$ )
            End for
        b) Selection
            For each segment $S_i$
                If (Random [0, 1] <=1- $g_i$ + B) then
                    $P_s = P_s \cup \{ S_i \}$
                Else
                    $P_r = P_r \cup \{ S_i \}$
            End for
        c) Allocation
            For each segment $S_i$ in $P_s$.
            Perturb the weight vector for 10 to 15 trails and choose the best one.
    End for
                $P = P_r \cup P_s$
    end for $j$
Step7: Return the final Fitted Curve with NURBS
End
```

The proposed algorithm is run for 100 iterations. The bias value is taken as -0.5. The segment with the least goodness out of the total selected segments is added a knot using the middle point criteria [3]. The algorithm converged at 29th iteration for character "*h*" (Fig. 1) and at 35th iteration for fork (Fig. 7) is shown in Fig. 6 and Fig. 12 respectively. The intermediate results are shown in the rest of the figures. Figures 2 and 7 are outlines whereas the corner points are detected in Figs. 3 and 8 for for the "*h*" and fork objects respectively. The Sum Square Error (SSE) between the boundary of the image and the NURBS fitted curve with respect to the number of iterations is as shown in Fig 13. The results obtained using the proposed algorithm are found to be better compared to the results obtained using the genetic algorithm and simulated annealing discussed in [9] and [11].

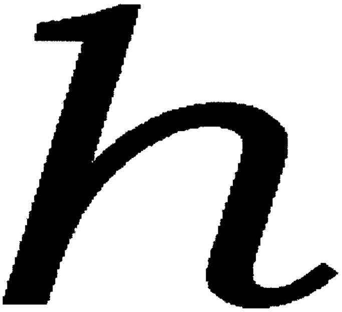


Fig. 1. Bitmap image of character '*h*'.

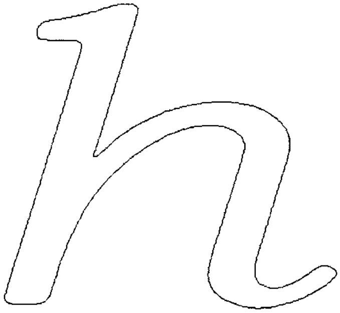


Fig. 2. Outline after boundary detection

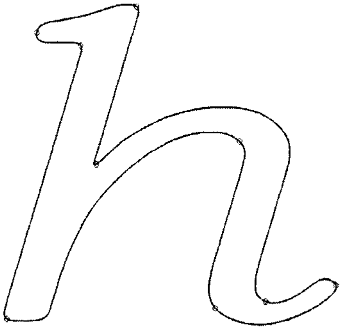


Fig. 3. Significant point detection

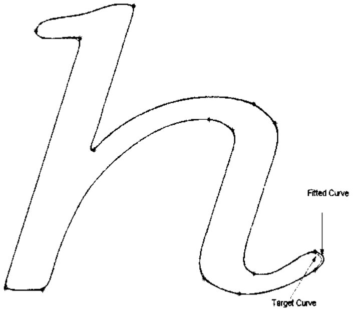


Fig. 4. NURBS Fitted object at First iteration

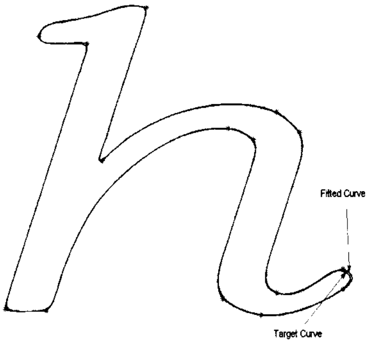


Fig. 5. NURBS Fitted object at 18th iteration.

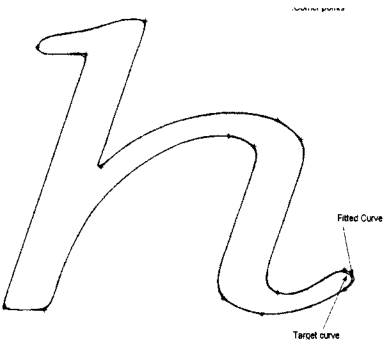


Fig. 6. NURBS Fitted object after final iteration.

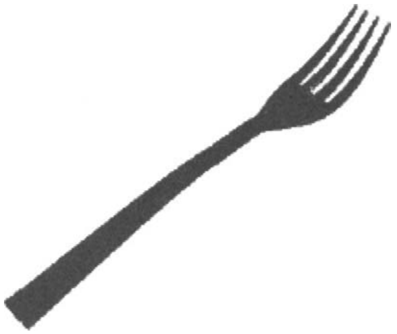


Fig. 7. Bitmap image of object (fork)

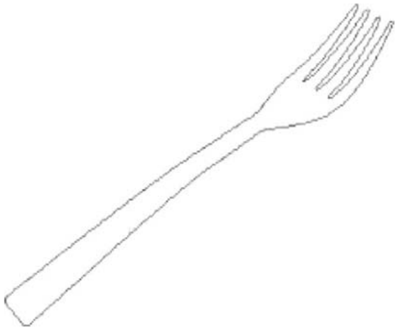


Fig. 8. The contour of the image obtained.

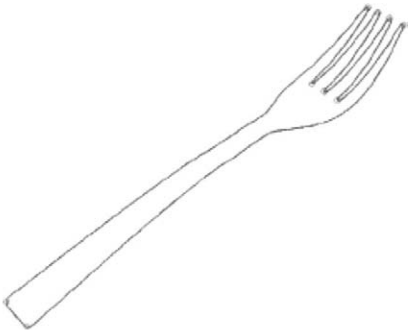


Fig. 9. Significant points obtained after Corner detection

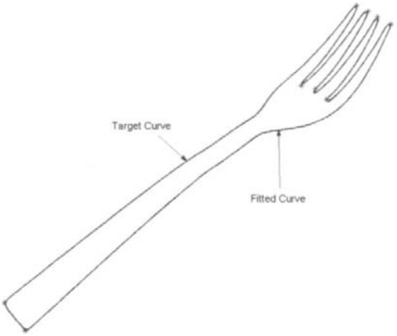


Fig. 10. NURBS Fitted object at First iteration.

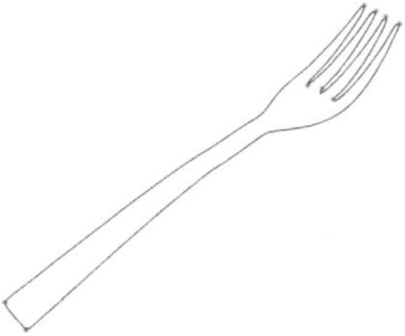


Fig. 11. NURBS Fitted object at 40th iteration

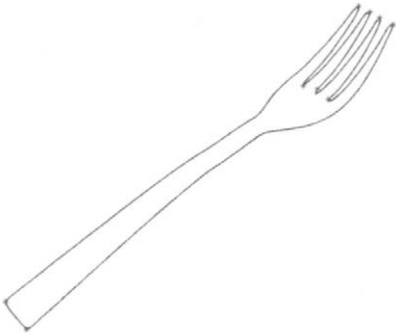


Fig. 12. NURBS Fitted object at final iteration.

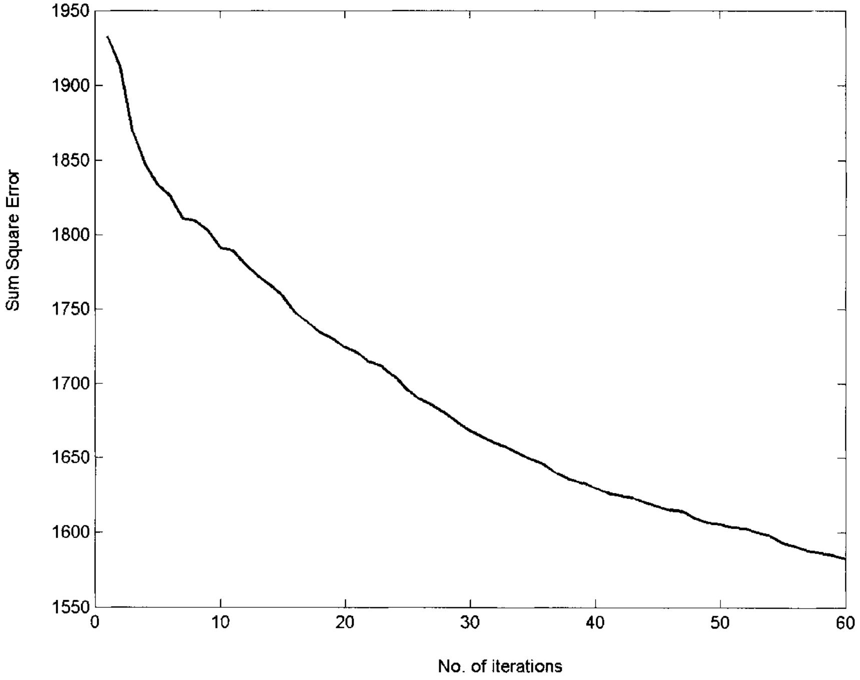


Fig. 13. SSE vs number of iterations

8 Conclusion

The proposed approach is effective in the determination of appropriate number of knots using NURBS. The corner detection algorithm provides an initial solution to start with. The subsequent iterations evolutionarily approach towards the desired solution. Quite pleasing results have been obtained in a comparable amount of time with the existing non-deterministic approaches in the literature. A detailed comparative study is under the study of the authors and is left for publication elsewhere.

Acknowledgments

The authors are thankful to the anonymous referees for the valuable suggestions towards the improvement of this manuscript. This work has been supported by the King Fahd University of Petroleum and Minerals for the project entitled "Reverse Engineering for Geometric Models Using Evolutionary Heuristics".

References

1. Akaike, H.: A new look at the statistical model identification," IEEE Transaction. Automatic Control, vol (1974) 716-723
2. Chetverikov, D., Szabo, Z.: simple and efficient algorithm for detection of high curvature points in planar curves. Proc. 23rd Workshop of the Australian Pattern Recognition Group. (1999) 175-184
3. Dierckx, P.: Curve and surface fitting with Splines. Clarendon Press (1993)
4. Farin, G.: From Conic to NURBS: A tutorial and survey. IEEE Computer Graphics and Applications. vol.12(5) (1992) 78-86
5. Farin, G.:Trends in curves and surface design. Computer-Aided Design, vol. 21(5) (1989) 293-296
6. Piegl, L., Tiller, W.: *The NURBS Book.* Springer-Verlag, New York, (1997)
7. Piegl, L.,Tiller, W.: Curve and surface reconstruction using rational B-splines. *Computer-Aided Design*, vol. 19(9) (1991) 485-498
8. Quddus, A.: Curvature Analysis Using Multi-resolution Techniques. PhD Thesis. Dept. Elect. Eng., King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia(1998)
9. Sarfraz, M., Raza, S. A.: Capturing Outline of Fonts using Genetic Algorithm and Splines. The Proceedings of IEEE International Conference on Information Visualization-IV'2001-UK, IEEE Computer Society Press, USA, (2001) 738-743
10. Sarfraz, M., Raza, S.A. and Baig, M.H., Computing Optimized Curves with NURBS Using Evolutionary Intelligence, *Lecture Notes in Computer Science Vol. 3480: Computational Science and Its Applications*, Eds.: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Springer-Verlag, ISSN 0302-9743, (2005) 806 - 815.
11. Ralph Michael kling, Benerjee, P.: Empirical and Theoretical studies of Simulated Evolution Method Applied to standard cell Placement. IEEE Transactions on computer Aided Design vol. 10(10) (1991)
12. Sarfraz, M., Riyazuddin, M. and Baig, M. H., Capturing Planar Shapes by Approximating their Outlines, International Journal of Computational and Applied Mathematics, Elsevier Science, (2005)
13. Sait, M.S., Youssef, H.: Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems. IEEE Computer Society Press, California (1999)
14. Youssef, M.: Reverse Engineering of Geometric Surfaces using Tabu Search Optimization Technique. Master Thesis, Cairo University, Egypt (2001)

Part II

Control and Robotics

Design of A Takagi-Sugeno Fuzzy Compensator for Inverted Pendulum Control Using Bode Plots

Xiaojun Ban[1], X. Z. Gao[2], Xianlin Huang[3], and H. S. Lin[4]

[1] Department of Control Theory and Engineering, Harbin Institute of Technology, Harbin, China, xiaojun1978@xinhuanet.com.
[2] Institute of Intelligent Power Electronics, Helsinki University of Technology, Espoo, Finland, gao@cc.hut.fi.
[3] Department of Control Theory and Engineering, Harbin Institute of Technology, Harbin, China, xlinhuang@hit.edu.cn
[4] Department of Control Theory and Engineering, Harbin Institute of Technology, Harbin, China, haishenglin318@yahoo.com.cn

Abstract. In our paper, the frequency-response method is combined with the fuzzy control theory to design a hybrid inverted pendulum control system. A Takagi-Sugeno (T-S) fuzzy compensator, which consists of four linear local compensators, is constructed by using the Bode plots. To take further advantage of the frequency-response method, an integrator is cascaded with each of the four local compensators to eliminate the effect of a constant friction on the cart. Performance comparison is also made between the T-S fuzzy controller and linear compensators. Simulation results demonstrate the considerable feasibility of our proposed approach.

Keywords: Fuzzy control, Takagi-Sugeno fuzzy controller, linear compensator, Bode plots, inverted pendulum.

1. Introduction

The Takagi-Sugeno (T-S) fuzzy model [7] is a well-known landmark in the history of fuzzy control theory. Numerous fuzzy control problems, such as stability analysis, systematic design, robustness, and optimality, can be addressed within the framework of the T-S model. Especially, given a T-S fuzzy model, a fuzzy controller design method named Parallel Dis-tributed Compensation (PDC), has been proposed by Sugeno and Kang [5]. The corresponding stability analysis of T-S fuzzy systems is widely discussed in [2,4,6,9]. The unique advantage of this technique is that a lot of conventional linear controller design solutions oriented from both classical and modern control theory, e.g., state space optimal control and robust control, can be employed in designing the T-S fuzzy controllers as well, which are actually nonlinear controllers [3].

As we know, the frequency-response method has been well-developed and widely used in industrial applications, which is straightforward and easy to follow by practising engineers. Worthy of mentioning, the effect of noise in a control system can be evaluated by its frequency response. This advantage is very useful for system analysis, since unavoidable noise usually deteriorates the overall control performance [1]. Therefore, fusion of the T-S fuzzy model and frequency-response method is of great significance in the perspective of control engineering.

Inspired by the above idea, in our paper, the frequency-response method is merged with fuzzy control theory for the inverted pendulum controller design. Four local linear compensators are first designed by using the Bode plots. A T-S fuzzy compensator is next constructed based on these compensators. An integrator is connected with each of the four local compensators to combat with the constant friction existing on the cart. Moreover, performance comparison is made between this T-S fuzzy compensator and linear controllers.

Our paper is organized as follows. In Section 2, four linear local models are derived from the original nonlinear differential equations of the inverted pendulum. As an example, a linear compensator is synthesized in more details in the next section. Based on the four local compensators designed, the T-S fuzzy compensator is constructed in Section 4. Finally, simulations and conclusions are given in Sections 5 and 6, respectively.

2. Mathematical Models of Inverted Pendulum

The nonlinear model of an inverted pendulum is given as the following differential equations:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{g\sin(x_1(t)) - amlx_2^2(t)\sin(2x_1(t))/2 - a\cos(x_1(t))u(t)}{4l/3 - aml\cos^2(x_1(t))}, \end{aligned} \tag{1}$$

where $x_1(t)$ represents the vertical angle of the pendulum, $x_2(t)$ is the angular velocity, $g = 9.8\,m/s^2$ is the gravity constant, m is the pendulum mass, M is the cart mass, l is half length of the pendulum, u is the force exerted on the cart, and $a = 1/(m + M)$. In our simulations, these parameters are chosen as $m = 2.0\,kg$, $M = 8.0\,kg$, and $l = 0.5\,m$ [8].

To design a compensator for the inverted pendulum in the frequency domain, four linear models are first acquired based on (1). More precisely, when $x_1(t)$ is about zero and $x_2(t)$ is zero, the nonlinear model of our inverted pendulum can be linearized as follows:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{gx_1(t) - au(t)}{4l/3 - aml} \cdot \end{aligned} \tag{2}$$

When $x_1(t)$ is about $\pm\frac{\pi}{8}$, $\pm\frac{\pi}{4}$, $\pm\frac{3\pi}{8}$, and $x_2(t)$ is zero, the corresponding linear models are shown from (3) to (5), respectively:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{g\frac{8}{\pi}\sin(\frac{\pi}{8})x_1(t) - a\cos(\frac{\pi}{8})u(t)}{4l/3 - aml\cos^2(\frac{\pi}{8})}, \end{aligned} \tag{3}$$

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{g\frac{4}{\pi}\sin(\frac{\pi}{4})x_1(t) - a\cos(\frac{\pi}{4})u(t)}{4l/3 - aml\cos^2(\frac{\pi}{4})}, \end{aligned} \tag{4}$$

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{g\frac{8}{3\pi}\sin(\frac{3\pi}{8})x_1(t) - a\cos(\frac{3\pi}{8})u(t)}{4l/3 - aml\cos^2(\frac{3\pi}{8})}. \end{aligned} \tag{5}$$

3. Local Compensator Design Using Bode Plots

In this section, four local linear compensators are designed based on the above linear models in the frequency domain. When $x_1(t)$ is about zero, the transfer function can be derived from (2):

$$G_1(s) = \frac{-3}{17s^2 - 294}. \tag{6}$$

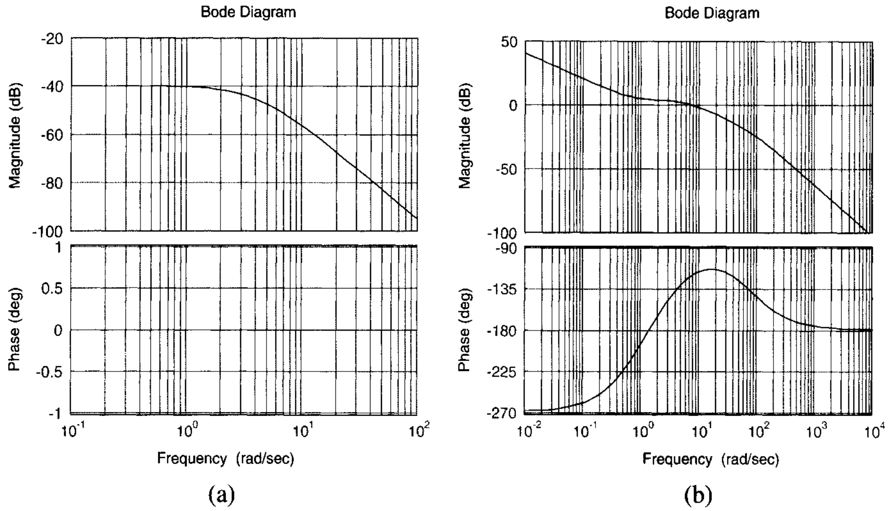


Fig. 1. Bode plots of inverted pendulum
(a) without the compensator.
(b) with the final compensator $C_1(s)$.

It is observed from $G_1(s)$ that there is a pole on the right half of the complex plane, which definitely makes the inverted pendulum unstable. The corresponding Bode plot is shown in Fig. 1(a). Employing the Nyquist's stability criterion, we utilize a lead compensator in (7) to stabilize the inverted pendulum:

$$C_1^1(s) = K(\tau s + 1), \tag{7}$$

where $K = -138$, and $\tau = \dfrac{1}{2.7}$. To eliminate some constant disturbances, such as existing friction, an integrator is also introduced as shown below:

$$C_1^2(s) = \frac{s + \tau_1}{s}, \tag{8}$$

where τ_1 is 0.724, which is ten times less than the crossover frequency. In addition, a first order compensator in (9) is exploited here to suppress the high-frequency noise:

$$C_1^3(s) = \frac{1}{\tau_2 s + 1}, \tag{9}$$

where τ_2 is 1/80.

Cascading all the above three compensators, a linear compensator is obtained:

$$C_1(s)=\frac{K(\tau s+1)(s+\tau_1)}{s(\tau_2 s+1)}=\frac{-138(\frac{1}{2.7}s+1)(s+0.724)}{s(\frac{1}{80}s+1)}. \qquad (10)$$

The final Bode plot is shown in Fig. 1(b). The phase margin and the crossover frequency are now 58.7° and 7.24 rad/sec, respectively. It can be concluded from these two criteria that the performance of the overall inverted pendulum control system is satisfactory.

Similar with $G_1(s)$, when $x_1(t)$ is about $\pm\frac{\pi}{8}, \pm\frac{\pi}{4}, \pm\frac{3\pi}{8}$, the local compensators are designed as follows:

$$C_2(s)=\frac{-141.3(\frac{1}{2.45}s+1)(s+0.744)}{s(\frac{1}{80}s+1)}. \qquad (11)$$

$$C_3(s)=\frac{-175.8(\frac{1}{2.4}s+1)(s+0.68)}{s(\frac{1}{70}s+1)}, \qquad (12)$$

$$C_4(s)=\frac{-285.1(\frac{1}{2.21}s+1)(s+0.612)}{s(\frac{1}{65}s+1)}, \qquad (13)$$

4. Design of T-S Fuzzy Compensator

Based on the above four local compensators, a T-S fuzzy compensator is designed in this section, which can be described by the following four rules.

If $x_1(t)$ is "about zero", then $u(t)$ should be calculated according to

$$C_1(s)=\frac{-138(\frac{1}{2.7}s+1)(s+0.724)}{s(\frac{1}{80}s+1)},$$

If $x_1(t)$ is "about $\pm\frac{\pi}{8}$", then $u(t)$ should be calculated according to

$$C_2(s) = \frac{-141.3(\frac{1}{2.45}s+1)(s+0.744)}{s(\frac{1}{80}s+1)},$$

If $x_1(t)$ is "about $\pm\frac{\pi}{4}$", then $u(t)$ should be calculated according to

$$C_3(s) = \frac{-175.8(\frac{1}{2.4}s+1)(s+0.68)}{s(\frac{1}{70}s+1)},$$

If $x_1(t)$ is "about $\pm\frac{3\pi}{8}$", then $u(t)$ should be calculated according to

$$C_4(s) = \frac{-285.1(\frac{1}{2.21}s+1)(s+0.612)}{s(\frac{1}{65}s+1)},$$

where "about zero", "about $\pm\frac{\pi}{8}$", "about $\pm\frac{\pi}{4}$", and "about $\pm\frac{3\pi}{8}$" are all linguistic values that are denoted as A_i, $i=1,\cdots 4$. In our paper, they are quantified by the triangular membership functions, as shown in Fig. 2. The configuration of our T-S fuzzy compensator is illustrated in Fig. 3.

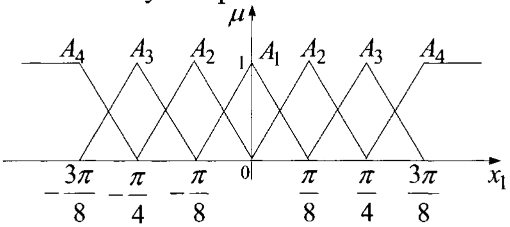


Fig. 2. Membership functions of $A_i, i=1,\cdots 4$.

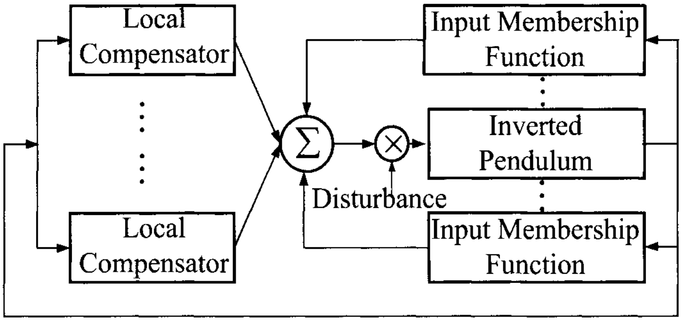


Fig. 3. Configuration of the T-S compensator

The final control output is given in (14):

$$u(t)=\frac{\mu_1(x_1)u_1(t)+\mu_2(x_1)u_2(t)+\mu_3(x_1)u_3(t)+\mu_4(x_1)u_4(t)}{\mu_1(x_1)+\mu_2(x_1)+\mu_3(x_1)+\mu_4(x_1)}, \tag{14}$$

where $\mu_i(x_1)$, $i=1,\cdots 4$ are the degrees of membership calculated from A_i, $i=1,\cdots 4$, respectively, and $u_i(x_1)$, $i=1,\cdots 4$ are the local control outputs of those four local compensators.

5. Simulations

Firstly, performance comparison is made between our T-S fuzzy compensator and the local compensator synthesized for the linearized model in case of $x_1(t)$ is "about zero". We focus on both the region of attraction and maximum outputs of compensators. The results are given in Table 1.

Table 1. Comparison between T-S fuzzy and linear compensators.

Initial angle	Maximum output of compensators		Stable/unstable	
	T-S	Linear	T-S	Linear
5	367	357	Yes	Yes
15	1162	1071	Yes	Yes
25	2038	1784	Yes	Yes
35	2993	2498	Yes	Yes
45	4027	3211	Yes	Yes
55	6312	3925	Yes	Yes
65	9102	4639	Yes	Yes
75	10976	5352	Yes	Yes
80	11708	5709	Yes	Yes
81	11854	-	Yes	No
82	12001	-	Yes	No
87	12733	-	Yes	No
88	-	-	No	No

Secondly, the T-S compensator is compared with the linear compensator designed based on the linearized model at the linearization point $(\pm\frac{3\pi}{8},0)$. The results are given in Table 2.

Table 2. Comparison between T-S fuzzy and linear compensator.

Initial angle	Maximum output of compensators		Stable/Unstable	
	T-S	Linear	T-S	Linear
5	367	732	Yes	Yes
15	1162	2195	Yes	Yes
25	2038	3659	Yes	Yes
35	2993	5122	Yes	Yes
45	4027	6586	Yes	Yes
55	6312	8049	Yes	Yes
65	9102	9513	Yes	Yes
75	10976	10976	Yes	Yes
80	11708	11708	Yes	Yes
81	11854	11854	Yes	Yes
82	12001	12001	Yes	Yes
87	12733	12733	Yes	Yes
88	-	-	No	No

Finally, simulations are performed when a constant friction is exerted on the inverted pendulum. The T-S fuzzy compensator and the linear compensator for the equilibrium point are compared. The results are shown in Figs. 4 and 5, where the initial angles are 5°, 80°, respectively. The solid line represents the performance of the linear compensator, and the dotted line the T-S fuzzy compensator.

Some remarks and conclusions are drawn as follows.

(a) Compared with the linear compensator synthesized for the zero point, the region of attraction of the T-S fuzzy compensator is larger. In this case, the control output of the T-S compensator is also larger than that of the linear compensator.

(b) When the initial angle is small, the output of the T-S fuzzy compensator is much smaller than that of the linear compensator, which is derived from the linearization point $(\frac{3\pi}{8}, 0)$.

(c) The integrator cascaded with the T-S fuzzy compensator can efficiently eliminate the negative effects caused by the constant disturbance on the pendulum cart.

(d) It can be concluded from Figs. 4 and 5 that the performances of the T-S fuzzy compensator and linear compensator are almost the same, when the initial angle is small. However, the T-S compensator significantly outperforms the linear one, while the initial angle becomes larger.

6. Conclusions

In our paper, a T-S fuzzy compensator for the inverted pendulum control is designed based on the frequency-response method. This fuzzy compensator consists of four linear local compensators, which are synthesized by using the Bode plots. To eliminate the effect of constant friction existing on the pendulum cart, an integrator is cascaded with each of the local compensators. Simulations show that our T-S fuzzy controller is an 'interpolation' between the linear compensators for the zero point and linearization point $(\frac{3\pi}{8}, 0)$. We can conclude that the classical frequency-response method and modern fuzzy control theory can be merged together to develop new and feasible controllers with superior performances. The fusion technique can certainly benefit from both these two methods.

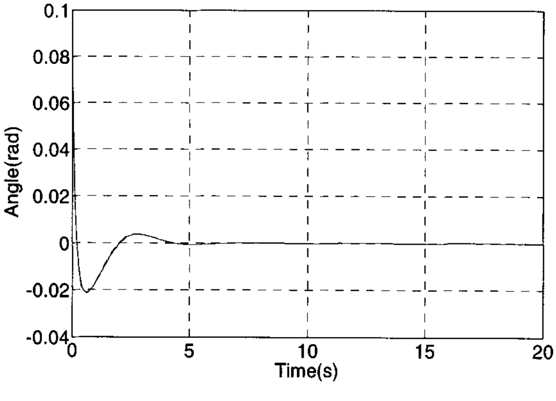


Fig. 4. Angle trajectory of inverted pendulum, when initial angle is 5°.

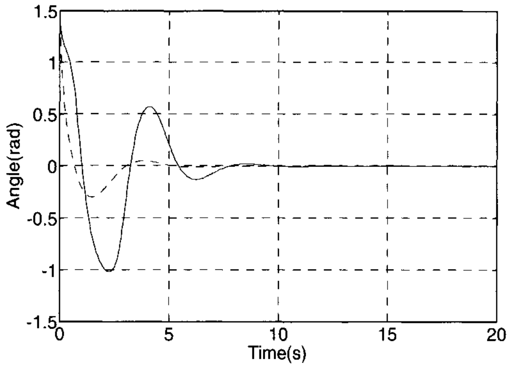


Fig. 5. Angle trajectory of inverted pendulum, when initial angle is 80°.

Acknowledgments

X. Z. Gao's research work was funded by the Academy of Finland under Grant 201353.

References

1. J J D'azzo, C H Houpis (1995) Linear control system analysis and design. McGraw-Hill Companies Inc, China
2. K Tanaka, M Sugeno (1992) Stability analysis and design of fuzzy control systems. Fuzzy Sets and Systems 45: 135–156
3. K Tanaka, H O Wang (2001) Fuzzy control systems design and analysis: a linear matrix inequality approach. Wiley-Interscience Publication, New York
4. L Luoh (2002) New stability analysis of T-S fuzzy system with robust approach. Mathematics and Computers in Simulation 59: 335–340
5. M Sugeno, G T Kang (1986) Fuzzy modeling and control of multilayer incinerator. Fuzzy Sets and Systems 18: 329–346
6. N Li, S Y Li (2004) Stability analysis and design of T-S fuzzy control system with simplified linear rule consequent. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 34:788–795
7. T Takagi, M Sugeno (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics 15: 116–132
8. W T Baumann, W J Rugh (1996) Feedback control of nonlinear systems by extended linearization. IEEE Transactions on Automatic Control 31: 40–46
9. X D Liu, Q L Zhang (2003) Approaches to quadratic stability conditions and H∞ control designs for T-S fuzzy systems. IEEE Transactions on Fuzzy Systems 11: 830–839

Soft Computing in Accuracy Enhancement of Machine Tools

Y. Liu[1], X. Z. Gao[2] and X. Wang[2]

[1] Bosch Automotive Diesel Systems Co., Ltd. RBCD/EVL1, 17 Xinhua Road, New District Wuxi 214028, P.R. China. (email: billly_1980@yahoo.com.cn)
[2] Institute of Intelligent Power Electronics, Helsinki University of Technology, Otakaari 5 A, FIN-02150 Espoo, Finland. (email: gao@cc.hut.fi, tel.: +358-9-4512434; xiaolei@cc.hut.fi.)

1 Introduction

The development of Computerized Numerical Control (CNC) machine tools has progressed and quickly played a significant role in modern manufacturing, due to the falling prices of small computers. Recently, the development trend of the CNC towards precision machining of parts with higher dimensional accuracy has become a necessity to satisfy ever-increasing demands of users. It is well known that the accuracy of samples machined on a CNC system is largely influenced by misalignment of machine axes, thermal deformation of machine structure, friction between moving components, etc. [1]. As long as machine errors are systematic or repeatable and can be measured and stored, "Software-based Error Compensation" can be taken as a cost-effective alternative to achieve improvement of the machine accuracy [2,3]. Three major error sources including friction induced errors, geometric errors, as well as thermal deformation errors are more concerned in error compensation techniques. Due to the complexity and uncertainty of practical machining processes, it is very difficult to understand internal properties about these three errors. Therefore, soft computing becomes a promising alternative to deal with them. Employment of soft computing in accuracy enhancement of machine tool can provide us significant adaptation, flexibility, and embedded linguistic knowledge.

In this overview, sections 2–4 discuss the applications of fuzzy logic, neural networks, and genetic algorithms in accuracy enhancement of machine tools, respectively. In each section, the motivations of applying soft computing method for individual field are first given. The effectiveness of the soft computing methods is illustrated, and their advantages and disadvantages are summarized. Finally, some perspectives about the future developments and applications of soft computing are given.

2 Fuzzy Logic for Friction Compensation

Friction characteristic has strong nonlinearity, and is commonly position-dependent as well as velocity-dependent. It has great impact on servo system dynamics especially for fine motions at low velocities. Techniques, such as adaptive controls, nonlinear controls, and repetitive controls [4], have been employed in friction compensation. Most of these existing methods are based on known friction models. However, in practical manufacturing processes, these extant friction models may be invalidated due to the increasing nonlinearities and rigorous working conditions. In contrast to the conventional control methods, fuzzy logic control (FLC) incorporates linguistic rules based on human intelligence or experts' knowledge, and endorses an effective means of capturing an approximate, inexact nature of friction phenomena.

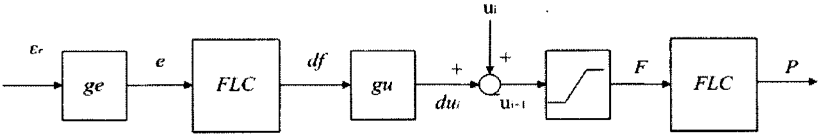


Fig. 1. Structure of one-input-one-output FLC.

For a CNC machine tool, an advanced servo-controller for single axis does not guarantee good overall performance on a multi-axes machine tool. Koren [5] proposed a cross-coupled control (CCC) for control of machine tool. Tarng et al. [6] applied fuzzy logic on the CCC design for a multi-axis CNC system as illustrated in Fig. 1. For real-time implementing friction compensation, contour error ε_r (difference between ideal contour P_r and actual contour P_a) at time $i+1$ should be estimated at previous step time i. The estimated contour error ε_r' is normalized within a range of $[-1,1]$ by multiplying scaling factor ge, and is fed into the FLC to generate an output df. Next, feedrate change du is generated by the product of output df and its scaling factor gu. As a result, feedrate u at time

$i+1$ is equal to $u_{i+1} = u_i + du_i$. In fact, real feedrate F is obtained by passing feedrate u through a limiter to guarantee that F is always smaller than the maximum federate $F_{\max}$ as in Fig. 1.

When the estimated contour $\varepsilon_r{}'$ increases, feedrate u must be reduced. Otherwise, the feedrate should be increased to speed up contouring. Triangular-shape membership functions are used in [6] to classify contour errors. From the results of circular testing, contour errors are reduced with proposed cross-coupled fuzzy feedrate control. Particularly, the quadrant errors are suppressed as expected. Even given an increasing feedrate, the controller performance is still kept better than that without the proposed FLC method. Lin et al. [7] used fuzzy learning control to enhance transient performance and robustness of a derived adaptive control system concerning to transmission stiffness. Both tracking error ε_r, time derivative of tracking error $\dot{\varepsilon}_r$ and transmission deflection sp are selected as the input variables of proposed FLC. The three variables are normalized by the corresponding scaling factors within a range $[-6,6]$, and then discretized for binary coding. To provide an active damping and prevent occurrence of stick-slip as well as consider desired motion direction, the fuzzy control rule base is composed of two parts: one for the positive direction and the other for the negative direction.

In normal case, contour error $\varepsilon_r(i) = P_r(i) - P_a(i)$ and/or its first-order derivative $\Delta\varepsilon_r(i+1) = \varepsilon_r(i+1) - \varepsilon_r(i)$ are used for input variables in PID control rules. Higher order derivatives of contour error are not commonly used, since they would increase the computational burden in fuzzy inference portion and worse might amplify high frequency noise remarkably. To improve steady-state behavior of an FLC, Jee and Koren [8] add an integral control action to the proposed PD–FLC, as shown in Fig. 2. The con-troller inputs are axial position errors e at current time step and its change Δe between previous and current step. The performance of a fuzzy logic controller, however, is dependent on the pre-defined fuzzy sets or mem-bership functions and control rules. To utilize some self-organizing algo-rithms to create fuzzy control rules is an ongoing research object. The SelfOrganizing Controller (SOC) has a hierarchical structure consisting of two rule bases [9]. The first one is a general rule base of the FLC, and the second one is constructed by "meta-rules", which exhibits some human-like learning algorithm to create and modify the first rule base. Jee et al. [9] gave an example of the Self-Organizing FLC (SOFLC) as illustrated in Fig. 3, where the shapes of input membership functions (A_i and B_j) are changed and centroids of output membership functions (C_{ij}) are shifted

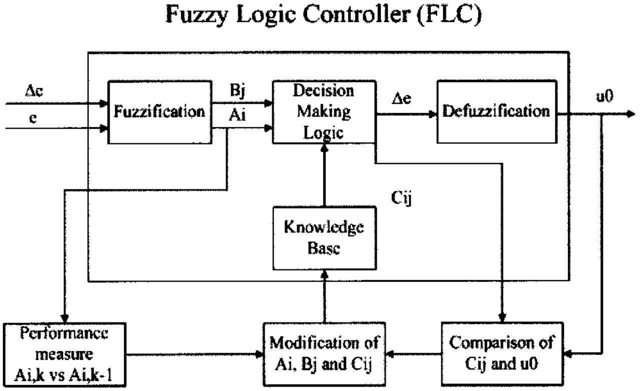


Fig. 3. Self-organizing FLC [9]

automatically, according to a performance evaluation of contribution of each activated control rule in last action to the overall control action. By contrast to the regular PID controller, the proposed SOFLC can reduce the RMS contour errors by the ratios of 1.4:1 and 2.6:1 for lower and higher feedrate, respectively.

Design of a FLC, including the choice of membership functions and control rules, are mainly based on the trial and error methods that heavily depend on the intuitive experience acquired from practicing operators. Fuzzy membership functions and fuzzy rules cannot be guaranteed to be optimal in any sense. Additionally, in fuzzy control systems, there is no analytical way to guarantee the stability of a general configuration. Therefore, it is necessary to perform a stability analysis for fuzzy logic control systems to determine the stable domain of control parameters. Moreover, definitions of controllability and observability in fuzzy control are not widely accepted yet. The future direction of fuzzy logic servo-control could be combining the conventional control strategy with the FLC, since for the later one there is a series of applicable theories. Through this integration, the two control strategies can compensate for each other's drawbacks.

3 Neural Networks-Based Error Modeling

With respect to the computing speed of microprocessors nowadays, neural networks can quickly perform competing hypotheses simultaneously using massively parallel nets composed of numerous computational elements connected by links with variable weights [10]. Neural networks have the ability of adaptation and continue learning from training data, recalling memorized information, and generalizing to the unseen patterns. Their characteristics ensure themselves powerful nonlinear function approximation capabilities that have drawn great attention in nonlinear modeling against geometric errors and/or thermal errors of a CNC machine tool.

Chen et al. [11] used a kind of Artificial Neural Network (ANN) with multiple-layer feedforward architecture to predict thermal errors on line. The input layer is an input buffer of temperature measurements $\Delta T_1, \Delta T_2, \cdots, \Delta T_n$ through n thermocouples at different temperature sensing positions. The output layer is an output buffer of predicted thermal error $\delta 1, \delta 2, \cdots, \delta m$ at m thermal deformation error sensing positions. The nonlinear and interaction characteristics of thermal errors are represented in a form of connected weights w_{ij} between input and hidden layers, and w_{jk} between hidden layer and output layer. These weights were obtained through a supervised Back-Propagation (BP) training algorithm. When temperature measurement inputs are applied, thermal error outputs of ANN are calculated and compared with the corresponding target error vector. The error between them is then fed back to adjust the weights. The training is iterated until the error reaches an acceptably low level. Based on this thermal error model for compensation, dimension errors of a cut part are reduced from 92.4 micrometers to 18.9 micrometers and the depth difference of milled surfaces are reduced from 196 to 8 micrometers.

The main shortcoming of the BP neural network is its slow convergence speed. Yang et al. [12] pointed out that the disadvantages of feedforward neural network used in [11] are the sensitivity to sensor locations and requirement of a great deal of experimental data for modeling (a long calibration time). In their research work, a CMAC is used to model thermal errors due to its fast learning characteristics for purpose of real-time compensate. It is discovered that the CMAC-based approach can accurately and fast predict new observations of thermal error under varying temperatures. Even if there are some difficulties in mounting thermal sensors on an optimal position, sufficient accuracy can still be achieved. Provided one sensor fails, the method can quickly detect the sensor failure, and recover the predicted error model within an acceptable tolerance.

To carry a correction action in machine tools, Mou [13] proposed a new integrated approach to simplifying and completing the problem of error modeling and correction by simultaneously using neural networks and inverse kinematics, as illustrated in Fig. 4. Simple BP neural networks are deployed to derive the thermal error model. Input vector T consists of 40 input units that represent the temperature profile of machine structure. Output vector e represents the positioning errors at particular measuring points. Neural network functions to track the time-varying positioning

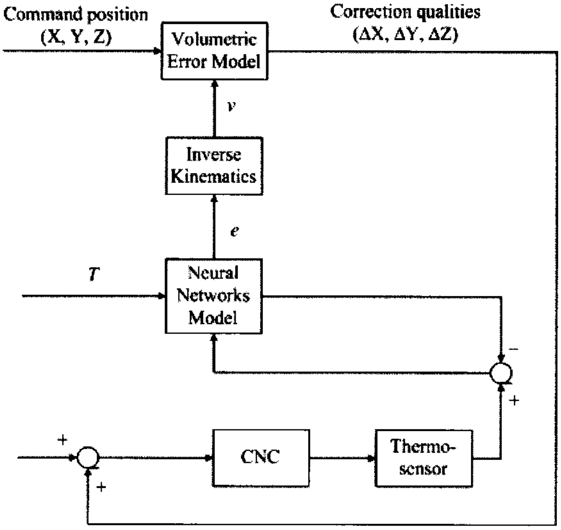


Fig. 4. Integrated approach for machine tool compensation.

errors e by monitoring changes in temperature profiles T. The inverse kinematics is used to find coefficients of volumetric error model v from e. For commanded tool position (X,Y,Z), corresponding correction qualities $(\Delta X,\Delta Y,\Delta Z)$ can be obtained after passing the volumetric error model. Finally, the correction qualities are added on the commanded positions of CNC drives as a new command. Performance analysis results show that the positioning errors are reduced from about 150 micrometers before correction to less than 20 micrometers after correction.

Geometric errors and thermal errors are regarded as the major sources to affect positioning error of machine tools. In the work of Raksiri and Parnichkun [14], a total of 21 geometric error components for a 3-axis machine tool are measured directly and a BP neural network is used to approximate geometric error model that is conventionally synthesized based homogeneous transformation matrix methodology [15]. In [14], the cutting force induced error is also analyzed by neural network and combined with geometric errors. The input vector is composed of the commanded tool position components and feedrate components along three axes plus cutting depth. Output vector are three correction qualities for compensation. Experimental cutting results show that positioning errors are decreased from hundreds micrometers to only tens micrometers.

From the discussions above, the motivation of employing neural networks for thermal error and/or geometric error modeling is because of their self-adaptation and nonlinear approximation abilities, which can set up an implicit relationship between indication of error components and commanded inputs including temperatures and/or moving position. Nevertheless, various modified neural networks of high level with self-organization converge so slowly that they are not suitable for real-time modeling. The inherent drawbacks of neural network learning, such as problem dependent selection and over- and under-training may also affect the modeling performance. Additionally, neural networks are "black-box" operating models. There is little possibility to understand the internal properties of error models, which are actually necessary to find the sticking point of error existence and improve the hardware structure of machine tool. Therefore, theoretical developments of neural networks, i.e., fast convergence speed and study of mapping relationship between neural network structure and practical models are expected. A proper selection of specific neural networks models is crucial in the system performance.

4 Genetic Algorithms for Parameter Optimization

Genetic Algorithms (GA) is a kind of global and parallel search algorithm based on the principles of natural selection and natural genetics. Usually, a simple GA consists of three operations: selection, genetic operation, and replacement. Initially, a population is generated randomly, which comprises a group of chromosomes. The fitness values of all chromosomes are first evaluated by calculating corresponding objective functions in a decoded form (phenotype). A particular group of chromosomes (parents) is selected from the population to generate offspring by defined genetic operation including crossover and mutation. The fitness of the offspring is evaluated in a similar fashion to their parents. Chromosomes in current population are then replaced by their off-springs, based on a certain replacement strategy [16]. Different from the conventional optimization schemes that usually require the derivative operation on objective functions, GA is a derivative-free and stochastic optimization method with less prior information of the problems to be solved, and has a better probability of locating the global optimum. Hence, it is attractive to employ a GA to automatically tune and optimize the parameters of proposed controllers, and geometric and/or thermal error models, which are difficult to handle by using the conventional optimization methodologies.

In [17], Tarng et al. discussed the employment of GA to optimize a fuzzy logic controller in turning operation. The design problem of a fuzzy logic controller includes determination of input and output scaling factors, design of membership functions, and generation of fuzzy rule base. First, each scaling of two inputs and one output of FLC is encoded by five bits.

The boundary vertices of seven initial membership functions are termed as the design variables as shown in Fig. 5: NM, NS, ZE, PS and PM have four vertices representing four design variables; but NB and PB have only two vertices with two design variables. Every termed design variable is encoded by four bits. The encoding of the outcome of each fuzzy rule requires three binary bits, which correspond to seven fuzzy sets and an empty fuzzy set. In their research, a population of 1000 strings is selected, and the performance index is defined based on the force error and force error change. Figure 4(b) shows the resulting membership functions after the GA optimization. From this study, the design cycle time for FLC in turning operations can be reduced from hours to minutes. The approach achieves an optimal or near-optimal control performance under variable feedrate and cutting force. It can also be extended to milling and drilling operations.

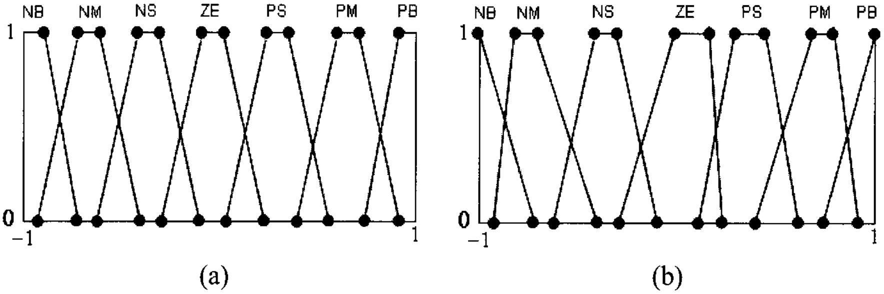


Fig. 5. Tuning of membership functions of FLC.

In [7], Lin et al. proposed a GA for constructing an appropriate Fuzzy-Enhancing Control (FEC) combined with adaptive strategy, as illustrated in Fig. 6, to improve the transient performance and robustness with respect to friction and transmission flexibility. The number and definition of fuzzy sets and scaling factors for corresponding variables, i.e., tracking error, tracking error change, transmission deflection and outputs, are learned by the GA. Membership functions are coded as in Fig. 6. In sense, this coding approach is similar to that in [17]. The only difference is that the membership functions are assumed as triangular shape. The membership function centers are chosen as 1-bits, and the base width of each membership function is defined to be the length between the centers of two neighboring fuzzy subsets. Computer simulations using the friction model verify the excellent steady-state performance and quick transition. Compared with the pure adaptive control, in the proposed fuzzy-tuning adaptive control system, the transient oscillation of equivalent spring is greatly suppressed, and convergence rates of parameters are faster.

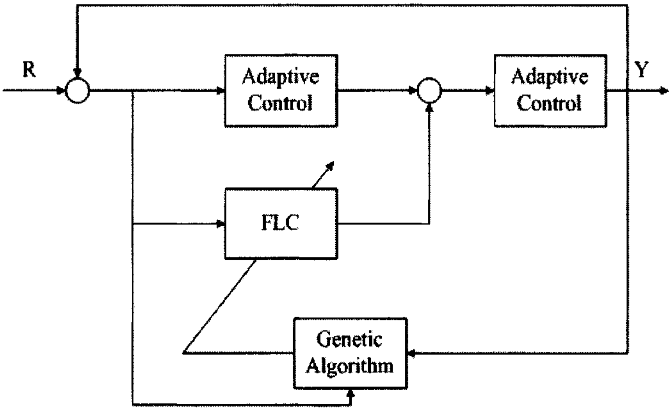


Fig. 6. Fuzzy-enhanced adaptive control with GA.

Since the GA is an auxiliary optimization method, it is usually applied together with other soft computing methods. Due to the random search operation and diverse chromosomes in the GA, its performance is difficult to predict. Thus, more theoretical works are needed to further investigate the performance of GA, and faster GA with parallel implementation capabilities have to be developed as well.

5 Conclusions

In this paper, an overview is given on the recent significant applications of soft computing methods for enhancing positioning accuracy of machine tools. Three main topics are discussed: 1. fuzzy logic controllers for compensating friction between moving components of machine tool; 2. neural networks for model approximation of geometric errors and/or thermal errors; 3. genetic algorithms for tuning and optimizing parameters of machine tool controllers and error models. It is observed that soft computing methods can provide improved solutions to complex and nonlinear practical machining processes. However, they are not supposed to compete with classical strategies, because soft computing methods have some properties, such as the controllability and observability of FLC, convergence speed of neural networks, and operation principle clarification of genetic algorithm, cannot be guaranteed. The integration of soft computing and conventionally classical methods is a promising means, and theoretical breakthroughs in soft computing methods are also critical.

References

1. Ramesh R, Mannan M, Poo A (2000) Error compensation in machine tools – a review. International Journal of Machine & Manufacture 40: 1235–1256
2. Bosch J (1995) Coordinate measuring machine and system. Marcel Dekker.
3. Ni J, Wu S (1993) An on-line measurement technique for machine volumetric error compensation. Journal of Engineering for Industry 115: 85–92
4. Koren Y (1983) Computer control of manufacturing systems. McGraw-Hill.
5. Koren Y (1980) Cross-coupled biaxial computer control for manufacturing systems. Journal of Dynamic Systems, Measurement and Control 102: 265–272.
6. Tarng Y, Chuang H, Hsu W (1999) Intelligent cross-coupled fuzzy feedrate controller design for CNC machine tools based on genetic algorithms. International Journal of Machine Tool & Manufacture 39: 1673–1692.
7. Lin L, Lin Y (1998) Fuzzy-enhanced adaptive control for flexible drive system with friction using genetic algorithms. Journal of Intelligent and Robotic Systems 23: 379–405.
8. Jee S (1999) Fuzzy logic controls for CNC machine tools. PhD Thesis, University of Michigan.
9. Jee S, Koren Y (1995) A self-organization fuzzy logic control for friction compensation in feed drives. Proceedings of the American Control Conference: 205–209.
10. Lippmann R (1987) An introduction to computing with neural networks. IEEE ASSP Magazine, April 1987, pp. 4–22.
11. Chen J, Yuan J, Ni J (1995) Computer-aided accuracy enhancement for multi-axis CNC machine tool. International Journal of Machine Tools & Manufacture 35: 593–605.
12. Yang S, Yuan J, Ni J (1996) The improvement of thermal error modeling and compensation on machine tools by CMAC neural network. International Journal of Machine Tools and Manufacture 36: 527–537.
13. Mou J (1997) A method of using neural networks and inverse kinematics for machine tool error estimation and correction. ASME Journal of Manufacturing Science and Engineering 119: 247–254.
14. Raksiri C, Parnichkun M (2004) Geometric and force errors compensation in a 3-axis CNC milling machine. nternational Journal of Machine Tool & Manufacture 44: 1283–1291.
15. Okafor A, Ertekin Y (2000) Derivation of machine tool error models and error compensation procedure for three axes vertical machining center using rigid body kinematics. International Journal of Machine Tool & Manufacture 40: 1199–1213.
16. Tang K, Man K, Kwong S (1996) Genetic algorithms and their applications. IEEE Signal Processing Magazine: 22–37.
17. Tarng Y, Yeh Z, Nian C (1996) Genetic synthesis of fuzzy logic controllers in turning. Fuzzy Sets and Systems 83: 301–310.
18. Gao X, Ovaska S (2001) Soft computing methods in motor fault diagnosis", Applied Soft Computing 1: 73–81.

Mobile Robot Navigation: Potential Field Approach Vs. Genetic-Fuzzy System

Nirmal Baran Hui[1] and Dilip Kumar Pratihar[2]

[1] Research Scholar
nirmal@mech.iitkgp.ernet.in
[2] Associate Professor
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur-721 302
India
dkpra@mech.iitkgp.ernet.in

Summary. The present paper deals with the issues related to collision-free, time-optimal navigation of an autonomous car-like robot, in the presence of some moving obstacles. Two different approaches are developed for this purpose. In the first approach, the motion planner is developed by using a conventional potential field method and a fuzzy logic-based navigator is proposed in Approach 2. In the present work, an attempt is made to develop a good knowledge base (KB) of an FLC automatically, by using a genetic algorithm (GA). During training, an optimal rule base of the FLC is determined by considering the importance of each rule. The effectiveness and computational complexity of both the approaches are compared through computer simulations.

Key words: Navigation, Car-Like Robot, Potential Field Method, Fuzzy Logic Control, Genetic Algorithm.

1 Introduction

To meet the increasing demand of robots, design and development of an autonomous robot has become a thrust area in robotic research. An autonomous robot should be able to plan its collision-free path on-line, in varying situations. The problems of collision-free navigation in a known terrain have been extensively studied by several investigators. Both graph-based as well as analytical techniques have been developed. However, all these techniques may not perform effectively in dynamic environments, where the motion of the robot is to be planned in a partially-known environment. Thus, building a complete mathematical model is not possible. Moreover, a car-like robot is subjected to both nonholonomic as well as dynamic constraints [1]. Therefore, it can only move forward or backward in a direction tangent to its trajectory.

Latombe [2] provides an extensive survey on different conventional motion planning schemes of car-like robots. Potential field method [3] has come out to be the most popular of all conventional approaches. But, its performance depends on the chosen potential functions, a proper selection of which is a tough task. Therefore, it may provide some feasible solutions to the present problem, which may not be optimal in any sense. Moreover, most of the traditional methods are unable to deal with uncertain and imprecise sensory informations. Thus, there is a need for an approach that can handle uncertainties at all levels and deal with various situations, those are not known a priori.

Fuzzy set theory had been introduced by Zadeh [4] in the year 1965, to deal with vague, uncertain and imprecise data related to the real-world problems. Recently, some researchers [5, 6] have started thinking, whether they should switch over to a fuzzy logic-based motion planner. However, the performance of an FLC depends on its KB, design of which is not an easy task. Several methods, such as least squares [8], gradient descent [9], back-propagation algorithm of neural network [10] and others, had been proposed by various investigators, to develop an optimal KB of an FLC. But, all such methods failed to give optimal solutions, due to the fact that they might have local minima problem. Moreover, the problem of knowledge acquisition of an FLC is a tough task, due to the fact that a human expert often finds it difficult to express his or her control actions. Several trials had been made by quite a few researchers, for the development of a suitable KB of an FLC, by using a GA [11]. Optimization of both the data base as well as rule base of an FLC had been carried out simultaneously using a GA by a few investigators [12, 6]. In this regard, there are three basic approaches, namely Michigan, Pittsburgh and iterative rule learning. Interested readers may refer to [7], for a detail study of the same. However, in some of these methods, a considerable amount of time was spent on manual design of rule base and a GA was used to further tune it. To overcome this difficulty, some investigators [13, 14] tried to design the FLC automatically by giving the complete task of designing a suitable KB to the GA. It is important to mention that the GA-learned rule base of an FLC may contain some redundant rules, but no effort was made to identify and remove them in the above earlier work. It happens due to the iterative nature of the GA. Realizing all such problems, an attempt is made in the present study, to design the FLC automatically, in such a manner that the redundant rules (if any) will be removed from the rule base. The effectiveness of both these approaches are tested through computer simulations, for solving the navigation problems of a car-like robot moving in a dynamic environment. It is to be noted that the present work differs from the earlier work [6] in the sense that a car-like robot has been considered including its kinematic and dynamic constraints, in place of a point robot. Thus, it is a more realistic problem compared to the earlier.

The rest of the paper is structured as follows: Dynamic motion planning problem of a car-like robot is stated and the motion planning scheme is explained in Section 2. The developed motion planning algorithms are discussed in Section 3. Results of computer simulations are presented and discussed in Section 4 and some concluding remarks are made in Section 5.

2 Dynamic Motion Planning of a Car-Like Robot

Motion planning problem of a car-like robot navigating in the presence of some moving obstacles, is considered in this paper.

2.1 Statement of the Problem

During navigation, a car-like robot has to find its collision-free, time-optimal paths, after starting from a fixed point and to reach a target point situated in an unknown environment populated with some moving obstacles. Moreover, motion planning problem of a car-like robot is a complicated one, due to the fact that it should satisfy both nonholonomic as well as dynamic constraints during its navigation. In the present work, both the nonholonomic as well as dynamic constraints (such as motor torque constraint, curvature constraint, sliding constraint) of the car-like robot [15] have been considered. To meet these requirements, a suitable motion planning technique has to be developed, which can plan and control the motion of a car-like robot, on-line, in an optimal sense. For simplicity, all the moving obstacles are represented by their bounding circles and the robot is assumed to move due to pure rolling action only. The developed motion planning scheme is discussed below.

2.2 Motion Planning Scheme

The complete path of the robot is considered as a series of small segments, either curved or straight or a combination of them. Each segment is assumed to be traversed during a fixed time ΔT and the robot's path is planned based on the position of the most critical obstacle, which is identified by considering the relative velocity of the robot with respect to the obstacle and the direction of movement of the obstacle. It is important to mention that only one obstacle is considered to be the most critical at a time and no two obstacles are allowed to overlap each other. If the robot finds any critical obstacle ahead of it, in the predicted time step, the motion planner is activated. Otherwise, it moves with the maximum possible velocity by following a straight path, directed towards the goal. The task of the motion planner is to determine the acceleration and deviation required by the robot to avoid collision with the most critical obstacle. This process continues till the robot reaches the target and the total traveling time is calculated by adding all the intermediate time steps.

Our aim is to design the motion planner in such a manner that the robot reaches the goal with a minimum possible traveling time. Thus, the present problem can be treated as a constrained traveling time minimization problem. It is important to note that a robot will reach the target with the minimum possible time, only when it moves with the maximum possible velocity and takes less deviation to avoid collision with the obstacles. Thus, the problem under this study, is solved by minimizing the error due to both acceleration as well as deviation required by the robot simultaneously, to avoid collisions with the most critical obstacle after satisfying the constraints.

3 Developed Motion Planning Algorithms

Two different approaches of robot motion planning are developed, in the present work, which are discussed below.

3.1 Approach 1: Potential Field Method

Potential field method, introduced by Khatib [16], is widely used for real time collision-free path planning of both manipulators as well as mobile robots. In this approach, the robot is modeled as a particle moving under the influence of an artificial potential field, which is determined by the set of obstacles and the target destination. The target is assumed to have attractive potential and the obstacles generate the repulsive potential. The movement of the robot is then achieved by determining the resultant force due to these two potential fields. However, the performance of the potential field method depends on the chosen artificial potential functions. Several potential functions, such as parabolic-well, conic-well, hyperbolic function, rotational field function, quadratic, exponential function, are tried by various investigators [3, 1], out of which, parabolic and hyperbolic functions are widely used for solving the similar problem [17], due to their nonlinear approximation capability about the system. The attractive $U_{att}(X)$ and repulsive $U_{rep}(X)$ potential fields, used in this study, can be expressed as follows.

$$U_{att}(X) = \frac{1}{2}\xi_{att}\, d^2_{goal}(X), \tag{1}$$

where ξ_{att} is a positive scaling factor of attractive potential and $d_{goal}(X)$ denotes the Euclidean distance of the robot from its goal.

$$U_{rep}(X) = \frac{1}{2}\xi_{rep}\left[\frac{1}{d_{obs}(X)} - \frac{1}{d_{obs}(0)}\right]^2, \tag{2}$$

where ξ_{rep} is a positive scaling factor of repulsive potential and $d_{obs}(X)$ indicates the Euclidean distance of the robot from the obstacle and $d_{obs}(0)$ represents the distance of influence of the obstacle and it is made equal to the center distance between the robot's bounding circle and that of the obstacle.

The attractive and repulsive potential forces are then calculated by differentiating the respective potential fields with respect to the goal distance and obstacle distance, respectively. It is important to mention that in the present study, acceleration of the robot is taken to be proportional to the magnitude of the resultant force and deviation is considered as the angle made between the direction of the resultant potential force and the new reference line joining the CG of the robot at the present time step and the goal position.

3.2 Approach 2: Genetic-Fuzzy System

An FLC may also provide some feasible solutions to the said problem. Two condition variables, such as (i) *distance* of the robot from the most critical obstacle and (ii) *angle* between the line joining the robot and the most critical obstacle and the reference line (joining the robot and its goal) are fed as inputs to the controller. The outputs of the controller are taken to be *deviation* and *acceleration* required by the robot to avoid collision with the most critical obstacle. In the present study, the range of *distance* is divided into four linguistic terms: very near (VN), near (NR), far (FR), very far (VF). Five linguistic terms have been considered for both the *angle* as well as *deviation*: left (LT), ahead left (AL), ahead (AH), ahead right (AR) and right (RT) and *acceleration* is considered to have four terms: very low (VL), low (L), high (H), very high (VH). Therefore, there will be a maximum of twenty input conditions, and for each input condition, there is a maximum of twenty output combinations. Thus, there is a maximum of 400 (i.e., 20 × 20) rules present in the rule base and a particular rule will look like the following.

IF *distance* is VF AND *angle* is LT, THEN *deviation* is AH and *acceleration* is VH.

For ease of implementations, membership function distributions of both the input as well as output variables are assumed to be symmetric triangles. Thus, the data base of the FLC may be represented by providing the four continuous variables representing the half base-widths of the triangular membership function distributions. The performance of an FLC depends on its both data base as well as rule base, which are to be optimized simultaneously. In the present work, an attempt is made to develop a good KB of an FLC automatically by using a binary-coded GA. A GA-string consisting of 440-bits is considered to indicate the KB of the FLC as shown below.

$$\underbrace{\underbrace{\underline{10\cdots1}}\ \underbrace{\underline{01\cdots1}}\ \underbrace{\underline{10\cdots0}}\ \underbrace{\underline{01\cdots0}}}_{\textit{Data base}}\quad \underbrace{\underline{10\cdots01}}_{\textit{Input combinations}}\quad \underbrace{\underline{10101\ldots0101\ldots11001}}_{\textit{Consequent of the rules}}$$

The first 40-bits in this string represent the half base-widths of the four triangles (10 bits for each variable) and the next 20-bits are used to indicate the presence or absence of the input combinations in the rule base (1 for presence and 0 for absence). Out of the remaining 380-bits of the string, every 19-bits will carry the information regarding the combination of the consequents, for a particular input condition. We count the number of 1s present in each 19-bits long sub-string. If it comes out to be zero, it will represent the first output combination, i.e., deviation is LT and acceleration is VL, and so on. The fitness of a GA-string is then calculated, as follows.

$$Fitness = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{S}\sum_{s=1}^{S}\sum_{v=1}^{2}(T_{nsv} - O_{nsv}), \tag{3}$$

where S denotes the total number of time steps in a planned path and the total number of training scenarios is indicated by N. O_{nsv} and T_{nsv} are representing the values of actual output and target output, respectively, of an output variable (say,v). The target output for deviation is considered to be equal to zero and that for acceleration is taken as the maximum permissible acceleration of the robot. A fixed penalty equals to 200 is added to the said fitness value, when the robot collides with the most critical obstacle during its movement from the present position to the predicted location. Moreover, another fixed penalty equals to 2000 is given to the string, if the FLC represented by it, is unable to provide any solution particularly in case of non-firing situation or the generated motion of the robot fails to satisfy the dynamic and/or kinematic constraints.

During optimization, an optimal rule base of the FLC is determined by considering the importance of each rule, which is calculated as $I_{ij} = p_{ij}\ \overline{\overline{C}}_j$, where p_{ij} denotes the probability of occurrence of j^{th} output combination corresponding to i^{th} input condition of the rule, where $i, j = 1, 2, \cdots, 20$ and $\overline{\overline{C}}_j = \frac{1}{2}\ (\overline{C}_q + \overline{C}_r)$, where $\overline{C}_q$ and $\overline{C}_r$ are the average worth of q^{th} linguistic term of the first output (i.e., deviation) and r^{th} term of acceleration output, respectively. It is important to note that the worth, corresponding to a linguistic term of an output, is determined by following the Gaussian distribution pattern, maximum being occurred for deviation output AH and acceleration output VH.

4 Results and Discussion

In the present study, a car-like robot is allowed to navigate among sixteen obstacles moving in a 2-D environment of size equal to 19.95 × 19.95m^2. To provide training to the FLC, two hundred training scenarios are generated at random. A particular training scenario is different from the other, in terms of the initial position of the moving obstacles, their size, speed and direction of movement. During optimization, half-base width of four triangular membership function distributions are varied in the ranges of (0.8,1.0), (20,40), (20,40) and (0.005,0.015), respectively. The ranges of variation for the different variables are selected through a careful study. Moreover, it is to be noted that the time interval (ΔT) is taken to be equal to sixteen seconds and the maximum and minimum accelerations of the robot are set equal to $0.05m/s^2$

and $0.005m/s^2$, respectively. During training, the best results are obtained with the following GA-parameters: crossover probability $p_c = 0.84$, mutation probability $p_m = 0.00166$, population size $Y = 140$, maximum number of generation $Maxgen = 98$.

Twelve good rules have been identified by the GA, out of which, two rules are found to be redundant (refer to Fig. 1). A rule is said to be redundant and may be eliminated, if the *importance factor* of that rule comes out to be smaller than a pre-specified value and the removal of which does not lead to any non-firing situation. It is observed that the number of non-firing incidences increases with the reduction of number of rules present in the rule base, as shown in Fig. 1.

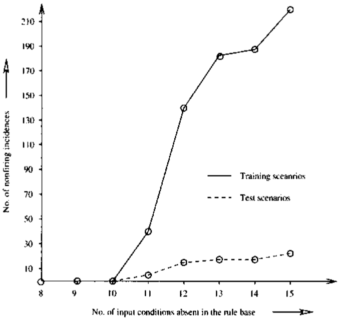


Fig. 1. Number of rules absent in rule base vs. number of non-firing incidences.

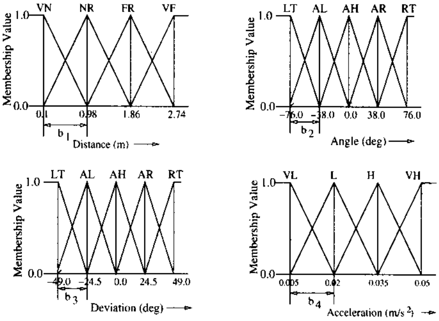


Fig. 2. Optimized membership function distributions of the FLC.

The optimized rule base along with the importance factors of the rules are shown in Table 1. In case of first and third rules of the optimized rule base

Table 1. Optimized rule base of the FLC.

Distance	Angle	Deviation	Acceleration	Worth $(\overline{\overline{C}}_j)$	Probability of occurrences	Importance factor
VN	AL	AL	VH	0.792484	0.038017	0.030128
VN	AH	AH	VH	1.000000	0.067505	0.067505
VN	AR	AR	VH	0.792731	0.046960	0.037227
NR	AH	AH	VH	1.000000	0.153483	0.153483
FR	LT	AH	VL	0.515048	0.052486	0.027033
FR	RT	AH	VL	0.515048	0.045074	0.023215
VF	LT	AH	VH	1.000000	0.029633	0.029633
VF	AL	AH	H	0.918319	0.105903	0.097253
VF	AH	AH	H	0.918319	0.157644	0.144767
VF	RT	AH	L	0.630077	0.040236	0.025351

(refer to Table 1), both the angle as well as deviation are coming out to be the same with respect to the linguistic terms, namely AL and AR, respectively. It is important to note that although the linguistic terms are the same, their numerical values are coming out to be the different during optimization. It happens because the membership function distributions of deviation will try to squeeze, to ensure a minimum traveling time. On the other hand, the optimizer may choose a wider membership function distributions for the angle input, just to increase the view angle of the robot. For example, the crisp values corresponding to AL are coming to be equal to -38.0° and -24.5° for angle and deviation, respectively (refer to Fig. 2). Thus an AL, angle may not be exactly equal to a deviation expressed by the same linguistic term – AL.

A close watch on the 2nd and 4th rules of the optimized rule base as shown in Table 1, reveals the fact that the linguistic terms for both angle as well as deviation are coming to be the same as AH. It may be due to the reason that during training, if there is a chance of the robot, in a particular step, that it collides with the most critical obstacle, a penalty equal to 200 is added to the fitness of the GA-string, to come out of this situation. It occurs, when the direction of movement of the most critical obstacle, in that particular step, intersects the direction of movement of the robot as planned by the motion planner. However, this penalty helps the robot to cross the predicted position of the obstacle as early as possible. Moreover, a VH acceleration as planned by the motion planner helps the robot to ensure this.

For FR and VF distances (irrespective of the angle input), the robot tries to navigate towards the AH direction, to achieve a time-optimal path. The optimized half base-widths of the membership function distributions are found to be as follows: $b_1 = 0.877$ (for *distance*), $b_2 = 37.947$ (for *angle*), $b_3 = 24.496$ (for *deviation*) and $b_4 = 0.0149$ (for *acceleration*) (refer to Fig. 2). The performances of both the approaches are tested for twenty test scenarios, created at random. Results of computer simulations are shown in Fig. 3 and it is observed that Approach 2 has outperformed Approach 1 in all the test scenarios. It could be due to the fact that in potential field method, the robot often may get stuck in the local minima. Moreover, it does not have any in-built optimization module. The movement of the obstacles as well as the robot in a particular test scenario (say, 4th), is shown in Fig. 4, in detail. It has been observed that Approach 1 is unable to provide good solutions, when any obstacle comes near to the line joining the robot and the goal or it moves perpendicular to the direction of movement of the robot. However, the GA-learned FLC has understood these situations well and behaved optimally. The CPU time of both the approaches are found to be small, making them suitable for solving the navigation problems of a car-like robot in a partially-known environment, on-line.

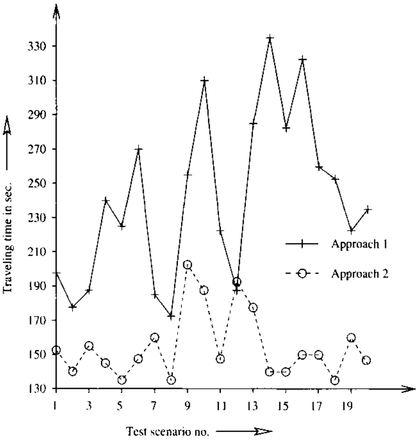


Fig. 3. Comparison of two approaches in terms of traveling time taken by the robot.

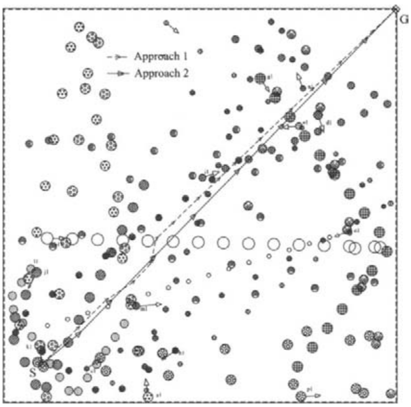


Fig. 4. Collision-free paths obtained by the robot using two different approaches.

5 Concluding Remarks

Potential field method is a widely used approach for robot motion planning, although it has a number of drawbacks. However, fuzzy logic-based motion planners are drawing interest, nowadays, due to its ease of implementation and ability to deal with imprecise and uncertain sensory readings. Several methods have been developed to find a suitable KB of an FLC, but each of these approaches has its inherent limitations. A method for automatic design of FLC is undertaken in the present study, in which a binary-coded GA is used to develop a good KB of an FLC. During training, an optimal rule base of the FLC is determined by the GA and it is further modified by considering the importance of each rule.

The effectiveness of these two approaches are studied, to solve the navigation problems of a car-like robot, in the presence of sixteen moving obstacles, through computer simulations. Approach 2 has proved its supremacy over Approach 1, for twenty randomly-generated test scenarios. It may be due to the fact that there is no in-built optimization module in the potential field method and it may have local minima problem also. It is interesting to note that importance of each rule present in the GA-designed rule base is determined to identify the redundant rules (which may be eliminated), if it does not lead to any non-firing incidence. Therefore, the present genetic-fuzzy approach may be utilized to develop an optimal KB of an FLC, which will contain only the significant rules. Moreover, as optimization of the FLC is carried out off-line, it might be suitable for solving the motion planning problems of a car-like robot, on-line. The performances of the present approaches are tested on computer simulations only. However, it will be more interesting to see their performances on a real robot.

Acknowledgment

This work is supported by the Department of Science and Technology, Govt. of India (Sanction No. SR/S3/RM/28/2003 dt. 12.12.2003). The authors gratefully acknowledge the cooperation of Dr. A. Roy Chowdhury of Department of Mechanical Engineering, IIT–Kharagpur, India.

References

1. Feng D, Krog H, Bruce H (1990), Dynamic steering control of conventionally steered mobile robots, Proc. of IEEE Conference on Robotics and Automation, 390–395.
2. Latombe J C (1991) Robot motion planning. Kluwer Academic Publishers
3. Bemporad A, Luca A D, Orilo G (1996), Local incremental planning for a car-like robot navigating among obstacles, Proc. of IEEE Conference on Robotics and Automation, Minneapolis, Minnesoia, 1205–1211.
4. Zadeh L A (1965), Fuzzy sets, Information and Control 8:338–353.
5. Fraichard T, Garnier P (2001), Fuzzy control to drive car-like vehicles, Robotics and Autonomous Systems, 34:1–22.
6. Pratihar D K, Deb K, Ghosh A (1999), A genetic-fuzzy approach for mobile robot navigation among moving obstacles, International Journal Approximate Reasoning, 20:145–172.
7. Cordon O, Herrera F, Hoffman F, Magdalena L (2001) Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases, World scientific.
8. Pham T D, Valliappan S (1994), A least squares model for fuzzy rules of inference, Fuzzy Sets and Systems, 64:207–212.
9. Nomura H, Hayashi I, Wakami W (1992), A learning method of fuzzy inference rules by descent method, Proc. of IEEE International Conference on Fuzzy Systems, San Diego, CA.
10. Jang J R (1992), Self-learning fuzzy controllers based on temporal back propagation, IEEE Transactions on Neural Networks, 3:714–721
11. Goldberg D E (1989) Genetic algorithms in search, optimization, machine learning. Addison-Wesley, Reading, Mass, USA
12. Pham D T, Karaboga D (1991), Optimum design of fuzzy logic controllers using genetic algorithms, Journal of Systems and Engineering, 1:114–118.
13. Nandi A K, Pratihar D K (2004), Automatic design of fuzzy logic controller using a genetic algorithm-to predict power requirement and surface finish in grinding, Journal of Material Processing Technology, 148:288–300.
14. Magdalena L, Monasterio F (1997), A fuzzy logic controller with learning through the evolution of its knowledge base, International Journal of Approximate Reasoning, 16(3–4):335–358.
15. Shiller Z, Gwo Y R (1991), Dynamic motion planning of autonomous vehicles, IEEE Transactions on Robotics and Automation, 7(2):241–249.
16. Khatib O (1986), Real-time obstacle avoidance for manipulators and mobile robots, International Journal of Robotics Research, 5(1):90–98.
17. Biswas D K (2003) Path planning of multiple robot working in the same workspace–potential field approach. M.Tech thesis, REC Durgapur, India

Intelligent Tuning and Application of a PID Controller Using Universal Model

Rodrigo R. Sumar[1], Antonio A. R. Coelho[1], and Leandro dos Santos Coelho[2]

[1] Department of Automation and Systems - Federal University of Santa Catarina
Box 476, Zip Code 88094-900, Florianópolis, SC, Brazil
{sumar,aarc}@das.ufsc.br

[2] Automation and Systems Laboratory - Pontifical Catholic University of Paraná
Imaculada Conceição, 1155, Zip Code 80215-901, Curitiba, PR, Brazil.
leandro.coelho@pucpr.br

1 Introduction

In spite of many efforts the PID (Proportional-Integral-Derivative) controller continues to be the main component in industrial control systems, included in the following forms: embedded and programmable logic controllers, and distributed control systems. From the viewpoint of simplicity and effectiveness, the PID controller represents a good solution for controlling many practical applications. However, the control literature still showing different tuning structures for the PID control to overcome complex dynamics and operational limitations [1]. Some difficult problems in the industry are presented in order to be answered or understood, not from the enthusiastic form neither from the absence of knowledge of plant operators or engineers, as the tuning aspects of a PID controller [2].

Since nineties, many methodologies for setting the gains of PID controllers have been proposed. In this context there are classical (Ziegler/Nichols, Cohen/Coon, Abas, pole placement and optimization) and advanced techniques (minimum variance, gain scheduling and predictive) [1; 3; 4]. Some disadvantages of these control techniques for tuning PID controllers are: i) excessive number of rules to set the gains, ii) inadequate dynamics of closed-loop responses, iii) difficulty to deal with nonlinear processes, and iv) mathematical complexity of the control design [5]. Therefore, it is interesting for academic and industrial communities the tuning aspect of PID controllers, especially with a reduced number of parameters to be selected and a good performance to be achieved when dealing with complex processes [4].

This paper discusses the intelligent tuning and design aspects of a PID controller based on the universal model of the plant, where there is only one design parameter to be selected. The proposed controller has the same structure of a conventional three-term PID controller. The design is based on a version of the generalized minimum variance control of Clarke and Gawthrop [6]. Simulation tests are shown for nonlinear plants (continuous stirred tank reactor and heat exchanger). Fuzzy and neural methods are included in order to improve the design of the PID controller.

This paper is organized as follows. The design idea of the PID controller, based on the universal model, is derived in section 2. Tuning aspects of the proposed approach is shown in section 3. Applications and conclusions are given in sections 4 and 5, respectively.

2 Control Design

In this section a PID control design is proposed based on an alternative representation for dynamic systems [7; 8]. The system output can be approximated by the universal model defined as follows:

$$\begin{aligned}\hat{y}(k+d) = & \sum_{i=0}^{N} \Delta^i y(k) + (d-1)b \sum_{i=0}^{N} \Delta^i u(k-1) + \\ & + b\left\{u(k-1) - \sum_{i=0}^{N-1} \Delta^i u(k-2)\right\}\end{aligned} \tag{1}$$

where Δ is the backward difference, d is the dead time, $y(k)$ is the output, $u(k)$ is the input, and $\hat{y}(k)$ is an estimate of the system output.

It can be observed that the universal model is simple and has only one parameter, b, which contains the system information. Considering Park *et al.*[7] approach the dimension of the universal model, N, can be chosen. The parameter b can be derived from the previous measurements of the system, i.e., if the values of $y(k)$, $\sum_{i=0}^{N} y(k-1)$ and $\Delta^N u(k-1-d)$ are given, then, from equation (1), b is calculated as follows:

$$b = \frac{\left\{y(k) - \sum_{i=0}^{N} \Delta^i y(k-1)\right\}}{\Delta^N u(k-1-d)} \tag{2}$$

Using the universal model, equation (1), and the generalized minimum variance control law, equation (3), it is possible to obtain the desired PID structure as follows [6]:

$$u(k) = \frac{1}{Q(z^{-1})} \left\{\Lambda(z^{-1}) y_r(k) - [\Gamma(z^{-1})\hat{y}(k+d)]\right\} \tag{3}$$

$$u(k) = \frac{1}{H(z^{-1})} \left[\Lambda(z^{-1}) y_r(k) - \Gamma(z^{-1}) \sum_{i=0}^{N} \Delta^i y(k)\right] \tag{4}$$

where $y_r(k)$ is the reference, $\Lambda(z^{-1})$ and $\Gamma(z^{-1})$ are weighting polynomials tuned by the operator and

$$
\begin{aligned}
H(z^{-1}) &= Q(z^{-1})+ \\
&+ \Gamma(z^{-1})b\left\{\left[1+(d-1)\sum_{\mathrm{i=0}}^{N}\Delta^{i}\right]z^{-1} - \sum_{\mathrm{i=0}}^{N-1}\Delta^{i}z^{-2}\right\}
\end{aligned}
\tag{5}
$$

Next, $N = 2$, $d = 1$ and $\Gamma(z^{-1}) = A(z^{-1}) = 1$ are applied to equations (4) and (5). Since $Q(z^{-1})$ is free to be chosen, then it is possible to tune $Q(z^{-1}) = q_0\Delta + \tilde{Q}(z^{-1})$, (in order to guarantee offset elimination for setpoint and load changes), where $\tilde{Q}(z^{-1})$ is a polynomial defined as

$$
\tilde{Q}(z^{-1}) = \Gamma(z^{-1})b\left\{\left[1+(d-1)\sum_{i=0}^{N}\Delta^{i}\right]z^{-1} - \sum_{i=0}^{N-1}\Delta^{i}z^{-2}\right\} \tag{6}
$$

Therefore, the digital PID control law, that has the same structure of a conventional PID controller, is given by

$$
\Delta u(k) = \frac{1}{q_0}\left\{y_r(k) - [3 - 3z^{-1} + z^{-2}]y(k)\right\} \tag{7}
$$

where q_0 is the single parameter to be tuned, it can penalize the control effort and it can determine the closed-loop dynamic of the system. For applications, it is desirable to have a systematic approach for obtaining a satisfactory value of q_0. Similar PID control design was proposed by Cameron and Seborg [9] with, at least, four parameters to be tuned. Figure (1) shows the I+PD structure that can be related to the control structure of the equation (7). The I+PD structure presents advantages over the conventional PID structure. The I+PD algorithm is implementing the proportional (P) and derivative (D) parts with the output and, therefore, avoiding the ringing state for step setpoint changes in the overall control input.

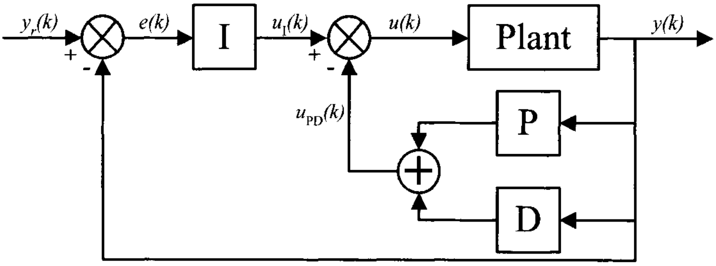


Fig. 1. I+PD control system.

The representation of the equation (7) can be written to the standard PID form through

$$
K_c = K_i = K_d = 1/q_0
$$

where K_c, K_i and K_d are the proportional, integral and derivative gains, respectively.

3 Tuning and Analysis

The tuning procedure of q_0 for the PID controller can be done from different control techniques. For the case studies of this paper fuzzy and neural methods are implemented.

3.1 Fuzzy Approach

In this approach the tuning of the controller is done using fuzzy logic issues [10]. In order to manipulate information of the dynamic system, it is necessary to use $e(k)$ as input and the variation of $e(k)$, denoted by $\Delta e(k)$. Then, the fuzzy tuner is designed with a rule base that has two linguistic input variables, E and DE, with respect to the crisp input variables $e(k)$ and $\Delta e(k)$. A linguistic output variable, O, with respect to the crisp parameter $q_0(k)$, is also included. The control law has the following form:

$$\Delta u(k) = \frac{1}{q_0(k)} \left\{ y_r(k) - [3 - 3z^{-1} + z^{-2}] y(k) \right\} \tag{8}$$

Figure (2) shows the idea of the fuzzy tuning based on the Eq. (8).

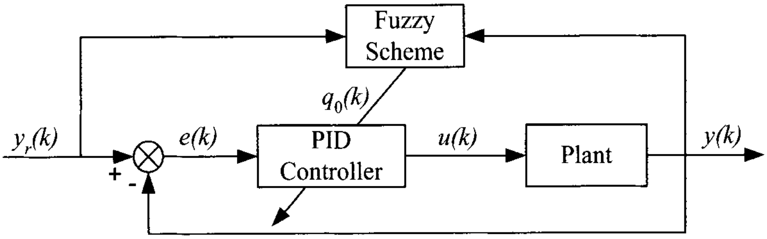


Fig. 2. PID control system with fuzzy tuning.

Linguistic sets, shown in Table (1) and membership functions, shown in Fig. (3), are defined by the linguistic variables E, DE and O in the universes of discourse U, V and W, respectively. In Fig. (3) the scale factors Se, Sde, and So are defined by using the Nelder-Mead optimization method [11], and $esc = Sc/$(number of output membership functions). Table (2) presents the rule base of the fuzzy tuner.

Table 1. Fuzzy Sets.

NE – $e(k)$ is negative ZE – $e(k)$ is zero PE – $e(k)$ is positive	ZO – $q_0(k)$ is zero PTO – $q_0(k)$ is positive tiny PSO – $q_0(k)$ is positive small
NDE – $\Delta e(k)$ is negative ZDE – $\Delta e(k)$ is zero PDE – $\Delta e(k)$ is positive	PMO – $q_0(k)$ is positive medium PBO – $q_0(k)$ is positive big

Table 2. Rule base for the fuzzy tunner.

$\Delta e(k)$ $e(k)$	**NDE**	**ZDE**	**PDE**
NE	PMO	PBO	PMO
ZE	PPO	PPO	PPO
PE	PMO	PBO	PMO

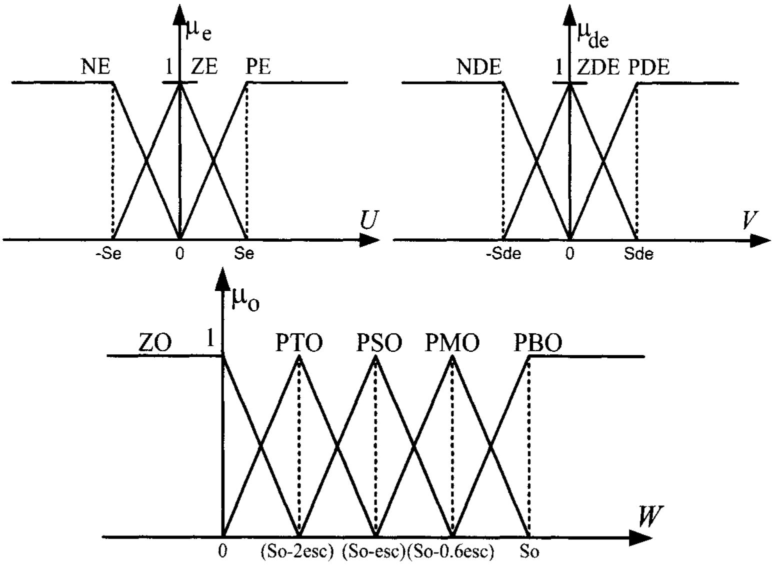


Fig. 3. Membership functions of fuzzy tuner.

3.2 Neural Approach

Neural networks (NNs) have been used in a vast variety of different control structures and applications, serving as controllers or as process models or parts of process models. They have been used to recognize and forecast disturbances, to detect and to recognize faults, to combine data from partially redundant sensors, to perform statistical quality control, and to adaptively tune conventional controllers such as PID controllers. Many different structures of NNs have been used, with feedforward networks, radial basis functions, and recurrent networks being the most popular.

A NN presents the following features: generalization, learning, fault tolerance, adaptability, inherent parallelism and abstraction. Thus, the major advantage of NN learning is the ability to accommodate poorly modelled and nonlinear dynamical systems.

Fig. (4) shows the neural network idea to update $q_0(k)$ with respect equation (8).

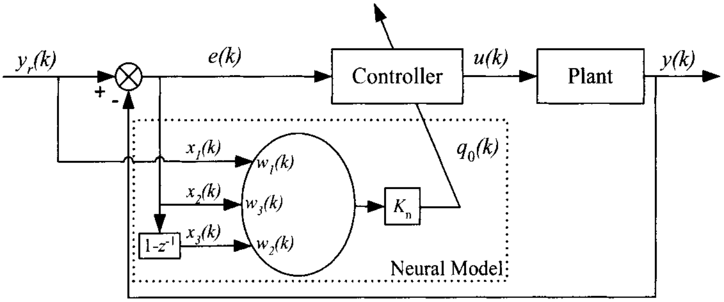


Fig. 4. PID control system with neural tuning.

The adaptive neural model shown in Fig. (4) for tuning the PID control parameter was proposed by Wang *et al.* [12]. The neuron output $q_0(k)$ can be derived as

$$q_0(k) = q_0(k-1) + K_n \frac{\sum_{i=1}^{n} w_i x_i(k)}{\sum_{i=1}^{n} |w_i|} \tag{9}$$

where, $x_i(k)$ $(i = 1, 2, , n)$ denote the neuron inputs; $K_n > 0$ is the neuron proportional coefficient; w_i are the connection weights of each $x_i(k)$ and are determined with learning rule or optimization algorithms. In this paper there are three inputs of neural $(n = 3)$ defined as $x_1(k) = y_r(k)$, $x_2(k) = e(k)$ and $x_3(k) = \Delta e(k)$.

3.3 Stability Analysis

The stability of systems with respect of modeling error has received attention in the control literature. Based on Fig. (5), Wu *et al.* [13] have derived a time-dependent robustness index defined as

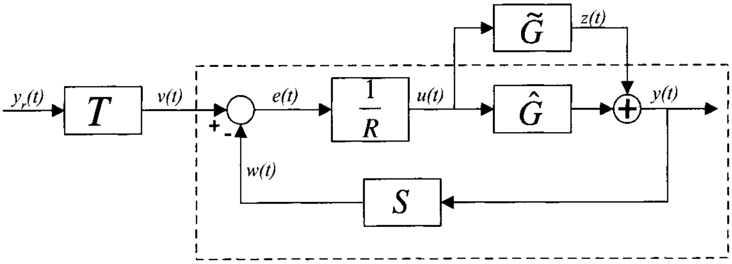


Fig. 5. RST loop with additive uncertainty.

$$RI(k) = \frac{\left|\sum_{i=0}^{k} \hat{y}(i) - \sum_{i=0}^{k} y(i)\right|}{\left|\frac{\sum_{i=0}^{k} \hat{y}(i) \sum_{i=0}^{k} v(i)}{\sum_{i=0}^{k} w(i)}\right|} \tag{10}$$

where $y(k)$, $v(k)$ and $w(k)$ are shown in Fig (5). Here $\hat{y}(k)$ can be determined from the identified model, equation (1), and $v(k)$ and $w(k)$ are obtained from the equations (7) and (11) as follows:

$$R(z^{-1})u(k) = T(z^{-1})y_r(k) - S(z^{-1})y(k) \tag{11}$$

where

$$\begin{aligned} R(z^{-1}) &= \Delta q_0 \\ S(z^{-1}) &= 3 - 3z^{-1} + z^{-2} \\ T(z^{-1}) &= 1 \end{aligned}$$

$RI(k)$ provides an on-line indication of the process robustness. When $RI(k)$ is less than one, the closed-loop system is stable.

4 Simulation Results

This section presents the simulation results for the tuning procedures described in section 3 for the digital PID controller. The procedures have been applied to a continuous stirred tank reactor (CSTR).

4.1 CSTR Application

The study consists of an unstable nonlinear continuous stirred tank reactor (CSTR) as shown in Fig. (6). Discrete dynamic equations for the reactor are given by [14]

$$\begin{aligned} x_1(k+1) &= x_1(k) + \\ &+ T_s\left[-x_1(k) + D_a(1 - x_1(k))e^{\frac{x_2(k)}{1+x_2(k)/\gamma}}\right] \end{aligned} \tag{12}$$

$$\begin{aligned} x_2(k+1) &= x_2(k) - T_s x_2(k)(1+\beta) + \\ &+ T_s\left[BD_a(1 - x_1(k))e^{\frac{x_2(k)}{1+x_2(k)/\gamma}} + \beta u(k)\right] \end{aligned} \tag{13}$$

where x_1 and x_2 represent the dimensionless reactions concentration and reactor temperature, respectively, and the control input, u, is the dimensionless cooling jacket temperature. The physical parameters of the CSTR model equations are D_a, γ, B and β which correspond to the Damköhler number, the activated energy, the heat of reaction and the heat transfer coefficient, respectively. Nominal system parameters are $D_a = 0.072$, $\gamma = 20$, $B = 8$, $\beta = 0.3$, and T_s is the sampling time (0.2 seconds).

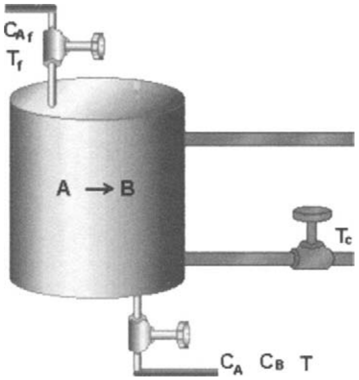


Fig. 6. CSTR plant.

In addition, Fig. (7) shows the phase plane of the CSTR, where the upper and lower steady-states are stable, while the middle one is unstable. So, this kind of behavior is a good challenge to assess the PID control algorithm.

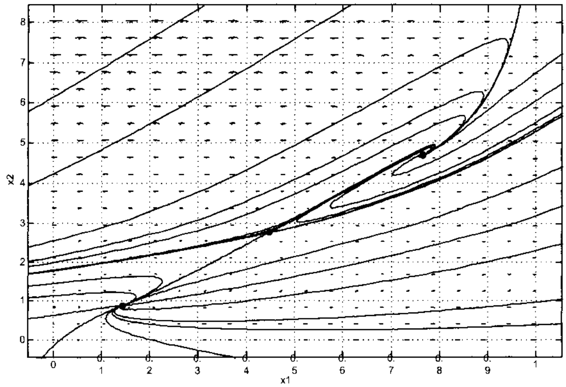
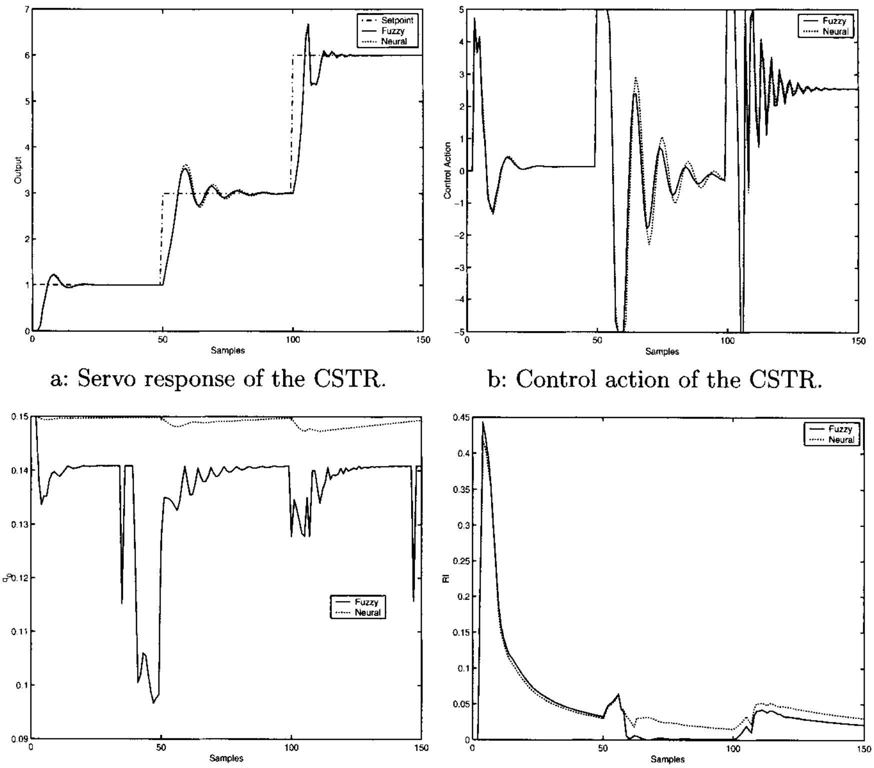


Fig. 7. Phase plane of the nonlinear CSTR.

Responses for servo essays are given in Figs. (8.a) and (8.b). For analysis of this behavior, the reference signal is changing such as: $y_r = 1$ (from sample 1 to 47), $y_r = 3$ (from sample 48 to 92) and $y_r = 6$ (from sample 93 to 140). For the fuzzy tuner the scale factors are $Se = 0.0001$, $Sde = 0.5202$, $Sc = 0.152$, $esc = 0.0394$, and for the neural tuner the parameters are $w = [58.981,\ 1.3417 \times 10^3,\ -1.4 \times 10^3]$, $K_n = 7 \times 10^{-4}$.

Results from Figs. (8.a) and (8.b) illustrate that the servo behavior of the system is good with a small control variation for $y_r = 1$. However, for references $y_r = 3$ and $y_r = 6$, it is necessary more energy for the control action in order to eliminate the steady-state error. It is due the complexity of the CSTR in these regions, as shown in Fig. (7).

Figure (8.c) shows the behavior of $q_0(k)$ calculate by fuzzy and neural tuning approaches. The gain is varying, depending on the change of the reference signal and a good PID control performance is obtained.



a: Servo response of the CSTR.

b: Control action of the CSTR.

c: Behavior of $q_0(k)$.

d: Robustness index of the CSTR.

Fig. 8. Closed-loop responses of the CSTR.

Figure (8.d) shows the behavior of the robustness index defined by equation (10). The index value is close to zero and it is showing that the closed-loop control system is stable for all operational points in a nonlinear application.

The results clearly demonstrate the successful performance of the PID controller with intelligent tuning approaches when applied to a CSTR.

5 Conclusion

In this paper a digital control with a conventional PID structure was developed. The design method is based on the universal plant representation. The proposed PID controller structure is very interesting from the viewpoint of operators because the control design presents only one parameter to be tuned. Simulation tests for nonlinear plant have indicated that the PID controller with fuzzy and neural tuning performed very well and, therefore, demonstrating the successful application of the intelligent tuning approaches.

Although the paper had shown only the step responses to setpoint changes, good disturbance rejection properties can be obtained.

Further works include multivariable, disturbance and time-delay applications.

References

[1] K. J. Åström and T. Hägglund, *PID controllers: theory, design and tuning*. Instrument Society of America, 2nd ed., 1995.

[2] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 09, pp. 1163–1175, 2001.

[3] T. Yamamoto, A. Inoue, and S. L. Shah, "Generalized minimum variance self-tuning pole-assignment controller with a PID structure," in *IEEE Int. Conf. on Control Applications*, (Hawaii, USA), pp. 125–130, 1999.

[4] C. W. Alexander and R. E. Trahan, "A comparison of traditional and adaptive control strategies for systems with time delay," *ISA Transaction*, vol. 40, pp. 353–368, 2001.

[5] W. K. Wojsznis and T. L. Blevins, "Evaluating PID adaptive techniques for industrial implementation," in *American Control Conference*, (Anchorage, AK), pp. 1151–1155, 2002.

[6] D. W. Clarke and P. J. Gawthrop, "Self-tuning controller," *IEE Proceedings*, vol. 132, pp. 929–934, 1975.

[7] Y.-M. Park, J.-H. Lee, S.-H. Hyun, and K. Y. Lee, "A synchronous generator stabilizer based on a universal model," *Electrical Power & Energy Systems*, vol. 20, pp. 435–442, 1998.

[8] R. R. Sumar, "A new platform for designing a digital PID control (in Portuguese)," Internal Report, Department of Automation and Systems, UFSC, Brazil, 2003.

[9] F. Cameron and D. E. Seborg, "A self-tuning controller with a PID structure," *Int. J. Control*, vol. 38, pp. 401–417, 1983.

[10] S. Chiu, "Using fuzzy logic in control applications: beyond fuzzy PID control," *IEEE Control Systems Magazine*, vol. 18, no. 05, pp. 100–104, 1998.

[11] J. A. Nelder and R. Mead, "A simplex method for funtion minimization," *Computer Journal*, vol. 07, pp. 308–313, January 1965.

[12] N. Wang, Z. Zheng, and H. Chen, "Model-free PID controller with gain scheduling for turning processes," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, (Beijing), pp. 2424–2429, 2003.

[13] W.-T. Wu, K.-C. Chen, and Y.-J. Jang, "Robustness index for adaptive control based on pole-zero placement," *Int. Journal of Systems Science*, vol. 20, no. 10, pp. 1967–1978, 1989.

[14] C.-T. Chen and S.-T. Peng, "A nonlinear control strategy based on using a shape-tunable neural controller," *Journal of Chemical Engineering of Japan*, vol. 30, pp. 637–646, 1997.

Part III

Design

Dynamic Reconfiguration Algorithm for Field Programmable Analog Scalable Device Array (FPADA) with Fixed Topology

Peter Tawdross and Andreas König

Institute of integrated sensors,
University of Kaiserslautern, Germany.
{tawdross,koenig}@eit.uni-kl.de

Summary. The sensor electronics and mixed signal processing are sensitive to the system changes by aging, temperature distribution inside the chip, or additional factors of the external environment. These factors can't effectively be considered in the design phase. For some hardware trimming is done to increase the accuracy of the circuits at a certain temperature, which increase the cost but still can't deal with all the mentioned problems. In this paper, we present a soft computing dynamic reconfiguration algorithm for FPADA dynamically reconfigurable hardware (DRHW) for compensating the temperature distribution, aging, and other environment influences dynamically without changing the main topology of the device, where the topology is done by human or by the synthesis tools. The main advantage of our algorithm is that it keeps the hardware structure, and changes only the dimensions of the model to evolve it with the new environment in order to have a reliable DRHW with a predictable performing. Evolving this hardware is done by evolving the dimensions of parameters of the device by the standard measurement techniques. The objective of the evolution is to optimize the set of all the device parameters. In our experimental work, we developed a working environment for DRHW, and demonstrated it with an operational amplifier for compensating the effect of temperature distribution inside the amplifier, this amplifier is evolved for optimizing selection of relevant amplifier parameters.

1 Introduction

Most of the hardware is designed according to the given specification at the operating point temperature which is usually assumed to be 27°C. Usually the aging, the temperature distribution inside the chip due to externally distributed heat, or internal heating at a spot point can't effectively be considered in the design phase. One of the other external effects on the system is the variable external load. For variable external load, the system must be reconfigured while the load is connected, which means that it is not suitable for all the applications as in some applications, the output of the system under test can negatively effect the whole system *(e.g., if an amplifier is controlling a heater, applying the test signals can overheat or destroy some parts in the system)*

Recently, the genetic programming has been used as a synthesis tool [6], which can be used for evolvable hardware (EHW) as well [5, 8, 9]. The genetic programming design a circuit which is a black-box circuit with unknown structure. Moreover, most of the evolvable hardware uses the fitness function in such a way to minimize the error sum between output signal and the reference *(required)* output signal [6, 5, 8, 9]. As a result, the system can accept an output which generates noise with low amplitude, or it can accept a non-stable system or critically stable system rather than a system with a phase-shift or low gain. Some designs use the error between the required output and actual output voltage in frequency domain for the fitness function, e.g., a low-pass filter, or a high-pass filter, which mean that the evolution can consider any improvement in the system as an error if the output and the reference output differ.

Deal with the dynamic environment increases complexity of the optimization algorithm as the environment is time dependent. Using a big mutation improves performance for fast changing environments [3]. Grefenstette modified the standard genetic algorithm by adding partial hyper-mutation step, that replaces a percentage of the population by randomly generated individuals [4].

In our paper, we used a fixed topology hardware which gives us the ability to modify the hardware specification [10] to fit the application without losing the hardware principle structure, in other words, the evolution incorporates a priori knowledge to design the hardware. Our goal is to develop a soft computation environment for fast, reliable adaptation of analog reconfigurable hardware from its starting point without changing the hardware structure. The fitness function is designed to evolve the hardware specification, not to reduce the error between the reference output and output. The results of this evolution is a predictable performing hardware. A predictable behavior and known topology are the essentials for industrial acceptance.

In our experimental work, we used a reconfigurable amplifier for demonstrating our algorithm. An overview about the used reconfigurable amplifier is given in Sect. 2. In Sect. 3, we demonstrate our algorithm for reconfiguring the operational amplifier. In Sect. 4, we show our experimental results. Finally in Sect. 5, we conclude our work.

2 Reconfigurable Operational Amplifier Baseline

The hardware which we target on should have basic circuits with programmable dimensions, and programmable interconnection between them. The basic circuits should be sufficient for constructing any analog hardware. The connection between the blocks is done by the known synthesis methods, or

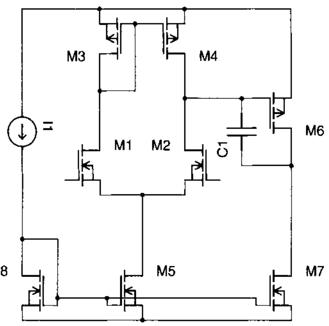


Fig. 1. The used amplifier model

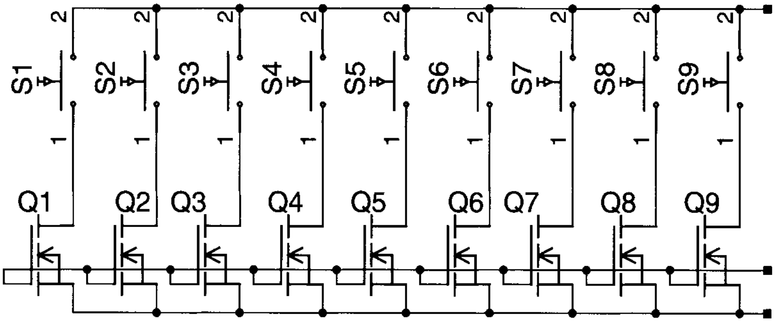


Fig. 2. The structure of one scalable transistor

human design. We constrict in the first step of our experimental work to the hardware described in [7]. We used a Miller model operational amplifier as given in Fig. 1. Each transistor of the amplifier is replaced by an array of parallel CMOS transistors each of them has an electronic switch with series in order to be able to choose the width of the transistors by the combination of the transistors by programming the electronic switches. They used an array of eight transistors with width of 1,2,4,8,16,32,64 and 128 micrometers. In our experiments, we added an additional transistor with width of 256 in parallel with them to have a wider range of selection. The length of the transistor is fixed, we chosen 10 micrometer[1] in our initial experiment. Each of the NMOS transistors is replaced with the configuration at Fig. 2, each of the PMOS transistors is replaced with an equivalent configuration with PMOS transistors[2].

[1] The length in [7] is $2\mu m$

[2] We used standard transistors with $\gamma = 1.3$, $\lambda = 0.01$, $K_p = 17\mu$ for NMOS and $\gamma = 0.6$, $\lambda = 0.02$, $K_p = 8\mu$ for PMOS

3 The Reconfiguration Algorithm

In our current experimental works, we use an extrinsic evolution approach, which means that the circuit is simulated by the computer and not downloaded directly to a real hardware. The main reason of using extrinsic approach is that the required hardware is currently under development. Our framework in Fig. 3, is written in C++ programming language, the used genetic algorithm library is GAlib. The simulation of the circuit is done by ngspice in batch mode which save the outputs to a raw file. An additional post-processing is done afterword in order to compute the parameters of the amplifier from the raw files. We assumed ideal electronic switches in our current simulation model. In the experiment, we try to change the transistor widths of the amplifier's transistors in order to compensate the temperature difference inside the amplifier. We assumed that all the transistors are working at the nominal temperature of 27°C except $M1$ is working at 100°C which means different of 73°C at the input transistors. As a result, the amplifier offset voltage is increased.

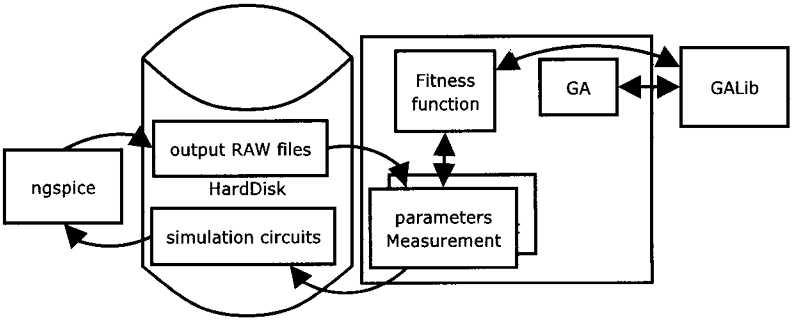


Fig. 3. Framework for dynamic reconfiguring the operational amplifier

The basic parameters of the operational amplifier are the open-loop gain, open loop frequency response, input-offset voltage, common-mode gain, power-supply rejection ratio, common-mode input- and output-voltage, open-loop output resistance, and transient response including slew rate, and settling time.

In our algorithm, the objective is to optimize the settling time, slew rate, open-loop gain, zero offset, and the CMRR (common mode rejection ratio). We choose these parameters as we believe that these parameters are sufficient to evolve the amplifier for proving our principle.

The settling time is the time required after a stimulus for the output to center and remain within a specified narrow band centered on its steady-state value. The slew rate is the measure of how fast a circuit is able to respond to fast changes in amplitude in the source signal. The open-loop gain is the gain of the amplifier when the amplifier has no feedback connection. The zero offset voltage is the output voltage when the inputs of the amplifier are grounded.

The system has a minimum requirement that it tries to fulfill, any additional improvement is not considered to the system, but accepted without affecting the fitness function. If the requirement of the amplifier is higher than the model ability, the system will try to be improved each new generation without reaching the zero error as it is not possible, in this case, the system will try to maximize the performance.

For the settling time, slew rate, we compute the rising settling time, falling settling time, rising slew rate and falling slew rate as equivalent objective functions. The totally used objective functions in our work are rising settling time, falling settling time, rising slew rate, falling slew rate, open-loop gain, CMRR, and zero offset voltage. The objectives for the slew rate, CMRR, and the open-loop gain are to be maximized, the objective for the settling time, offset voltage are to be minimized.

The error function for each of the parameters that have to be minimized is as in equation 1.

$$E_i = \begin{cases} \dfrac{f_i - spec_i}{spec_i} & \forall spec_i < f_i \\ 0 & \forall spec_i \geq f_i \end{cases} \tag{1}$$

Where f_i is the measure fitness value of objective i, $spec_i$ is the specified fitness value that the user specifies for the objective i .

The error function for maximization is as in equation 2

$$E_i = \begin{cases} \dfrac{spec_i - f_i}{spec_i} & \forall spec_i > f_i \\ 0 & \forall spec_i \leq f_i \end{cases} \tag{2}$$

As shown in the equations, all the error functions are normalized to decrease the dependency between the error function and the range of the parameters, e.g. the open-loop gain is in order of 10^4, the offset voltage is in order of $10^{-3}V$.

The fitness function of the amplifier at our experiment is in equation 3.

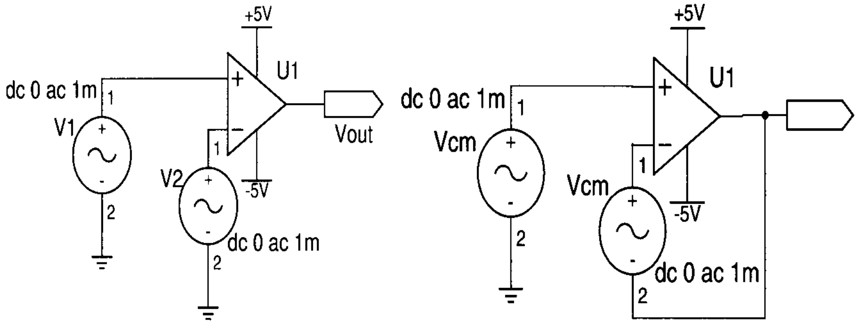
$$F = \sum K_i \times E_i \tag{3}$$

Where K_i is a weight, given by the user according to the required priority of the specification parameters.

Minimizing F minimizes $\frac{f_i - spec_i}{spec_i}$ and maximizes $\frac{spec_i - f_i}{spec_i}$.

If it is required to have a similar falling settling time and rising settling, or falling slew rate and rising slew rate, an additional error functions will be added with the error of the absolute value between the falling and the rising function.

To calculate the open-loop gain, we simulate the circuit at Fig. 4(a) with AC simulation. The open-loop gain is the minimum of the gain within a given frequency range. To calculate the CMRR, the circuit in Fig. 4(b) is simulated using AC simulation.



(a) The circuit to simulate the open-loop gain

(b) The circuit to simulate the CMRR

Fig. 4. simulation circuits for open loop and CMRR

The CMRR at the frequency i is calculated as following [2].

$$\frac{V_{out}}{V_{cm}} = \frac{\pm A_c}{1 + A_v - \left(\pm\frac{A_c}{2}\right)} \approx \frac{\mid A_c \mid}{A_v} = \frac{1}{CMRR} \tag{4}$$

Where A_v is the open loop gain, A_c is the differential frequency response, V_{cm} is the input voltage as in Fig. 4(b), and V_{out} is the output voltage of the amplifier.

In equation 5, the $CMRR$ as a function of the frequency is shown.

$$CMRR(f) = \frac{V_{cm}(f)}{V_{out}(f)} = \frac{1mV}{V_{out}(f)} \tag{5}$$

Where $CMRR(f)$ is the $CMRR$ at the frequency f, $V_{cm}(f)$ is the input voltage at frequency f, and $V_{out}(f)$ is the output voltage at frequency f.

We use the average $CMRR(f)$ over all the simulated frequency range as the $CMRR$ of the amplifier.

The response of the amplifier when changing the level from low level to high level can be differed from the response when changing from high level to low level. For this reason, we calculate rising settling time and falling settling time instead of settling time, and calculate rising slew rate, falling slew rate instead of only slew rate. In our experiment, we defined the settling time by the time needed to the system to have error at the output less than 1% of the peak input voltage after applying a step input. The circuit used for simulation for the slew rate and settling time calculation is a unity gain amplifier with a square wave input. The rising settling is calculated by an algorithm which takes the time of the starting point by detecting a high level after a low level, update the endpoint to the current point until the input level changed to low, or the output input error is within 1%. Similarly he falling settling time is calculated.

The rising slew rate is calculated by dividing the voltage difference from $2V$ to $3V$ $((1V))$ by the time taken to be risen. Similarly the falling slew rate by dividing the difference from $3V$ to $2V$ $((1V))$ by the time taken.

The zero offset voltage is measured by simulating a circuit of an amplifier with grounded inputs by transient simulation. The output when the amplifier is stable is the zero offset voltage.

The implemented genetic algorithm runs for 100 generations, each of the generations has a population size of 50. The used initialization scheme is the uniform initializer. Selecting the parents for mating is done by the tournament selection scheme. The number of generations is independent of the convergence as it is expected to have a certain period for reconfiguration in a real dynamic system. The dependence can be only with some minimum requirement, which mean that the system can't work even if the reconfiguration specified period is over if performance of the amplifier can't fulfill a certain requirement. In our experiment, we fix the number of generation without check for any minimum requirement, as the amplifier was already working before evolution, but with lower performance.

Each chromosome in the population has 7 genes, each gene represent a transistor. A gene has an integer value in the range between 1 and 255.

The used crossover probability is 0.1, the crossover scheme is the arithmetic crossover. The mutation probability start by 0.50. If the fitness function does not converge for 5 iterations, the mutation will be multiplied by 0.97 for each new iteration without convergence until the fitness function start to converge again. The mutation rate is big as increasing the mutation rate improves performance for faster changing environments [3]. The mutation starts with a big probability to cover more search space, then the probability is decreased in order not to miss the optimal point in the search. The used mutation scheme is the Gaussian mutation. The used genetic algorithm is simple genetic algorithm which has non-overlapping population. The elitism is used to keep the best five genomes it ever encounters.

4 Experimental Results

First the effect of non-homogeneous temperature distribution over the amplifier area is investigated. From nominal 27°C of all other devices, the transistor $M1$ temperature is increased to 100°C, results an offset voltage at the amplifier output as shown in Fig. 5. By applying the reconfiguration algorithm, Figs. 6(a) and 6(b) show two different results due to different weights in the fitness function. The algorithm improves the zero offset, the slew rate and the settling time of the original amplifier.

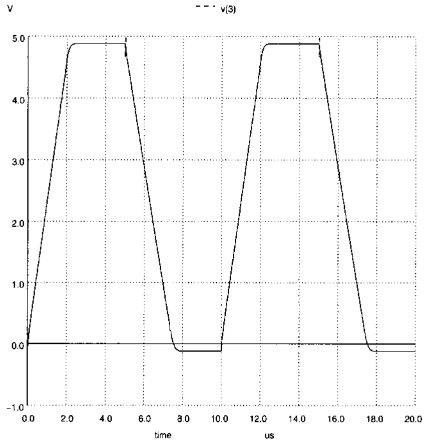
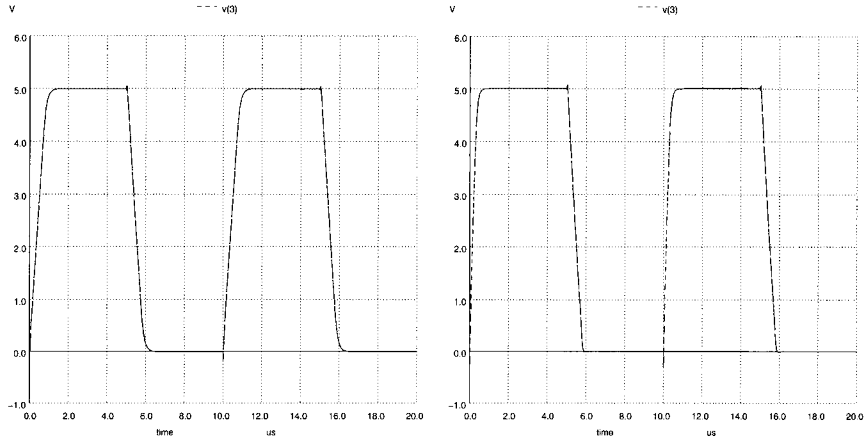


Fig. 5. The amplifier output before reconfiguration



(a) The amplifier output after evolution (b) Other amplifier output after evolution

Fig. 6. The operational amplifier after reconfiguring by different weighting at fitness function

In Figure 5, zero offset was about $100mV$, but it is decreased to $200\mu V$ in the result in Fig. 6(a). In the result in Fig. 6(b), the offset is about $1mV$, but the slew rate and the settling time is better than the result in Fig. 6(a). The aspect ratios of the transistors before evolution, and the two different configurations due to evolving under different weights in the fitness function are shown in Table 1.

All the objective functions are functions of the widths of the transistors. The result of changing the weights of the objective functions is getting an amplifier with different configuration (transistor widths) as shown in Table 1.

Table 1. The aspect ratios of the transistors before and after reconfiguration

Transistor Nr.	Before reconfiguration W/L	1^{st} Reconfiguration W/L	2^{nd} Reconfiguration W/L
M1	43/10	55/10	18/10
M2	43/10	67/10	14/10
M3	10/10	312/10	357/10
M4	10/10	206/10	289/10
M5	38/10	122/10	476/10
M6	344/10	462/10	478/10
M7	652[3]/10	170/10	216/10
M8	38/10	69/10	96/10

5 Conclusion

In our work, an algorithm for a dynamically reconfigurable analog hardware with fixed topology is demonstrated to compensate the internal temperature distribution of an operational amplifier. A multi-objective function is used to optimize the amplifier parameters, which is based on the standard measurements of the device. The reconfiguration algorithm succeeded to reduce the offset from $100mV$ to about $200\mu V$. As the hardware topology for the used dynamically reconfigurable hardware is known and the evolution is done in a way to optimize the parameters. The behavior of the circuit is predictable, which is required for industrial acceptance and applications. It is assumed, that the same scenario can be used to compensate the aging, and the other influences of the environment. As this system have some of the self-x specifications *(self-configuring, self-optimizing, and self-healing)*, it is an organic computing [1] system which can be integrated into a single chip. The dynamic reconfiguration gives promises to increase the lifetime, quality of service, the performance of the device, and eliminate the need for trimming in the production phase.

In the next phase of our research, we are going to include full specification *(all the circuit parameters)* for optimization. We are interested to investigate alternative methods for multi-objective function optimization *(e.g. Pareto)*. Alternative dynamic environment approachs will be invesitgated in order to deal with more varying environment *(e.g. adaptive mutation operator [3] and replace a percentage of the population by randomly generated individuals [4])*.

We will adapt our simulation environment to the target technology, and relate the technology models and parameters. We will consider real electronics switches as well in order to verify its compatibility with real systems. Then we will simulate our algorithm with more complex analog designs to verify its generality. On the availability of proprietary hardware, we will apply intrinsic evolution and more complex sensor electronics.

[3] This width is reduced by the evolution to be less than 512

References

1. Organic computing homepage: http://www.organic-computing.org/.
2. Phillip E. Allen and Douglas R. Holberg, editors. *CMOS Analog Circuit Design.* Oxford University Press, 1987.
3. H. G. Cobb. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments,". *Tech. Rep. AIC-90-001*, 1990.
4. John J. Grefenstette. Genetic algorithms for changing environments. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature 2 (Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, Brussels 1992)*, pages 137–144, Amsterdam, 1992. Elsevier.
5. M. Hartmann, P. Haddow, and F. Eskelund. Evolvable hardware solutions for extreme temperature electronics. In *The 2002 NASA/DoD Conference on Evolvable Hardware*, pages 36–45, Alexandria, Virginia, 15-18 July 2002. Jet Propulsion Laboratory, California Institute of Technology, IEEE Computer Society.
6. J.R. Koza, D. Andre, F.H. Bennett III, and M.A. Keane. Design of a 96 decibel operational amplifier and other problems for which a computer program evolved by genetic programming is competitive with human performance. 1997.
7. S.K.Lakshmanan and A.König. A contribution to reconfigurable analog electronics by a digitally programmable sensor signal amplifier. 2005.
8. A. Stoica, D. Keymeulen, R. Zebulum, A. Thakoor, T. Daud, G. Klimeck, Y. Jin, R. Tawel, and V. Duong. Evolution of analog circuits on field programmable transistor arrays. In *The Second NASA/DoD workshop on Evolvable Hardware*, pages 99–108, Palo Alto, California, 13-15 July 2000. IEEE Computer Society.
9. A. Stoica, R. Zebulum, and D. Keymeulen. Progress and challenges in building evolvable devices. In *The Third NASA/DoD workshop on Evolvable Hardware*, pages 33–35, Long Beach, California, 12-14 July 2001. IEEE Computer Society.
10. Ricardo Salem Zebulum, Marco Aurélio C. Pacheco, and Marley Maria B.R. Vellasco, editors. *Evolutionary Electronics. Automatic Design of Electronic Circuits and Systems by Genetic Algorithm.* CRC Press, 2002.

Advances in the Application of Machine Learning Techniques in Drug Discovery, Design and Development

S.J. Barrett† and W.B. Langdon*

† Analysis Applications, Research and Technologies, GlaxoSmithKline
R&D, Greenford Rd, Greenford, Middlesex, UB6 0HE. UK
*Computer Science, University of Essex, Colchester CO4 3SQ, UK

Abstract. Machine learning tools, in particular support vector machines (SVM), Particle Swarm Optimisation (PSO) and Genetic Programming (GP), are increasingly used in pharmaceuticals research and development. They are inherently suitable for use with 'noisy', high dimensional (many variables) data, as is commonly used in cheminformatic (i.e. In silico screening), bioinformatic (i.e. bio-marker studies, using DNA chip data) and other types of drug research studies. These aspects are demonstrated via review of their current usage and future prospects in context with drug discovery activities.

1 Introduction

Pharmaceutical discovery and development is an evolving [Ratti & Trist, 2001] cascade of extremely complex and costly research, comprising many facets [Ng, 2004] which create a vast diversity of data and sub-problems [Butte, 2002; Schrattenholtz, 2004; Watkins & German, 2002; Roses, 2002]. Drug design and optimisation increasingly uses computers [Hou and Xu, 2004; Schneider and Fechner, 2005] and more commonly against vast 'integrated' research datasets constructed from large inhomogeneous combinations of data (from disparate sources and disciplines) to answer novel lines of inquiry, and for the generation of research hypotheses.

Conventional statistical methods are currently better known and understood by Pharmaceuticals R&D scientists who benefit from the traditional statistical support toward design of experiments, data assessment, etc. However statistical groups are increasingly using other computational methods and recognising alternative approaches [Hand, 1999; Breiman, 2001], as existing (usually hypothesis testing) methods are found lacking. This is generally due to the increasing need for data exploration and hypothesis generation in the face of growing data, problem complexities, and *ad hoc* experimental design inadequacies from compromises due to cost and lack of prior knowledge. As individual techniques may only partly cope with these problems, multiple methods are often used for comparative analyses. Whilst conventional multivariate statistical methods remain of great utility, most are inherently linear lessening their suitability for a plethora of newer, more complex problems. Consequently, evaluation and early uptake of novel computational approaches continues within pharmaceuticals research, with scientists increasingly turning to recursive partitioning, Artificial Neural Networks (ANNs) and other methods. Whilst ANNs and genetic algorithms are established in traditional application areas [Jones, 1999; Solmajer and Zupan, 2004], tracking the uptake of more recent machine learning approaches is difficult due to the diversity of new applications and fragmented literature.

The newer predictive modeling approaches include Support Vector Machines (SVM) and evolutionary computing paradigms such as Genetic programming and Particle Swarm Optimisation (PSO). SVM algorithms arose [Boser *et al.*, 1992] from concepts of structural risk minimisation and statistical learning theory [Vapnik, 1995]. SVMs are commonly used for classification (SVC) and regression (SVR). They are a sophisticated synthesis of ANN-like hyperplane methodology, backed by a sound theory of learning and convergence, applying robust linear methods (and within kernel spaces for non-linear classifiers) to give excellent generalisation characteristics [Shawe-Taylor and Cristianini, 2000]. In contrast to the rigorous mathematical approach of SVMs, genetic programming [Banzhaf, *et al.*, 1998] appeals to metaphor. GP uses Darwins' natural selection to evolve a population of computer programs. The better programs are selected to be parents for the next generation. Children are created by crossover and mutation. Some are better and some are worse than their parents. Selection continually encourages better individuals to pass on their genes. Overtime and successive generations the population improves until an individual with satisfactory performance is found. Many drug discovery problems can be expressed as the problem of finding a computer program, and GP is general purpose requiring minimal assumptions and capable of solving very difficult problems. Particle Swarm Optimization (PSO) was inspired by swarms of insects, shoals of fish, etc [Eberhart, Kennedy and Shi, 2001]. In PSO's the creatures are abstracted to moving particles. These fly over the problem space. If they find a good point they are randomly attracted back to it. Fundamentally PSO also have a similar social force which attracts all the particles to the best solution to the problem found by the whole swarm.

We here review the current status of pharmaceutically relevant applications of Support Vector Machines (SVMs), Genetic Programming and Particle Swarm Optimisation and assess briefly assess their future.

2 SVM Applications in Pharmaceuticals Research

2.1 SVM in Cheminformatics and Quantitative Structure-Activity Relationship (QSAR) Modelling.

Cheminformatics in drug discovery has been reviewed by [Xu and Hagler, 2002]. An early task is the creation of virtual respresentations of molecules and assessment of their likely suitability for synthesis and viability for development for use in the body. SVC predictions of 'drug-likeness' from virtually-represented molecules are reportedly more robust than those from ANNs [Byvatov et al. 2004], achieving success in predicting chemists' intuitive assessments [Takaoka et al., 2003]. Cheminformatics combines chemical properties and high-throughput screening measurements, in large scale QSAR. Trained SVM-QSAR classifiers now enable 'virtual screening' for discovering molecules with specific therapeutic target affinities from millions of virtual representations [Jorissen and Gilson, 2005], reducing the scale of subsequent 'physical' screening of synthesised molecules. SVC 'active learning' has been used to reduce the number of drug-optimising synthesis-biotesting cycles [Warmuth et al., 2003]. Studying bio-active conformations of molecules aids understanding of mechanisms of action for improving specificity and selectivity and [Byvatov et al., 2005b] have used SVM methodology to study molecular pharmacophore patterns. [Chen, 2004] reports on SVM uses in the wider field of chemistry.

Predicting activity toward specific therapeutic targets. G-protein coupled receptors (GPCRs) are the major class of drug targets. [Suwa *et al.*, 2004] provided physicochemical features of GPCRs and their ligands to a Radial Basis Function-SVC (RBF-SVC) to predict specific G-protein couplings. [Cheng *et al.*, 2004] used an RBF-SVR to predict antagonist compound metabolism and inhibitory activity toward human glucagon receptor to select 3D QSAR features. [Byvatov, *et al.*, 2005] used binary SVC active learning to enrich dopamine receptor agonists, applying SVR to the enriched set to predict D2/ D3 receptor selectivity. [Takahashi *et al.*, 2005] used multiclass SVC to predict D1 receptor agonists, antagonists and inactives. [Burbidge, 2004] applied SVM to a variety of monoamine QSAR problems, but good performance could come with non-sparsity: a large number of training points as support vectors can severely reduce prediction speed in virtual screening. [Burbidge *et al.*, 2001a] devised an algorithm to counter this.

Predicting Absorption Distribution Metabolism Excretion Toxic (ADMET) effects. [Burbidge, *et al.*, 2001b] favourably compared SVC to ANNs, decision trees and K-nearest-neighbour (k-NN) classifiers for predicting human blood-brain barrier penetration, human oral bioavailability and protein-binding. [Brenemann *et al.* 2003] applied SVM to cell permeability prediction. Bacterial P-glycoprotein (P-gp) mediated efflux of substrate antibiotics results in drug resistance. [Xue *et al.*, 2004a] used Gaussian SVC Recursive Feature Elimination (SVC-RFE) to predict P-gp substrates, outperforming ANN and k-NN. [Xue *et al.*, 2004b] used similar approach for predicting human intestinal absorption and serum albumin binding. [Doniger *et al.*, 2002] demonstrated benefits of RBF-SVC over ANNs against a small dataset to predict central nervous system (Blood-Brain Barrier) permeability. [Norinder, 2003] had overfitting problems with SVR, needing simplex optimization for parameter and feature selection to achieve good predictors for BBB penetration and human intestinal absortion. [Liu *et al.*, 2005] used Gaussian SVR for predicting human oral drug absorption.

[Yap *et al.*, 2004] used Gaussian SVC to differentiate drugs that can cause *torsade de pointes* (TdP), an adverse drug reaction which involves multiple mechanisms. Prediction accuracy compared favourably with k-NN, ANN and C4.5. [Xue *et al.*, 2004b] also used SVC, but with RFE to predict TdP inhibition. [Tobita, *et al.*, 2005] used RBF-SVC to predict chemical inhibition of *HERG* potassium channel that is associated with heart arrhymia which can trigger TdP. Non-Steroidal Anti-Inflammatory Drugs reduce inflammation by blocking cyclo-oxygenase enzymes and selective blocking of the COX-2 form reduces gastro-intestinal side effects. [Liu, *et al.*, 2004] employed RBF SVC/SVR to discriminate between COX inhibitors.

Cytochrome p450 (CYP) enzymes are important chemical (and drug substrate) metabolisers within the body, and significant drug inhibition of these is to be avoided. Superior prediction of CYP3A4 inhibition has been reported with SVC compared to other methods [Merkwirth *et al.*, 2004; Arimoto & Gifford, 2005]. SVM methods have also been used to predict CYP2D6, CYP2C9 [Yap & Chen, 2005] and CYP1A2 inhibition [Kless & Eitrich, 2004].

2.2 SVM in Bioinformatics

SVM application in bioinformatics has been reviewed by [Byvatov and Schneider, 2003]. Here we present an update.

Gene Expression Micro-Array Data in the Prediction of Disease Traits. As with SNPs data, input dimensionality can be extremely large (10Ks of genes) whilst the number of examples is relatively small (typically 10s to 100s). Whilst SVMs are relatively well suited to this situation, [Malossini *et al.*, 2004] showed significant performance degradation with just a few incorrectly labelled training examples (as can occur in complex disease diagnosis). Large numbers of correlated and irrelevant genes also diminish performance, making feature selection essential. [Guyon *et al.*, 2002] invented Recursive Feature Elimination (RFE), employing SVC within a wrapper-based approach although [Ambroise and Mclachan, 2002] reported gene selection bias with this. Related 'entropic' [Furlanello *et al.*, 2003] and Recursive Feature Replacement (RFR) [Fujarewicz and Wiench, 2003] followed outperforming earlier methods, with RFR best for smaller gene subsets [Simek *et al.*, 2004].[Fung and Mangasarian, 2004] have achieved sparse models directly with fast linear programming SVC. SVCs are regularly used to predict cancer cases using gene expression training data [Wang *et al.*, 2005], and chemo-genomic studies (of functional relationships between genes and drugs) are also increasing [Bao and Sun, 2002; Thukral *et al.*, 2005].

Receptor Classification and Protein Function Annotation. SVM methods are now often employed to predict the functional classes of proteins from sequence data, i.e. GPCR families or nuclear receptor sub-family [Bhasin and Raghava, 2004a,b] and enzyme class [Dobson and Doig, 2005].

Gene Functional Classes and Annotation. [Brown *et al.*, 2000] first employed SVC to predict functional classes of genes, others have continued in this vein, i.e. [Vinayagam *et al.*, 2004] devised a large-scale gene annotation system exploiting the gene-ontology DAG structure using multiple SVCs for prediction correctness.

Proteomics/Protein Expression. [Jong *et al.*, 2004] Studied predictability of prostate and ovarian cancers using SELDI-TOF mass spectronomy (MS), achieving excellent performance with linear SVC. [Seike *et al.*, 2004] used SVC within a methodology to rank protein spots (in expression profiles from 2D-gel electrophoresis) in terms of their discrimatory ability for human cancers. [Prados *et al.*, 2004] found linear-SVC to out-perform k-NN, ANN and decision tree approaches in predicting ischemic and haemorrhagic stroke from SELDI-MS data applying weight interrogation to identify candidate biomarkers. [Bock and Gough, 2003] used SVC in a system generating protein-protein interaction hypotheses for constructing protein interaction networks.

Other Bioinformatics Applications. [Schneider and Fechner, 2004] have reviewed machine learning approaches (including SVMs) to protein sub-cellular localisation for target identification in drug discovery. There is a growing use of SVC prediction of functionally critical sites within proteins, i.e. sites of: phosphorylation [Kim *et al.*, 2004], ATP-binding [Guo, *et al.*, 2005], catalysis [Dubey *et al.*, 2005] and cleaving [Yang and Chou, 2004]. Specialist kernels have arisen here, i.e. for protein homology [Saigo *et al.*, 2004]) and siRNA design for 'gene-silencing' [Teramoto *et al.*, 2005].

2.3 SVM in Clinical Diagnosis and Epidemiology

Molecular Genetic Epidemiology. Single-Nucleotide Polymorphisms (SNPs) are common individual base changes within human DNA. Millions have been identified. Unlike gene expression measures, SNPs represent unchanging patient-specific variation that may relate to an individuals' prognosis. The feasibility of using SVC methodology to predict disease using multiple SNP variations has been demonstrated for coronary heart disease [Yoon *et al.*, 2003] and breast cancer [Listgarten *et al.*, 2004]. [Barrett, 2005] used SVC to find SNPs associated with drug effect via iterative training and SNP-removal using 1-norm linear SVC weight-vector interrogation.

Epidemiology and Clinical Diagnostics. Apart from in the 'molecular-related' contexts (as above) the use of SVM in epidemiology remains in its infancy. Observing that variable interactions are often not considered in standard univariate analyses, [Fradkin, 2005] discusses the potential of SVM models to provide an alternative to the standard logistic regression method used to identify risk factors in cross-sectional studies. In the only reported study of SVM modelling of large epidemiological observational data, [Muchnik, 2001] used the SEER database, computing multiple SVC models (using variable perturbation) to identify candidate epidemiological factors influencing on breast cancer survival time.[Härdle and Moro, 2004] used SVM to achieve breast cancer survival analysis. [Zhao *et al.*, 2004] used SVC to differentiate anorexic patients. There is a much wider use of SVC in clinical diagnostics with large complex data from sophisticated equipment such as EEG (epilepsy: [Miwakeichi *et al.*, 2001]; CT (colon cancer: [Jerebko, *et al.*, 2005]), MRI (brain glioma: [Li *et al.*, 2005]) and sonography (breast cancer: [Huang & Chen, 2005]).

3 Drug Research Applications of Genetic Programming

In most Pharmaceutical applications, GP evolves predictive models. Typically these take data (i.e. number of positively charged ions, presence of aromatic rings, , etc.) and predict whether a molecule inhibits an enzyme or not. There are now at least two annual workshops on EC uses in Biology: BioGEC (2002-06) and EvoBIO (2003-06).

3.1 GP in Cheminformatics and QSAR.

GP has been used for combinatorial design [Nicolotti et al., 2002], modelling drug bioavailabity [Langdon et al., 2002] and HERG inhibition [Bains et al., 2004], whilst ensembles of ANNs have been evolved to predict p450 inhibition [Langdon et al., 2002a].

3.2 GP in Bioinformatics.

Hot topics include: DNA and protein sequence alignment [Shyu et al., 2004]; protein localisation [Heddad et al., 2004]; using genetic algorithms etc. to infer phylogenetics trees [Congdon and Septor, 2003]; classification and prediction [Hong and Cho, 2004]; recognising transmembrane regions of proteins [Koza and Andre, 1996]; and finding DNA promoters [Howard and Benson, 2003] and gene regulatory sites. Infrared spectroscopy, DNA chip and Single Nucleotide Polymorphisms (SNPs) [Reif et al., 2004] datasets have huge numbers of features. Often the immediate problem is to discover which of the thousands are relevant. In [Johnson et al., 2003] isolation of the relevant wave numbers using GP revealed new insights into commercial crops. GP has also been used to sift thousands of inputs in DNA chip data to discover which genes are important to a metabolic process [Langdon and Buxton, 2004; Moore et al., 2002] or to reduce the number of inputs required so a diagnostic test is practicable [Deutsch, 2003]. While GAs can achieve high multi-class accuracy [Ooi and Tan, 2003] they are also commonly combined with other classifiers, e.g. linear [Smits et al., 2005], SVM [Li et al., 2005], naive Bayes [Ando and Iba, 2004] and k-nearest neighbour. It is no wonder that GP is increasingly being used in Bioinformatics data mining [Kell, 2002]; modelling genetic interactions [Moore and Hahn, 2004] and organisms; inferring metabolic pathways [Koza et al., 2001; Tsai and Wang, 2005] and gene regulatory networks.

3.3 GP in Clinical Diagnosis and Epidemiology Research.

So far, GP is not so used, although GP has been applied to diagnosing pulmonary embolism [Biesheuvel, 2005] and atherosclerosis risk [Sebag et al., 2004].

4 Biological Applications of Particle Swarm Optimisation

Unlike GP, the current use of PSOs in pharmaceutical research is relatively unexplored. Commonly PSOs are used in hybrids with other approaches. PSOs naturally search widely, making them suited to finding good regions. Exploitive local method is then used to refine the good starting points found by PSOs into excellent solutions.

4.1 PSO in Cheminformatics and QSAR.

In QSAR a few teams have used a two stage approach. In the first stage a binary PSO is used to select a few (typically 3-7) features as inputs to supervised learning method. In [Lu *et al.*, 2004] the BPSO selects 7 of 85 features. Then linear models of drug activity (IC50) with two enzymes, COX-1 and COX-2, are constructed. (In [Lin *et al.*, 2005] they use a PSO to divide low dimensional, e.g. 5 features, chemical spaces into pieces. A linear model is fitted to each sub-region.). To aid *in silico* design of drugs, [Lu *et al.*, 2004] produce models which may differentiate between binding to the two enzymes by (virtual) chemicals.

[Wang *et al.*, 2004] and [Shen *et al.*, 2004] use feed-forward ANN to classify the Bio-activity of chemicals using a few (3-6) features selected by a BPSO. They also consider replacing the ANN by a k-nearest neighbour classifier in combination with kernel regression. While they note some differences, many approaches turn out to be equally good at predicting which chemicals will be carcinogenic. The datasets typically only cover a few (31-256) chemicals but, for each one, a large number (27-428) of features are computed from its chemical formula. One can reasonably argue that some form of "feature selection", i.e. choosing which attributes can be used by the ANN, is essential. Even so, given the small number of chemicals involved, [Agrafiotis and Cedeno, 2002; Cedeno and Agrafiotis, 2003; Wang *et al.*, 2004] are still careful to prevent over fitting, e.g. by the use of "leave-n-out" cross-validation.

4.2 PSO in Bioinformatics.

DNA chip experiments often mean under-constrained biomarker search problems (many variables vs few examples). [Xiao et al., 2003] use self organising maps (SOM) to pick clusters of similar genes from datasets with thousands. The PSO is seeded with crude SOM results to refine the clusters.

4.3 PSO in Clinical Diagnosis and Epidemiology Research.

Two and three dimensional medical images, such as X-Rays and MRI, can contain millions of data per subject. [Wachowiak et al., 2004] propose a hybrid PSO to match images taken at different times and/or with different techniques (e.g. ultrasound, CT). Best results came by combining expert medical knowledge to give an initial alignment and a PSO. [Eberhart and Hu, 1999] used a PSO to train an ANN which, using wrist accelerometer data, identifies essential tremor and Parkinson's disease sufferers.

5 Discussion

Whilst the above survey clearly demonstrates a wide coverage of relevant problem areas, it remains unclear as to the underlying extent to which these approaches are actually deployed across the pharmaceuticals industry so their overall importance there is difficult to ascertain. Although becoming less sporadic, it seems that the use of machine learning is still largely driven by individuals either with their own expertise and/or external expert resources.

Machine learning has however proved its worth in many areas for fundamental reasons (for instance model transparency is a recognized benefit of evolutionary methods and SVMs are well known for their generalization). For these newer technologies to make further applications advances there is a need for ease-of-use; easier derivation of problem-specific representations; adequate ways of handling missing data; more widespread generation of reliable prediction confidence measures and attention to statistical power of datasets in model selection. Encouragingly, the machine learning research community is responding to publicised need. Deficiencies in individual methods are being countered by customizations, ensemble and hybrid approaches [Langdon *et al.*, 2003a; Runarsson and Sigurdsson, 2004; Li, *et al.*,2005b; Howley and Madden, 2005; Igel, 2005]. These remain the domain of experts and ease of blending of techniques incorporating multi-objective and constraint-based capabilities is awaited with anticipation.

Acknowledgements

This work was partially funded by EPSRC grant GR/T11234/01. The authors wish to thank GSK colleagues, past and present, for their efforts in expressing the nature of their research.

References

Agrafiotis and Cedeno, 2002. Feature selection for structure-activity correlation using binary particle swarms. Journal of Medicinal Chemistry, 45(5):1098-1107.

Amboise and McLachlan 2002. selection bias in gene extraction on the basis of microarray gene-expression data. PNAS, 99(10):6562-6566

Ando and Iba, 2004. Classification of gene expression profile using combinatory method of evolutionary computation and machine learning. GP&EM, 5(2):145-156.

Arimoto and Gifford, 2005. Development of CYP3A4 Inhibition Models: Comparisons of Machine-Learning Techniques and Molecular Descriptors. Journal of Biomolecular Screening, 10(3):197-205

Bains *et al.*, 2004. HERG binding specificity and binding site structure: Evidence from a fragment-based evolutionary computing SAR study. Progress in Biophysics and Molecular Biology, 86(2):205-233.

Banzhaf, *et al.*, 1998. Genetic Programming An Introduction; On the Automatic Evolution of Computer Programs and its Applications; Morgan Kaufmann.

Bao and Sun, 2002. Identifying genes related to drug anticancer mechanisms using support vector machine. FEBS Lett. 521(1-3):109-14.

Barrett, S.J. (2005) INTErSECT "RoCKET" : Robust Classification and Knowledge Engineering Techniques. Presented at : 'Through Collaboration to Innovation', Centre for Advanced Instrumentation Systems, UCL, 16th February 2005.

Bhasin and Raghava, 2004a. GPCRpred: an SVM-based method for prediction of families and subfamilies of G-protein coupled receptors. Nucleic acids research, 32:W383-W389

Bhasin and Raghava, 2004b. Classification of nuclear receptors based on amino acid composition and dipeptide composition. J. Biological Chemistry, 279(22):23262-23266

Biesheuvel, 2005. Diagnostic Research : improvements in design and analysis. PhD thesis, Universiteit Utrecht, Holland.

Bock and Gough, 2003. Whole-proteome interaction mining. Bioinformatics, 19 (1), 125-135.

Boser *et al.*, 1992. A training algorithm for optimal margin classifiers. 5th Annual ACM Workshop, COLT, 1992

Breiman, 2001. Random forests. Machine Learning, 45:5-32

Breneman 2002. Caco-2 Permeability Modeling: Feature Selection via Sparse Support Vector Machines.Presented at the ADME/Tox symposium at the Orlando ACS meeting,April 2002.

Brown *et al.*, 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl. Acad. Sci., USA 97:262-267

Burbidge *et al.*, 2001a. STAR Sparsity Through Automated Rejection. In Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence: 6th International Work Conference On Artificial and Natural Neural Networks, IWANN 2001, Proceedings, Part 1, Vol. 2084; Mira, J.; Prieto, A., Eds.; Springer: Granada, Spain, 2001.

Burbidge *et al.*, 2001b. Drug design by machine learning: support vector machines for pharmaceutical data analysis. Computers in chemistry, 26(1):4-15

Butte, 2002. The use and analysis of microarray data. Nat. Rev. Drug Discov. 1(12):951-60

Byvatov, and Schneider, 2004. SVM-Based Feature Selection for Characterization of Focused Compound Collections. J. Chem. Inf. Comput. Sci., 44(3): 993-999

Byvatov *et al.*, 2005a. From Virtual to Real Screening for D3 Dopamine Receptor Ligands. ChemBioChem, 6(6):997-999

Cedeno and Agrafiotis, 2003. Using particle swarms for the development of QSAR models based on K-nearest neighbor and kernel regression. J.Comput.-Aided Mol. Des.,17:255-263.

Chen, 2004. Support vector machine in chemistry. World Scientific, ISBN 9812389229

Cheng *et al.*, 2004. Insight into the Bioactivity and Metabolism of Human Glucagon Receptor Antagonists from 3D-QSAR Analyses. QSAR & Combinatorial Science, 23(8): 603-620

Congdon and Septor, 2003. Phylogenetic trees using evolutionary search: Initial progress in extending gaphyl to work with genetic data. CEC, pp320-326.

Cristianini and Shawe-Taylor, 2000. An Introduction to support vector machines and other kernel-based learning methods. Cambridge University Press ISBN: 0 521 78019 5

Deutsch, 2003. Evolutionary algorithms for finding optimal gene sets in microarray prediction. Bioinformatics, 19(1):45-52.

Dobson & Doig 2005.Predicting enzyme class from protein structure without alignments. J.Mol.Biol.,345:187-199

Doniger *et al.*, 2002. Predicting CNS Permeability of Drug Molecules: Comparison of Neural Network and Support Vector Machine Algorithms. J. of Computational Biol., 9(6): 849-864

Dubey *et al.*, 2005. Support vector machines for learning to identify the critical positions of a protein. Journal of Theoretical Biology, 234(3):351-361

Fradkin, 2005. SVM in Analysis of Cross-Sectional Epidemiological Data. http://dimacs. rutgers. edu/SpecialYears/2002_Epid/EpidSeminarSlides/fradkin.pdf

Eberhart and Hu, 1999. Human tremor analysis using particle swarm optimization. In CEC, pp1927-1930

Eberhart, Kennedy and Shi, 2001, Swarm Intelligence, Morgan Kaufmann.

Fujarewicz and Wiench, 2003. Selecting differentially expressed genes for colon tumor classification. Int. J. Appl. Math. Comput. Sci., 13(3):327-335

Fung and Mangasarian, 2004. A Feature Selection Newton Method for Support Vector Machine Classification. Computational Optimization and Applications 28(2):185-202

Furlanello *et al.*, 2003. Entropy-Based Gene Ranking without Selection Bias for the Predictive Classification of Microarray Data. BMC Bioinformatics, 4:54-74.

Guo *et al.*, 2005. A novel statistical ligand-binding site predictor: application to ATP-binding sites. Protein Engng., Design & Selection, 18(2):65-70

Guyon *et al.*, 2002. Gene selection for cancer classification using support vector machines. Machine learning, 46(1-3):389-422

Hand, 1999. Statistics and data mining: intersecting disciplines.*SIGKDD Explorations*, 1: 16-19

Härdle and Moro, 2004. Survival Analysis with Support vector Machines. Talk at Universite Rene Descartes UFR Biomedicale, Paris http://appel.rz.hu-berlin.de/Zope/ise_stat/wiwi/ise/stat/personen/wh/talks/hae_mor_SVM_%20survival040324.pdf

Heddad *et al.*, 2004. Evolving regular expression-based sequence classifiers for protein nuclear localisation.In: Raidl, *et al.*eds.,Applications of Evolutionary Computing,LNCS 3005, 31-40

Hong and Cho, 2004. Lymphoma cancer classification using genetic programming with SNR features. In Keijzer, *et al.* eds., EuroGP, LNCS 3003, 78-88.

Hou and Xu, 2004. Recent development and application of virtual screening in drug discovery: an overview. Current Pharmaceutical Design, 10: 1011-1033

Howard and Benson, 2003. Evolutionary computation method for pattern recognition of cis-acting sites. Biosystems, 72(1-2):19-27.

Howley and Madden, 2005. The Genetic Kernel Support Vector Machine: Description and Evaluation". Artificial Intelligence Review, to appear.

Huang and Chen, 2005. Support vector machines in sonography: Application to decision making in the diagnosis of breast cancer. Clinical Imaging, 29(3):179-184

Igel, 2005. Multiobjective Model Selection for Support Vector Machines. In C. A. Coello Coello, E. Zitzler, and A. Hernandez Aguirre, editors, Proc. of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), LNCS 3410: 534-546

Jerebko, *et al.*, 2005. Support vector machines committee classification method for computer-aided polyp detection in CT colonography. Acad. Radiol., 12(4): 479-486.

Johnson *et al.*, 2003. Metabolic fingerprinting of salt-stressed tomatoes. Phytochemistry, 62(6): 919-928.

Jones, 1999. Genetic and evolutionary algorithms, in: Encyclopedia of Computational Chemistry, Wiley.

Jong *et al.*, 2004. Analysis of Proteomic Pattern Data for Cancer Detection. In Applications of Evolutionary Computing. EvoBIO: Evolutionary Computation and Bioinformatics. Springer, 2004. LNCS, 3005: 41-51

Jorissen and Gilson, 2005. Virtual Screening of Molecular Databases Using a Support Vector Machine. J. Chem. Inf. Model, 45(3): 549-561

Kell, 2002. Defence against the flood. Bioinformatics World, pp16-18.

Kim *et al.*, 2004. Prediction of phosphorylation sites using SVMs. Bioinformatics, 20: 3179-3184.

Kless and Eitrich, 2004. Cytochrome P450 Classification of Drugs with Support Vector Machines Implementing the Nearest Point Algorithm. LNAI, 3303:191-205

Koza, 1992.Genetic Programming: On the Programming of Computers by Means of Natural Selection; MIT Press

Koza *et al.*, 2001. Reverse engineering of metabolic pathways from observed data using genetic programming. Pac. Symp. Biocomp, 2001, 434-435.

Langdon and Barrett, 2004. Genetic programming in data mining for drug discovery. In Ghosh and Jain, eds., Evolutionary Computing in Data Mining, pp211-235. Springer.

Langdon *et al.*, 2001. Genetic programming for combining neural networks for drug discovery. In Roy, *et al.* eds., Soft Computing and Industry Recent Applications, 597-608. Springer. Published 2002.

Langdon *et al.*, 2002. Combining decision trees and neural networks for drug discovery. In Foster, *et al.* eds., EuroGP, LNCS 2278, 60-70.

Langdon *et al.*, 2003a. Comparison of AdaBoost and genetic programming for combining neural networks for drug discovery. In Raidl, *et al.* eds., Applications of Evolutionary Computing, LNCS 2611, pp87-98.

Li *et al.*, 2005. Degree prediction of malignancy in brain glioma using support vector machines. Computers in Biology and Medicine, In Press.

Li *et al.*, 2005b. A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. Genomics, 85(1):16-23.

Lin *et al.*, 2005. Piecewise hypersphere modeling by particle swarm optimization in QSAR studies of bioactivities of chemical compounds. J. Chem. Inf. Model., 45(3):535-541.

Listgarten *et al.*, 2004. Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms. Clin. Cancer Res., 10: 2725-2737

Liu *et al.*, 2004. QSAR and classification models of a novel series of COX-2 selective inhibitors: 1, 5-diarylimidazoles based on support vector machines. Journal of Computer-Aided Molecular Design 18(6): 389-399

Liu *et al.*, 2005. Preclinical *in vitro* screening assays for drug-like properties. Drug Discovery Today: Technologies, 2(2):179-185

Lu *et al.*, 2004. QSAR analysis of cyclooxygenase inhibitor using particle swarm optimization and multiple linear regression. J. Pharm. Biomed. Anal., 35:679-687.

Malossini et al., 2004. Assessment of SVM reliability for microarrays data analysis. In: proc. 2nd European Workshop on data mining and text mining for bioinformatics, Pisa, Italy, Sept. 2004.

Merkwirth *et al.*, 2004. Ensemble Methods for Classification in Cheminformatics. J. Chem. Inf. Comput. Sci., 44(6): 1971-1978

Miwakeichi *et al.*, 2001. A comparison of non-linear non-parametric models for epilepsy data. Computers in Biology and Medicine, 31(1): 41-57

Moore and Hahn, 2004. An improved grammatical evolution strategy for hierarchical petri net modeling of complex genetic systems. In Raidl, *et al.* eds., Applications of Evolutionary Computing, LNCS 3005, pp63-72.

Moore *et al.*, 2002. Symbolic discriminant analysis of microarray data in automimmune disease. Genetic Epidemiology, 23:57-69.

Muchnik, 2004. Influences on Breast Cancer Survival via SVM Classification in the SEER Database. http://dimacs.rutgers.edu/Events/2004/abstracts/muchnik.html

Ng, 2004. Drugs–From Discovery to Approval. Wiley, New Jersey. ISBN: 0-471-60150-0

Nicolotti *et al.*, 2002. Multiob jective optimization in quantitative structure-activity relationships: Deriving accurate and interpretable QSARs. Journal of Medicinal Chemistry, 45(23):5069-5080.

Norinder, 2003. Support vector machine models in drug design: applications to drug transport processes and QSAR using simplex optimisations and variable selection. Neurocomputing, 55(1-2): 337-346

Ooi and Tan, 2003. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics, 19(1):37-44.

Prados *et al.*, 2004. Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents Proteomics, 4(8): 2320-2332

Ratti and Trist, 2001. Continuing evolution of the drug discovery process in the pharmaceutical industry. Pure Appl. Chem.. 73(1):67–75

Reif *et al.*, 2004. Integrated analysis of genetic, genomic, and proteomic data. Expert Review of Proteomics, 1(1):67-75.

Roses, 2002. Genome-based pharmacogenetics and the pharmaceutical industry. Nat. Rev. Drug Discov.1(7):541-9

Runarsson and Sigurdsson, 2004. Asynchronous parallel evolutionary model selection for support vector machines. Neural Information Processing – Lett. & Reviews, 3(3):59-67

Saigo *et al.*, 2004. Protein homology detection using string alignment kernels Bioinformatics, 20: 1682-1689.

Schneider and Fechner, 2004. Advances in the prediction of protein targeting signals Proteomics, 4(6): 1571-1580

Schneider & Fechner, 2005. Computer-based *de novo* design of drug-like molecules. Nat. Rev. Drug Discovery, 4(8):649-663

Schrattenholz,2004. Proteomics: how to control highly dynamic patterns of millions of molecules and interpret changes correctly? Drug Discovery Today: Technologies, 1(1): 1-8

Sebag et al., 2004. ROC-based Evolutionary Learning: Application to Medical Data Mining. Artificial Evolution '03, 384-396 Springer-verlag, LNCS

Seike, *et al.*, 2004. Proteomic signature of human cancer cells. Proteomics, 4(9): 2776-2788

Shawe-Taylor and Cristianini, 2000. An introduction to support vector machines. CUP.

Shen *et al.*, 2004. Hybridized particle swarm algorithm for adaptive structure training of multilayer feed-forward neural network: QSAR studies of bioactivity of organic compounds. Journal of Computational Chemistry, 25:1726-1735.

Shyu *et al.*, 2004. Multiple sequence alignment with evolutionary computation. GP&EM, 5(2):121-144.

Simek *et al.*, 2004. Using SVD and SVM methods for selection, classification, clustering and modeling of DNA microarray data. Engineering Applications of Artificial Intelligence, 17: 417-427

Smits *et al.*, 2005. Variable selection in industrial datasets using pareto genetic programming. In Yu, *et al.* eds., Genetic Programming Theory and Practice III. Kluwer.

Solmajer and Zupan, 2004. Optimisation algorithms and natural computing in drug discovery. Drug Discovery Today: Technologies, 1(3): 247-252

Suwa *et al.*, 2004. GPCR and G-protein Coupling Selectivity Prediction Based on SVM with Physico-Chemical Parameters. GIW 2004 Poster Abstract: P056. http://www.jsbi.org/journal/GIW04/GIW04Poster.html

Takahashi *et al.*, 2005. Identification of Dopamine D1 Receptor Agonists and Antagonists under Existing Noise Compounds by TFS-based ANN and SVM. J. Comput. Chem. Jpn., 4(2): 43–48

Takaoka *et al.*, 2003. Development of a Method for Evaluating Drug-Likeness and Ease of Synthesis Using a Data Set in Which Compounds Are Assigned Scores Based on Chemists' Intuition. J. Chem. Inf. Comput. Sci., 43(4): 1269-1275.

Teramoto *et al.*, 2005. Prediction of siRNA functionality using generalized string kernel and support vector machine. FEBS Lett. 579(13):2878-82

Thukral *et al.*, 2005. Prediction of Nephrotoxicant Action and Identification of Candidate Toxicity-Related Biomarkers. Toxicologic Pathology, 33(3): 343-355

Tobita *et al.*, 2005. A discriminant model constructed by the support vector machine method for HERG potassium channel inhibitors Bioorganic & Medicinal Chemistry Letters, 15:2886-2890

Vinayagam *et al.*, 2004. Appplying support vector machines for gene ontology based gene function prediction. BMC Bioinformatics. 5:116-129

Tsai and Wang, 2005. Evolutionary optimization with data collocation for reverse engineering of biological networks. Bioinformatics, 21(7):1180-1188.

Vapnik, V. N. The Nature of Statistical Learning Theory; Springer: New York, 1995.

Wachowiak *et al.*, 2004. An approach to multimodal biomedical image registration utilizing particle swarm optimization. IEEE Trans on EC, 8(3):289-301.

Wang *et al.*, 2004. Particle swarm optimization and neural network application for QSAR. In HiCOMB.

Wang *et al.*, 2005. Gene selection from microarray data for cancer classification - a machine learning approach. Computational Biology and Chemistry, 29(1): 37-46

Warmuth *et al.*, 2003. Active Learning with Support Vector Machines in the Drug Discovery Process. J. Chem. Inf. Comput. Sci., 43(2): 667-673

Watkins and German, 2002. Metabolomics and biochemical profiling in drug discovery and development. Curr. Opin. Mol. Ther., 4(3): 224-8

Xiao *et al.*, 2003. Gene clustering using self-organizing maps and particle swarm optimization. In HiCOMB

Xu and Hagler 2002. Chemoinformatics and drug discovery. Molecules, 7: 566-600

Xue *et al.*, 2004a. Prediction of P-Glycoprotein Substrates by a Support Vector Machine Approach. J. Chem. Inf. Comput. Sci. 44(4): 1497-1505

Xue, *et al.*, 2004b. QSAR Models for the Prediction of Binding Affinities to Human Serum Albumin Using the Heuristic Method and a Support Vector Machine. J. Chem. Inf. Comput. Sci., 44(5): 1693-1700

Yang and Chou, 2004. Bio-support vector machines for computational proteomics. Bioinformatics, 20: 735 - 741.

Yap and Chen, 2005. Prediction of Cytochrome P450 3A4, 2D6, and 2C9 Inhibitors and Substrates by Using Support Vector Machines. J. Chem. Inf. Model, To appear.

Yap *et al.*, 2004. Prediction of Torsade-Causing Potential of Drugs by Support Vector Machine Approach. Toxicol. Sci., 79: 170-177

Yoon *et al.*, 2003. Analysis of Multiple Single Nucleotide Polymorphisms of Candidate Genes Related to Coronary Heart Disease Susceptibility by Using Support Vector Machines. Clinical Chemistry and Laboratory Medicine, 41(4): 529-534.

Zhao *et al.*, 2004. Diagnosing anorexia based on partial least squares, back-propagation neural network, and support vector machines. J. Chem. Inf. Sci. 44, 2040-2046.

A Review on Design Optimisation and Exploration with Interactive Evolutionary Computation

Alexandra Melike Brintrup, Jeremy Ramsden, Ashutosh Tiwari

School of Industrial and Manufacturing Science, Cranfield University
Cranfield, Bedfordshire, MK43 0AL, UK

Abstract. Interactive Evolutionary Computation (IEC) utilises human-computer interaction as part of system optimisation and therefore constitutes an ideal platform for developing and improving systems that are subjectively influenced. Recently, interest in the usage of IEC for subjectively influenced design practice has shown an increase. In this paper the current state of the utilisation of IEC based optimisation platforms for varying design applications are reviewed. The design fields are categorized by conceptual design, industrial design, and finally, artistic design. We also present problems facing IEC and current research practice to resolve them.

1. Introduction

Subjectivity is an important aspect of many optimisation practices where, it is needed to take advantage of human experience and judgment, to complete a problem definition, to allow flexible problem reformulation, or to represent different views and criteria on design evolution. However, due to its variable and fuzzy nature, it is difficult to model and optimise.

Interactive Evolutionary Computation promotes cooperation between human and computer and optimises target systems based entirely or partially on subjective input from a human user. Qualitative modelling is thereby eliminated and their handling is outsourced to where it comes from: the human system user.

The term IEC includes the computational methods of Interactive Genetic Algorithms (IGA), Interactive Genetic Algorithms (IGP), Interactive Evolutionary Strategies (IES), and Interactive Evolutionary Programming (IEP) under its domain. In this paper, we refer to IEC as the general framework containing either one of these sub categories.

The approach has proved very versatile and has been used in various applications where subjectivity is an inherent part of the target system to be optimised. Some example applications include music [3], graphic art [4], hearing aid fitting [5] game development [6], and more recently, mental health diagnosis [7] and industrial design optimisation processes that typically involve subjectivity [8].

This paper reviews key papers including the usage of IEC in design applications, identifies its advantages and shortcomings when based in a design context. Section 1 looks into conceptual design applications where a single design optimisation objective is pursued, Section 2 reviews systems where IEC has been applied to multi-objective industrial design optimisation. Section 3 briefly looks into artistic design domains where IEC has been utilized, and Section 4 outlines current trends and issues in IEC research.

2. Single Objective IEC for Conceptual Design

The conceptual stage holds the key to successful problem formulation, which influences the following stages of design development and therefore needs to be handled successfully. Successful knowledge accumulation on possible solution space at this stage can result in the softening of strict constraints, shifting of search space into areas previously thought inappropriate with discovery of new solution properties, and inclusion of new design success criteria. As a result problem definition often is observed to shape up during the conceptual design development stage where qualitative features of possible solutions are observed. This nature of cognitive design development brings on the need for (1) easy exploration of the solution space and experimentation with possible changes, (2) the flexible accommodation of qualitative design exploration. Additionally, the conceptual design stage is the stage where innovation is intense. Innovation is a human attribute, where subjectivity and creativity plays an important role and experimentation with different solution scenarios helps the triggering of innovation as different design solution possibilities are explored and problem space is better understood.

In this context IEC can be used to help provide the computer with a means of "understanding" of the humans' subjective opinion on how the solution search is to be lead, while the user is provided a means of how qualitative opinions and solutions space are interrelated. In this sense, the visual interface is used as a communication tool between the user and the computer, and IEC based interfaces enhance this design exploration stage as the impact of user opinion is almost immediately felt upon the next generation of designs. As a result, IEC based platforms are observed to provide ideal tools for conceptual design. This section outlines some key research where IEC has been used solely or partly used for aiding conceptual design.

2.1 Machine Design

Ochi and Hagiwara [9] applied IEC to machine design support. In their system the previous applications of Kotani [10] was improved upon. An initial library of designs was used as a start-off point to accelerate the evolution. Three basic design stages were modelled into the system: the generation of the fundamental machine module, the generation of a functional module, and final stage where these generated parts were combined. It was reported that the proposed systems achieved user satisfactory results in less than half the time previously proposed systems did and that unexpected and innovative results were obtained from the human-computer fusion.

2.2 Image Evolution

Graf and Banzhaf [11] demonstrated how interactive evolution can be applied to 2-D bitmap images. Their system was an early indicator of the usage of artificial evolution for achieving flexibility and complexity in image design with user-input and detailed knowledge. The image structure is made of tie points which could be directly manipulated and reinserted into the design population by the users. Evolution of the images is achieved by the modification of the tie points. The user interaction is a simple one, asking the favourite image(s) to be selected as survivors to the next generation. As the system is built on image manipulation, it was found to be applicable across a wide range of conceptual design fields such as visualizations of automotive and aerial vehicles, and engineering products. The system is useful for conceptual design for the generation of innovative novel design ideas since design experimentation is easy and interesting results can be achieved even with low population sizes and few generations.

2.3 Virtual Modelling System

In Nishino *et al.*'s virtual modelling system [12], a freehand sketch was taken as an input to the system which automatically converted the 2D sketched image into a 3D deformable model. This 3D model then was used by IEC based 3D shape explorer which triggered the evolutionary phase that generated various different shapes by varying the parent shape's geometric parameters. After the completion of the evolutionary phase the finalization of the shape took place using a parametric 3D modeller which provided a user interface to sophisticate the model. It was reported by the experimental subjects that the system was capable of generating shapes that are difficult to create by traditional Computer Aided Design (CAD) tools that normally require the user to master many esoteric commands before starting to work. In this system no special knowledge except simple sketching was necessary. On the other hand, the system's efficiency depended highly on the user's performance to intuitively sketch and consistent evaluation of the evolved shapes as is mostly the case in general IEC based systems. The system was an ideal tool for conceptual design in which approximate shape exploration is what is needed rather than a precise modelling tool that is used to control model parameters tightly for the manufacturing stage.

2.4 Aesthetic Design

As part of the conceptualization stage IEC can be an ideal tool to be used by marketing to gather customer requirements or to aid market segmentation. As most IEC interfaces do not require extensive training for usage, it can be used as the basis for a simple methodology to understand the customer's likes and dislikes and help make the communication with the market more efficient. Yanagisawa and Fukuda [13] have used an enhanced IEC to help a customer set design parameters by simple evaluation of displayed samples. Design attributes to which a user pays more attention were estimated with *reduct* in rough sets theory. New design candidates were then generated by the user's evaluation of design samples. While values of attributes estimated as favoured features are fixed in the refined samples, the others were generated at random. It was reported that the enhanced IEC was better in the generation of user-satisfactory designs than regular IEC.

3. Combining Multi-objective Optimisation with IEC for Industrial Design

Moving from the conceptual design stage to lateral stages, improvement or optimisation on the design takes place. It is here that various qualitative or quantitative design criteria and trade-offs are discussed. Evolutionary Multi-Objective Optimisation (EMOO) platforms play an important role to observe the search space available across the various criteria and negotiate compromise solutions. As EMOO platforms are built on genetic principles, the introduction of IEC into the platform is a simple and efficient way to handle subjectivity. This section reviews some of these combination approaches that were used in industrial design.

3.1 Micro-electrical Mechanical System Design

In Kamalian *et al.*'s [14] Micro-electrical Mechanical System design platform, IEC was used as a single objective optimisation method, after the generation of trade-off solutions with a multi-objective genetic algorithm that optimised designs according to various quantitative objectives. Their system showed that the preferences of subjects were significantly towards designs that were optimised by the IEC enhanced system, as opposed to the designs optimised without IEC. Human evaluation resulted in the embodiment of design expertise to identify potential design or manufacturing flaws which otherwise would be invisible to the evolutionary algorithm.

3.2 Interactive Evolutionary Design System

Cvetkovic and Parmee [15] have developed the Interactive Evolutionary Design System (IEDS), where a rule based preference component allowed the designer to interactively express his/her preferences in terms of natural language. These preferences were used to direct the EMOO search. The IEDS has been developed as a design tool to allow the initial dominance of a cognitive design model that evolves to adopt subjective information, which in following design stages is strictly dominated by a quantitative problem statement. The IEDS made extensive use of IEC to accommodate the inclusion and removal of objectives, changes to their relative importance, changes in upper and lower parameter bounds that define the problem.

The IEDS have been applied to Preliminary gas turbine design and military airframe design, and produced promising results, suggesting that the integration of evolutionary search, problem exploration and optimisation using human-computer interaction could improve the handling of complex conceptual design environments.

3.3 Interactive Multi-objective Optimisation Design Strategy

Tappeta and Renaud [16] developed the interactive Multi-Objective Optimisation Design Strategy (iMOODS), which makes use of preference exexpressions that are user-defined by an interactive process. In the iMOODS method, the decision maker (DM) starts the solution search by specifying an initial "aspiration point". Together with the DM's initial preferences, the Pareto set in the preferred region are generated using projection. The preferred Pareto points and Pareto sensitivity information are then used to create an approximate Pareto surface generation for the current design. The DM can alter his/her preference settings real time. Although the iMOODS accommodates the DM's subjective views in terms of preference handling, it does not take into account objectives that are qualitative in their nature. The strategy was applied to the design of an autonomous hovercraft system and an aircraft concept sizing problem. The authors reported that the strategy is effective and efficient in capturing the DM's preferences and arriving at an optimum design that reflects these preferences.

3.4 Interactive Multi-objective Design Optimisation

Brintrup *et al.* [17] proposed the integrated optimisation of qualitative and quantitative criteria in an interactive multi-objective framework. This method treated qualitative criteria as one of the objectives of EMOO, and gathered its fitness from the user through the IEC module of the framework. The platform was tested by a floor-plan design where two conflicting objectives were negotiated. The experimental subjects reported satisfaction with the framework and the framework proved to be an ideal method to combine criteria that are different in nature.

3.5 Multi-criteria Decision Making Strategy

Hsu and Chen [18] have applied a Learning Classifier System (LCS) to cooperate with an interactive GA approach to aid multi-criteria decision making (MCDM), based on the concept that IGA has unique characteristics to facilitate good MCDM. IGA is particularly suitable for supporting this decision making approach while the design of the GA does not require a fixed fitness function, and makes the search among attributes easier as this is dealt by the computer. The LCS mimicked user ratings and was used to aid user fatigue in the model. The model was tested on the design of a cartoon facial mask. Preliminary results indicated a promising direction in multi-criteria decision making research.

3.6 Interactive Multi-objective Animation Design

Designing human like motions in animation by computer graphics is a difficult task. From a dynamics point of view the problem can be modelled by many quantitative objectives such as the minimization of the joint torques, change of joint torques, acceleration of the handled object and completion time of the motion. On the other hand, as humans have the capacity to evaluate the naturality of the motion, a subjective viewpoint is also needed. Shibuya *et al.* [19] proposed to use a multiple objective genetic algorithm based platform to optimise the quantitative objectives. Later the individuals generated by the algorithm were presented to the user, who picks and ranks preferred individuals. This preference ranking was then used to create the new generation of individuals. The experimentation showed that the approach generated high quality solutions with less stress on users than conventional methods.

4. IEC as the Generator of Artistic Design

As the IEC does not need an explicitly defined fitness function, it is reported that its potential applicability in artistic domains is high. This section briefly looks into the key reports that use IEC in artistic design.

Ventrella [20] has applied IEC to the design of animated characters. Here, the users were asked to identify "amusing behaviour" in the evolved individuals. The usage of interactive techniques in this case has proved not only to be efficient in generating innovative figures but also a task that was observed to include an entertaining element in human-computer interaction.

Lee *et al.* [21] applied IEC to the fashion design domain. The system uses IGA to get a preferable design by user. The IGA used is a simple one that evolves designs based on user scores.

In 3D lighting design system by [22], a designer is given a design motive and is asked to evaluate the lighting of the design from an artistic point of view. The location and intensity of lights sources are the variables which affect the appearance of the design. The study compared the IEC with manual lighting of designers. Designers whose manual lighting was evaluated poor have been able to significantly improve their lighting design capability using the IEC system.

5. IEC as a Design Tool: Trends and Issues

IEC based design systems come in many forms that range from as qualitative enhancers of quantitative multi-objective design optimisation, to conceptual 3D model exploration tools. It has been applied to diverse design domains ranging from machine parts to artistic design domains such as fashion design. This shows the flexibility of evolution with human-computer cooperation. Human subjectivity is a necessary part of design as its accommodation can help mandatory parts of the design process such as innovation, problem definition, alternative solution exploration, incorporation of human experience and intuition. EC provides the necessary design solution exploration and optimisation while the interactive component provides an ideal way to incorporate human subjective opinion into evolutionary synthesis.

One of the remaining problems in IEC that hinder its application as a stand-alone design exploration tool is human fatigue. Human fatigue is the inability of the human designer to assess a large number of designs over a large number of generations due to psychological or physical exhaustion. It is also observed that evaluation of solutions with very minor differences as the convergence reaches to an optimum is another contributor to human fatigue. Other difficulties may arise from the type of application. For example motion or music must be evaluated as a whole and memory limitations of the human makes this task difficult.

Rating scales can be problematic too. Due to the subjectivity involved, we cannot guarantee a mathematically consistent rating that will impact in proportion or accordance with the changes in the qualitative space. The reasons for this might be simply due to user inconsistency or human fatigue such as user not remembering older designs for an accurate comparison or provide a rating that can represent this comparison.

The evaluation might differ when the user compares designs with the whole range of designs seen until current generation, or designs within one population only. In the former case, convergence is more likely to occur but as the number of generations increase, it gets more difficult for the user to compare.

Current non-applicational IEC research focuses on these problems and various proposals to combat human fatigue, and problems due to inconsistency or rating perspectives. Te usage of neural networks [1] or other prediction mechanisms to mimic user evaluation after a number of generations of user evaluation, combining regular EC and IEC to get user guidance only in some generations [14], outsourcing quantitative design objectives to a regular EC and subjective design objectives to IEC thereby reducing the evaluation numbers, active user intervention [24] for human fatigue; scaling user ratings to absolute [1] for scaling problems; providing better user interfaces [23] for inconsistency problems are discussed in the IEC research community to combat the problems of IEC and enable it to be used as an efficient tool for design applications.

In its pure alone form, IEC is an excellent tool for innovative conceptual design generation. As part of other optimisation systems or soft computing packages IEC can also be applied as a knowledge input tool for lateral stages of design improvement. Combination of IEC with EMOO [2], and other soft computing techniques such as Fuzzy Systems and Agent Based Systems [4] show promising research directions.

References

[1] S. Wang, H. Takagi, "Evaluation of User Fatigue Reduction Through IEC Rating-Scale Mapping", 4th IEEE Int. Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST2005), Springer-Verlag, pp. 672-681, May 2005, Hokkaido, Japan

[2] A. Brintrup, A. Tiwari , J. Gao, "Handling Qualitativeness in Evolutionary Multiple Objective Engineering Design Optimisation", Proc. of International Conference of Computational Intelligence, December 2004, Istanbul, Turkey

[3] N. Tokui, H. Iba, "Music Composition with Interactive Evolutionary Computation", 3rd International conference on generative art (GA2000), pp. 215-226, December 2000, Milan, Italy

[4] T. Unemi, "SBART 2.4:An IEC tool for creating 2D images, movies and collage", Workshop on GA in Visual Art and Music, July 2000, pp.21-23, Las Vegas, USA

[5] H. Takagi, M. Ohsaki, "IEC fitting: New framework of hearing aid fitting based on computational intelligence technology and user's preference for hearing", Poster Session PB9, International Hearing Aid Research Conference (IHCON 2000), pp. 49-50, August 2000, Lake Tahoe, CA, USA

[6] L. Pagliarini, A. Dolan, F. Menczer, H. H. Lund, "ALife meets web: Lessons Learned", 1st International Conference on Virtual World (VW 98), pp. 156-167, Springer Verlag, July 1998, Berlin, Germany

[7] H. Takagi, T. Takahashi, K. Aoki, "Applicability of interactive evolutionary computation to mental health measurement," IEEE International Conference on Systems, Man, and Cybernetics, Hague, Netherlands

[8] A. Brintrup, J. Ramsden, A. Tiwari , "Integrated Qualitativeness in Design by Multi-Objective Optimisation and Interactive Evolutionary Computation", Accepted for publication: Proceedings of Congress of Evolutionary Computing, September 2005, Edinburgh, Scotland

[9] T. Ochi, M. Hagiwara, "Machine Design Support System using Interactive Evolutionary Techniques", IEEE International Conference on Systems, Man and Cybernetics, pp.1075-1080, October 2001

[10] [J. Kotani, M. Hagiwara, "An evolutionary design support system with structural representation", IEEE Conference on Industrial Electronics, Control and Instrumentation, 2000

[11] J. Graf, W. Banzhaf, "Interactive Evolution for simulated natural evolution, artificial evolution", European Conference AE'95, Selected Papers, (eds. J. M. Alliot, E. Lutton, E. Ronald, M. Shoenauer), pp. 259-272, September 1995, Berlin, Germany

[12] H. Nishino, H. Takagi, S. Saga, K. Utsumiya, "Virtual modelling system for intuitive 3D shape conceptualization", IEEE International Conference on Systems, Man, and Cybernetics (SMC2002), , vol.4, pp. 541-546, October 2002, Hammamet, Tunisia

[13] H. Yanagisawa, S. Fukuda, "Interactive Reduct Evolutional Computation", Proceedings of DETC'03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference September 2003, Chicago, Illinois USA

[14] R. Kamalian, H. Takagi, A. M. Agogino, "Optimised Design of MEMS by Evolutionary Multi-objective Optimisation with Interactive Evolutionary Computation" Genetic and Evolutionary Computation (GECCO2004), pp. 1030-1041, June 2004, Seattle, WA

[15] D. Cvetkovic, I. C. Parmee, , "Agent-based Support within an Interactive Evolutionary Design System", Artificial Intelligence for Engineering Design, Analysis and Manufacturing Journal; Cambridge University Press, Vol.16 No.5, 2002

[16] R. Tappeta, J. Renaud, "Interactive Multiobjective Optimisation Procedure", J. AIAA, No. 7, Vol. 37, pp. 881-889

[17] A. Brintrup, A. Tiwari , J. Gao, "A Proposed Framework for Handling Qualitative and Quantitative Data in an Evolutionary Multiple Objective Design Space", Accepted for publication: Journal of Computational Intelligence

[18] F. C. Hsu, J. Chen, "A study on multi criteria decision making model: interactive genetic algorithms approach", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, 1999, pp. 634-639

[19] M. Shibuya, H. Kita, S. Kobayashi, "Integration of multi-objective and interactive genetic algorithms and its application to animation design", IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028), 1999, vol.3, pp. 646-51

[20] J. Ventrella, "Disney meets Darwin", Conference of Computer Animation '95, 1995, pp. 35-43

[21] J. Lee, H. Kim, S. Cho, "Accelerating Evolution by Direct Manipulation for Interactive Fashion Design", Proceedings 4th International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2001, pp. 343-7

[22] K. Aoki, H. Takagi, "3D CG Lighting with an Interactive GA", 1st International Conference on Conventional and Knowledge-based Intelligent Electronic Systems (KES'97), pp.296-301, May 1997, Adelaide, Australia

[23] Y. Todoroki, H. Takagi, "User Interface of an interactive evolutionary computation for speech processing", 6th International Conference On Soft Computing (IIZUKA2000), pp. 112-118, October 2000, Fukuoka, Japan

[24] H. Takagi, "Active User Intervention in an EC Search", 5th Joint Conference On Information Sciences (JCIS2000), pp. 995-998, February 2000, Atlantic City, New Jersey, USA

Part IV

Pattern Recognition

Mapping of Natural Patterns by Liquid Architectures Implementing Neural Cliques

Karina Odinaev, Igal Raichelgauz, and Yehoshua Y. Zeevi

Department of Bio-Medical Engineering, Department of Electrical Engineering,
Technion – Israel Institute of Technology, Haifa 32000, Israel
{karinao@tx.technion.ac.il, igal.raichelgauz@intel.com, zeevi@ee.technion.ac.il }

Abstract. Computational tasks related to processing and recognition of natural signal require identification of complex patterns and relationships in massive quantities of low precision, ambiguous noisy data. While state-of-the-art techniques and architectures fail to provide sufficient solutions, cortical neural networks have an inherent computational power in this domain. A recently-introduced Liquid-State-Machine (LSM) paradigm provides a computational framework for applying a model of cortical neural microcircuit as a core computational unit in classification and recognition tasks of real-time temporal data. In this study we apply the concept of "Neural Cliques" and extend the computational power of the LSM framework by closing the loop. By incorporating functions of readout, reward and feedback, we implement such a closed-loop framework of neural architecture in classification and recognition tasks of real-time temporal data. This approach is inspired by several neurobiological findings from ex-vivo multi-cellular electrical recordings and injection of dopamine to the neural culture. Finally, we illustrate the performance of the proposed architecture in word-recognition tasks.

1 Introduction

The lack of adequate interface between the natural environment and the computing devices constitutes a significant barrier to computer application in many real-world tasks. To incorporate computational means in the execution of day-to-day tasks, the physical world must be instrumental in the process, so that the computer systems will be exposed to, and linked with, the natural environment. The latter involves the transformation of data across the boundary between the real and the digital world, whenever a computer is sampling and/or acting on real world data. Examples of these "boundary transformation" problems include the computer recognition of human speech, computer vision, textual and image content recognition, robot control, OCR, ATR, and more. These are difficult problems to solve on conventional computers, since they require the computer to find complex structures and relationships in massive quantities of low precision, ambiguous and noisy data.

It has been proposed and supported by empirical evidences [5] that large, generic, random, massively connected cortical networks are not built specifically for each computational task but, rather, are used as a basic computational unit for diverse natural computational tasks in different cortical areas. Therefore, realistic models of these networks are good candidates for a core of biologically-motivated computational architectures. Moreover, even a relatively simple model composed of ~100 sparsely connected leaky-integrate-and-fire (LIF) neurons by dynamic synapses, with stochastic heterogeneous parameters, depicts powerful computational capabilities in a domain of parallel processing of temporal noisy data in real-time.

A new computational paradigm, called Liquid-State-Machine (LSM), recently introduced by Maass, Natschläger and Markram [1], provides a theoretical basis for applying a model of neural microcircuit to generic computational tasks. The LSM system is composed of two parts: *(1)* A liquid computational unit– a model of neural microcircuit is used as a "reservoir" of complex dynamics to transform the input time series *u(.)* into "liquid states" *x(t)*, and to *(2)* A readout – a memoryless function which maps the liquid state *x(t)* at time *t* onto the output *v(t)*. Readout may be implemented by a simple one-layer network of perceptron, trained by linear algorithm to build a function mapping liquid-states onto desired outputs. It was shown by means of simulations [2] that such a system is computationally effective in executing parallel tasks of recognition and classification of temporal data.

In the framework of computational LSM, a neural microcircuit is used as an efficient generic filter transforming different temporal inputs into significantly different liquid states. The task-dependent part is executed by the readout after being trained by supervised-learning algorithm to map these states onto predefined output. Obviously, neural systems are not composed of these two different components – liquid-states generators and readout layers. Thus the functions of both, Readout and Liquid should be incorporated into co-sets of the same generic neural ensemble. However, by this simplification and by emphasizing that recurrent neural ensemble, rather than individual neurons, should be viewed as basic computational units, the LSM computational framework suggests a radically different paradigm for neural computation. Moreover, the LSM framework enables the application of real cortical neural ensembles in real-world tasks by embodiment of cortical neural culture in artificial environments [15].

In this study we extend such a non-Turing paradigm for neural computations by incorporating biologically-motivated computational functions and components, such as reward-based feedback, observed in experiments performed on ex-vivo neural culture [10]. Several principles of computational neurobiology are assumed: (1) Computational tasks are carried out by spatio-temporal patterns, coined "Cliques", generated by generic neural ensembles which are vastly mutually communicated [16]. (2) Learning processes drive the generation of new subsets of cliques dictated by the environment through reward and/or penalizing signals. Reward signals are sent through a feedback from the environment and allow the success in computational tasks.

2 Neural Microcircuit as a Generic Computational Unit

The neocortex is characterized by precise structure of columns and layers. Within neocortical layers neurons are mapped into each other, where anatomical and physiological properties are unique for each type of pre- and post-synaptic combination. However remarkable morphological, electrophysiological and spatial stereotypy exists in these networks, in addition to very stereotypical connectivity and patterning of synaptic connections between neighboring cells. This clear stereotypy exists across different regions of the brain, suggesting that there is a generic template of microcircuit and that all neocortical microcircuits are merely subtle variations of that common microcircuit template. Such templates could subserve the apparent omnipotent functional capacity of the neocortical microcircuitry [5]. A computational model of generic neural microcircuit is inherently endowed with powerful and versatile information processing capabilities. We used a similar model to [2], composed of a 3-dimentional recurrent network of 150 Leaky-Integrate-and-Fire (LIF) neurons with random connectivity, and similarity to generic cortical microcircuit, 20% of the neurons are randomly chosen to be inhibitory and, accordingly, 80% excitatory. The probability of connection between two neurons depends on the distance between them according to,

$$C \cdot \exp(-D(i,j)/\lambda^2), \tag{2.1}$$

Where in λ and C are parameters that determine the average number of connections for a certain Euclidean distance D between the neuron i and neuron j. This connectivity characterization by primary local connections and a few longer connections is biologically realistic. Long range connections will be incorporated, and their functional effects on the computational properties of the network will be investigated within a context of a different study.

Random, heterogeneous parameters of NM model fit neurobiological data from rat somatosensory cortex [2]. Synaptic short-term plasticity of the NM is implemented by dynamic synapses in which the amplitude of each post-synaptic-current depends on the spike train that is impinging on the synapse [6], and causes facilitation and depression processes. The model was implemented using CSIM simulator [7].

3 Computing with Neural Cliques

Spatio-temporal firing patterns may be considered as basic information units of neural ensemble's response; however it seems that there is no unique information encoded in the dynamics characteristics of these firing patterns, but rather in the timing and specificity of the firing/non-firing neurons [17]. These spatio-temporal patterns are sensitive to input signals and are indicative of network states. "Synfire-chains", a concept originally introduced by Abeles [18], [11], and recently confirmed in neural culture activity by Yuste and associates [12], [19], emphasize the importance of correlated spatio-temporal firing patterns generated by neural ensembles with relevance to their connectivity characteristics. For this reason, we introduce and define the concept spatio-temporal patterns produced by neural ensembles' "Cliques". A clique does not directly depend on the connectivity characteristics of the firing/non-firing neurons and therefore may be composed of several synfirechains active in the same or even different brain loci and structures. Figure 1 illustrates the meaning of a clique by means of a spatio-temporal cube with finite temporal length, determined by short-term dynamics of the neural ensemble. The selection of subsets of neurons participating in the "clique" is determined by subsequent neuronal layers through closed-loop interaction with the environment and, thus, is a function of the defined computational task.

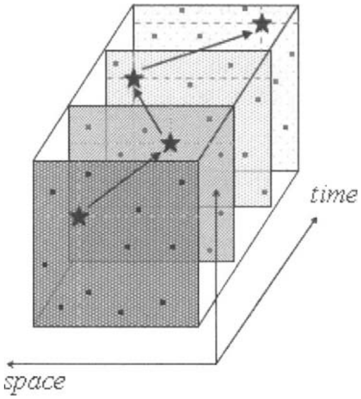


Fig. 1. Spatio-temporal representation of neural clique depicted as a sliding cube with finite temporal length. Points indicate neural responses that do not participate in the clique, stars indicate neural responses that participate in the clique.

Learning process drives a neural microcircuit to the desired cliques defined by configuration of sets of associations between stimuli and responses. This dynamical process begins with exploration of various network's cliques through modification of neuronal correlations. Two mechanisms which may be responsible for changing neuronal correlations are driving stimuli and neuromodulation by dopamine. Experiments on ex-vivo culture have shown [8], [4] that both mechanisms enhance changes in neuronal correlations by dispersing existing correlations, i.e. decorrelating previously acquired correlated activity. It is assumed that both mechanisms that cause decorrelation (dispersion) are mediated by a biophysical jittering of the synaptic strengths at polysynaptic level.

The second phase of learning, the recognition, is responsible for "freezing" the NM state by stopping the exploration process after the desired cliques were obtained. In recent years, a major effort has been devoted to mapping of the behavioral concept of reward to neural mechanisms that change the functionality of a given NM based on its past performance [9]. The regulation of exploration process, driven by dopamine neuromodulation, is enabled by reward prediction error (RPE) signals. Learning by reward can occur by associating a stimulus or an action with a reward [3]. In this type of learning known as "Learning by Dispersion" [4], [14], the synaptic efficacies are jittered according to the RPE values, i.e. the higher the error in the computational task, the larger the amplitude of jittering. The process continues until the error converges to zero and the system "freezes". In other words, the mechanism of jittering the synaptic efficacies, discovered by Eytan and Marom, is instrumental in avoiding trapping into a fixed point. When the best clique dictated by the environment is found, the system reaches the recognition phase, and by stopping the dopamine emission, network's associations are "frozen".

A mathematical model of this process, in which the synaptic efficacies are randomly jittered by regulation of RPE is formulated by:

$$\Delta W = \psi(W_0 \cdot K \cdot RPE), \tag{3.1}$$

wherein ψ is uniform distribution in the range of the argument, *W0* is the previous value of the synaptic strength, *K* is a constant, and *ΔW* is the change in the strength of the synapse. The model illustrates exploration and recognition processes, by dispersion of the NM synaptic strengths, regulated by the success in achieving the task of the overall system.

The overall framework is described in Fig. 2. Time-varying stimuli from the environment excite NM with a continuous input stream $(P_i(t))$. At any time t_0, the clique of the microcircuit $(C_i(t_0))$ holds a substantial amount of information about recent inputs $P_i(t < t_0)$. Memoryless function maps the cliques $C_i(t_0)$ onto discrete predefined values (*j*). Discrete value *j* is a decision/action of the system in its environment.

If the system succeeds in the task, i.e. $i = j$ for classification task, reward signal is sent by the environment to the system. Reward signals, injected by the environment, are determined by system's performance and activate the Decorrelator by setting the value of RPE. Decorrelation mechanism modifies the NM synaptic strengths according to previously defined algorithm and drives the exploration phase of learning. When system's performance is sufficient, RPE is low, the recognition phase is reached and NM state is “frozen” by stopping the dispersion of the synaptic strengths.

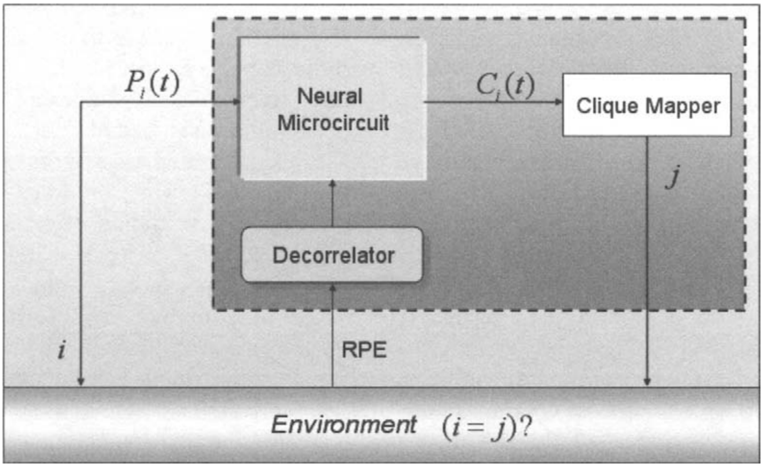


Fig. 2. Closed-loop liquid architecture implemented in a classification task of time-varying inputs. NM is composed of 135 LIF neurons. Time-varying stimuli *Pi(t)* are transformed by NM onto cliques, $C_i(t_0)$, defined as firing patterns of NM at time t_0. Memoryless function maps the cliques $C_i(t_0)$ onto discrete output to the environment (*j*). A feedback on system's performance is sent by the environment in form of reward signals to determine the RPE. Decorrelation, regulated by RPE, enables the exploration process of the NM until a desired performance is obtained.

4 Word Recognition Task

Closed-loop liquid architecture was applied in a well-studied computational benchmark task – an isolated word recognition task. The dataset consists of 230 input files: 40 samples of the word “one”, 40 samples of the word “zero” by different speakers, and 150 samples of other words by 10 different speakers. The task was to recognize the word “one” out of 190 other words. To verify that the system can be extended to recognition of words other than “one”, similar experiment was done for recognition of the word “zero”.

The waveforms of the input sound were preprocessed by performing Fourier transform. Each of the frequency bands was composed of one or more of the following three events: onset (the start of the phase of significant energy), offset (the end of this phase), and peak (the first maximum of energy). The entire waveform is normalized to have maximum amplitude of 0.7, the sampling rate used in this case is 12000 samples/sec. The running average power and its second derivative are subsequently used in identification of events in the sound's spectrogram. This sound preprocessing converts the sound signal into a spatiotemporal sequence of events, suitable for recognition. Monosyllabic words are encoded into such sequences by retrieving features in different frequency bands in their spectrogram. Finally, sound waveform is converted into a list of 40 single events that are converted in turn into their respective times of occurrence [13].

For recognition of the word "one" 20 signals were randomly chosen out of 40 as a training data and the other 20 as a testing data, in addition to other 150 samples of different words. A previously described, randomly generated NM was implemented in a in a closed-loop setup. The mapping function from the NM cliques onto the output of the system was implemented by a simple algorithm identifying the stable neuronal spikes within the spatio-temporal firing pattern of the NM, after each step of exploration process. The average error in this classification task, achieved by this closed-loop system, was 0.065, as depicted in Fig. 3.

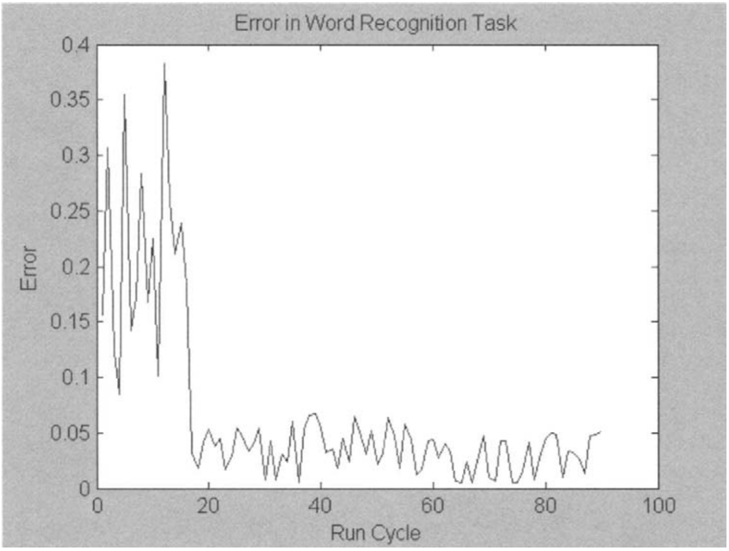


Fig. 3. Error-in-task of closed-loop liquid architecture vs. time.

5 Discussion

The liquid architecture paradigm, motivated by the cortical NM model, enhances the computational and learning capabilities characteristic of neural networks. On the one hand, the liquid architecture paradigm depicts a rich behavior that can be tested experimentally on the ex-vivo tissue setup. These emerging architectures are motivated by neurobiological findings obtained in experiments with neural culture [4], [8], [10]. On the other hand, the liquid architecture paradigm inspires the development of new computational hardware, suitable for processing and mapping of natural signals. We selected speech as an example of physiologically generated natural signal, and demonstrated that indeed such architecture is suitable for accomplishing relatively complicated processing tasks. Lastly, and most interestingly, is the possibility of coupling a real NM tissue and a liquid architecture. Such a hybrid computational system is likely to provide further insight into the rules that govern the functioning of brain tissue and its computational capacity. Such a hybrid system can also serve as a closed-loop model of a brain interaction with its environment.

References

1. W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation, 14(11):2531-2560 (2002).
2. W. Maass, T. Natschläger, and H. Markram. Computational models for generic cortical microcircuits. In J. Feng, editor, Computational Neuroscience: A Comprehensive Approach, chapter 18, pages 575-605. Chapman & Hall/CRC, Boca Raton, (2004)
3. Schultz W.: Neural coding of basic reward terms of animal learning theory, game theory, microeconomics and behavioural ecology. Curr Opin Neurobiol. Apr;14(2):139-47 (2004)
4. D. Eytan, A. Minerbi, N. Ziv and S. Marom. Dopamine-induced Dispersion of Correlations Between Action Potentials in Networks of Cortical Neurons. J. Neurophysiol. 92:1817-1824 (2004)
5. G. Silberberg, A. Gupta and H. Markram: Stereotypy in neocortical microcircuits, Trends Neurosci. May;25(5):227-30 (2002)
6. M Tsodyks, K. Pawelzik, H. Markram , Neural networks with dynamic synapses. Neural Computation 10, 821-835 (1998)
7. T. Natschläger, H. Markram, and W. Maass. Computer models and analysis tools for neural microcircuits. In R. Kötter, editor, *A Practical Guide to Neuroscience Databases and Associated Tools*, chapter 9. Kluver Academic Publishers (Boston), 2002. in press. (http://www.lsm.tugraz.at)
8. Goded Shahaf and Shimon Marom. Learning in networks of cortical neurons. J. of Neuroscience [volume 21(22):8782-8788, Nov. 15 (2001)
9. Schultz W. Predictive reward signal of dopamine neurons. J Neurophysiol, 80:1-27 (1998)
10. Danny Eytan and Shimon Marom. Learning in Ex-Vivo Developing Networks of Cortical Neurons Progress in Brain Research, Volume 147, "Development, Dynamics and Pathology of neural Networks", van Pelt et al, Editors, (2004)
11. Aviel Y., Pavlov E., Abeles M., and Horn D. Synfire chain in a balanced network. Neurocomp. 44:285-292 (2002)
12. Ikegaya Y., Aaron G., Cossart R., Aronov D., Lampl I., Ferster D., Yuste R., Synfire Chains and Cortical Songs: Temporal Modules of Cortical Activity. Science. 304 (5670): 559-564, (2004)
13. Prashant Joshi. Synthesis of a liquid machine with Hopefield/Brody transient synchrony. M.SC Thesis, (2002)
14. I. Raichelgauz, K. Odinaev and Y.Y. Zeevi, "Closed-Loop Liquid Neural Architectures", Neurocomputing (Invited Paper), 2005 (in press).
15. DeMarse T., Cadotte A., Douglas P., He P., Trinh V., Computation Within Cultured Neural Networks, Proceedings of the 26th Annual International Conference of the IEEE EMBS, (2004)

16. Unger H., Zeevi Y. Y., Blind Separation of Spatio-temporal Data Sources. ICA 2004: 962-969 (2004)
17. M. Abeles, I. Gat, Detecting precise firing sequences in experimental data, Journal of Neuroscience Methods, Vol. 107, Issues 1-2, pp. 141-154 (2001)
18. Abeles M., Neural Circuits of the Cerebral Cortex, Cambridge University Press (1991)
19. Yuste, R., Lanni, F. and Konnerth, A., Imaging Neurons: a Laboratory Manual, Cold Spring Harbor Press, (1999)

Pattern Recognition Using Modular Neural Networks and Fuzzy Integral as Method for Response Integration

Patricia Melin, Gabriela Martinez, Claudia Gonzalez, Diana Bravo, and Felma Gonzalez

Dept. Computer Science, 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.

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.

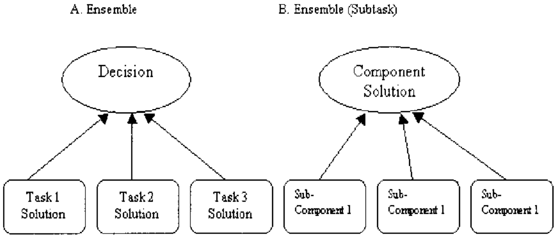


Fig. 1. Ensembles for one task and subtask.

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.

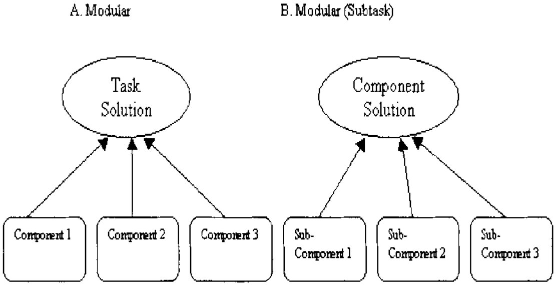


Fig. 2. Modular approach for task and subtask.

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.

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.

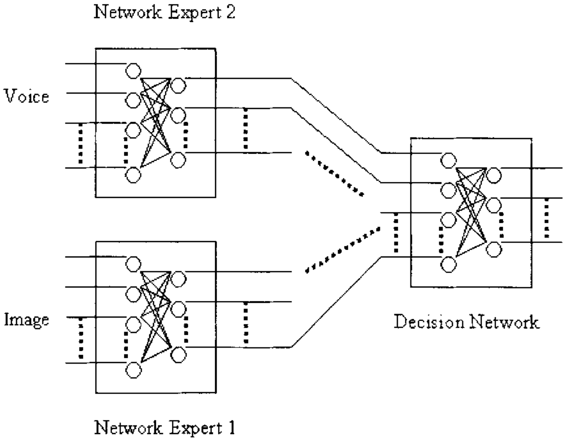


Fig. 3. Multiple networks for voice and image.

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

$$\lim_{i\rightarrow\infty} g(A_i) = g(\lim_{i\rightarrow\infty} A_i) \qquad (1)$$

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 $A \cap B = \theta$,

$$g(A \cup B) = g(A) + g(B) + \lambda\, g(A)g(B), \qquad (2)$$

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 [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 $h:X\rightarrow[0,1]$ 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,

$$h(x) \circ g(x) = \max_{E \subseteq X} \left[\min \left(\min_{x \in E} h(x), g(E) \right) \right] = \sup_{\alpha \in [0,1]} \left[\min(\alpha , g(h_\alpha)) \right] \tag{3}$$

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%.

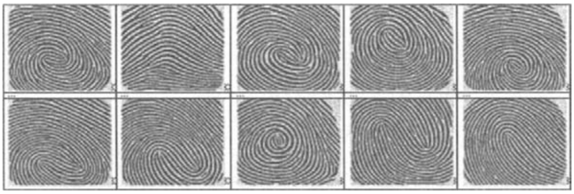


Fig. 4. Sample Images used for Training.

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.

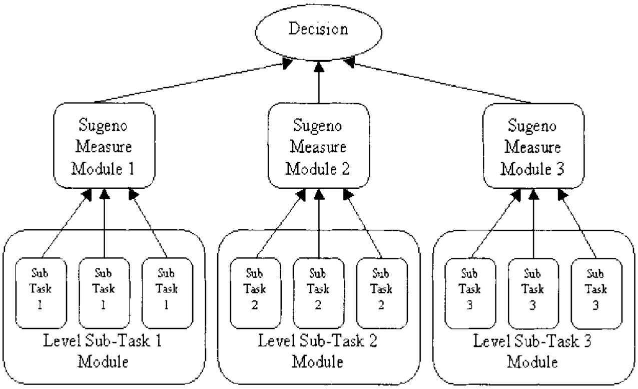


Fig. 5. Final Proposed Architecture.

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.

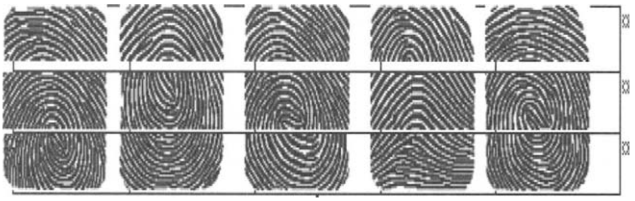


Fig. 6. Example of Image Division.

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.

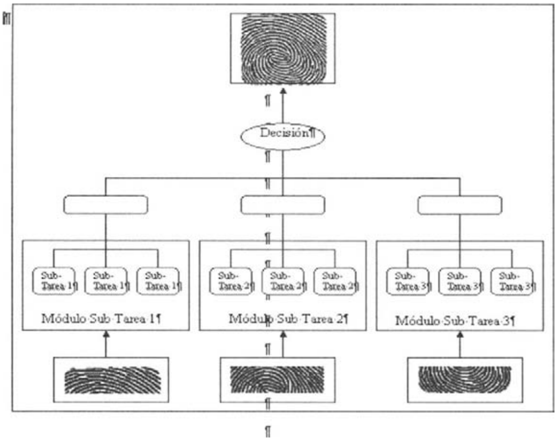


Fig. 7. Final Architecture Showing Inputs and Outputs.

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,

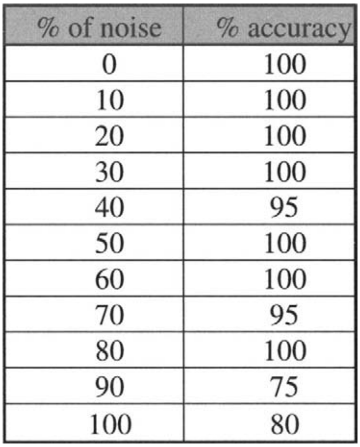
$$g(M_i) = h(A) + h(B) + \lambda\, h(A)\, h(B) \tag{5}$$

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



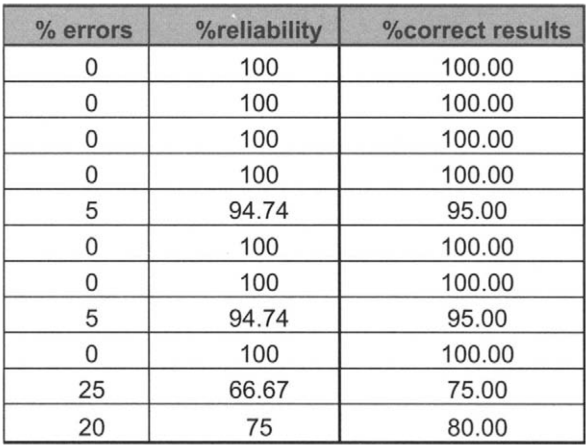
% of noise	% accuracy
0	100
10	100
20	100
30	100
40	95
50	100
60	100
70	95
80	100
90	75
100	80

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.

Genetic Algorithm-Evolved Bayesian Network Classifier for Medical Applications

Matthew Wiggins[1a], Ashraf Saad[1], Brian Litt[2] and George Vachtsevanos[1]

[1]School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia, USA.
[2]Departments of Neurology and Bioengineering,
University of Pennsylvania, Philadelphia, Pennsylvania, USA.

Abstract. This paper presents the development of a Bayesian Network (BN) classifier for a medical application. Patient age classification is based on statistical features extracted from electrocardiogram (ECG) signals. The computed ECG features are converted to a discrete form to lower the dimensionality of the signal and to allow for conditional probabilities to be calculated for the BN. Two methods of network discovery from data were developed and compared: a greedy hill-climb search and a search method based on evolutionary computing. The performance comparison of these two methods for network structure discovery shows a large increase in classification accuracy with the GA-evolved BN as measured by the area under the curve of the Receiver Operating Characteristic curve.

Keywords: bayesian networks, evolutionary computing, genetic algorithms, hybrid soft computing techniques, evolved bayesian network classifier.

1 Introduction

The human heart is a complex system that gives many clues about its stability in its electrocardiogram (ECG) signal. Many of these clues are difficult to discern due to the multitude of characteristics of the signal. In order to compress these characteristics into more comprehensible components, researchers have developed numerical quantifications that reflect certain signal behaviors. Though there are many types of quantification, here referred to as signal features, it is often difficult to combine and assess them

[a] Correspondence email for first author: gte986h@mail.gatech.edu.

in a meaningful and useful way. A Bayesian Network (BN) is a relatively new way of taking the data provided and using probabilistic correlations to make predictions or assessments of class membership. The difficulty with this type of classifier is that it needs large amounts of data to determine the probabilities that populate the network. It is also difficult to formulate a reliable method to develop the structure of the network once the data is obtained. One network structure discovery method is a greedy algorithm that connects a new node to a node of interest only if an overall behavioral improvement is gained. This greedy addition of nodes can result in the algorithm arriving at a local maximum; therefore, some perturbation must be introduced to break from the local maximum to find the global optimum [1]. The construction of Bayesian Networks using a Genetic Algorithm (GA) instills randomness through the mutation and crossover operators it uses to evolve the network structure. This allows intelligent model construction without requiring an exhaustive search on all possible structure combinations of nodes. This paper demonstrates the applicability of evolving a BN classifier to distinguish between two groups based on ECG features derived from 21 to 43 year old and 68-85 year old healthy adults, hereby referred to as young and old patient groups, respectively. If this method can distinguish between the two patient groups, a more complex classification problem, such as cardiac disease risk stratification, might be attempted, yielding better accuracy than traditional methods for fusion of multiple clinical measurements.

2 Signal Processing

ECG signals from 20 young and 20 old patients were downloaded from the *Fantasia* database (available on www.physionet.org [2]). Figure 1 depicts the block diagram of the signal processing that was performed. Figure 2 shows a typical ECG signal prior to preprocessing (top) and after (below). The signals were then used to calculate the feature set, *F*, comprising the following 12 feature measures, namely: energy, nonlinear energy, 4th power, peak power, curve length, Hurst parameter, peak frequency, mean frequency, median frequency, spectral entropy, Katz fractal dimension, and Shannon entropy. The decomposition of the large ECG signals into feature values reduces the dimensionality of the problem to a computationally tractable level. In this process, signal information is encoded so that it can be used for classification and prediction.

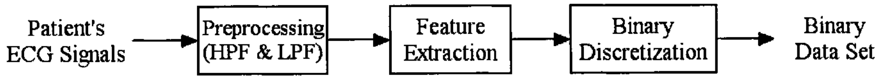


Fig. 1. Feature Extraction System Diagram

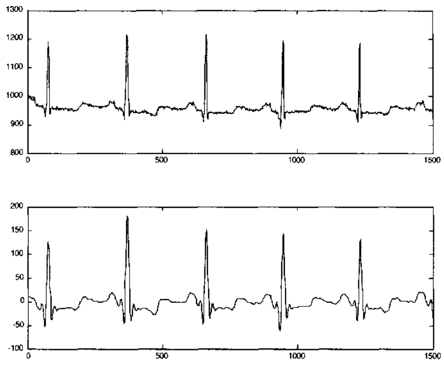


Fig. 2. Original ECG signal (top) and ECG signal (bottom) after preprocessing by low-pass and high-pass filters.

For example, the energy of the signal can show the signal's tendency to avoid the baseline; frequency based features reveal other characteristics of the signal such as the region of maximal power density; fractal dimension gives a measure of the self-similarity and complexity of the signal. Feature values were discretized into binary form based on their value being above or below a certain threshold. This threshold was set using a Receiver Operating Characteristic (ROC) curve where a feature value is predictive of the variable of interest, in this case, *age*.

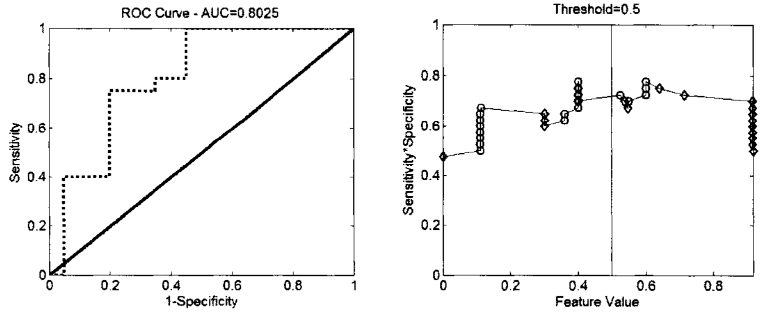


Fig. 3. ROC curve (left) created to determine the classification threshold set as the maximum of the product of the sensitivity and specificity plot (right). The diamonds represent old while the circles represent young patients. The vertical line is at the feature value of the threshold.

A location on the feature value continuum slightly greater than the maximum of the sensitivity times the specificity is used as the threshold. This is depicted in Figure 3, showing the ROC on the left and the product of sensitivity and specificity on the right. The diamonds represent old patients and circles represent young patients, while the vertical line shows the chosen threshold for binary discretization. The technique described above culminates in class and feature information in the form of binary numbers. This allows for easy computation of conditional probability tables for the Bayesian Network.

3 Bayesian Networks

A conditional probability is the chance that some event, A, will occur given another event, B, has happened and some dependence relationship exists between A and B. In a graph, an arrow from B to A signifies the {*B influences A*} dependence. This probability is denoted as $P(A \mid B)$ where

$$P(A \mid B) = \frac{P(A,B)}{P(B)}. \tag{1}$$

Bayes' theorem is the method of finding the converse probability of the conditional,

$$P(B \mid A) = \frac{P(A \mid B) \cdot P(B)}{P(A)}. \tag{2}$$

This conditional relationship allows an investigator to gain probability information about either A or B with the known outcome of the other. Now imagine a complex problem with n binary variables where the relationships among them are not clear for predicting one output variable. If all variables were used in one combined joint distribution, the number of possible combinations of variables would be equal to (2^n-1). If dependence relationships between these variables could be determined in which variables that are independent were removed, fewer nodes would be adjacent to the node of interest. This makes the number of variable combinations decrease significantly. Furthermore, variables that are directly conditional not to the node of interest but to the parents of the node of interest can be related, which allows a more robust system when dealing with missing data points. This property of requiring less information based on pre-existing understanding of the system's variable dependencies is a major benefit of Bayesian Networks [3]. The first BNs usually dealt with fairly well understood principles and variable relationships. In many complex instances, a researcher may have ample data for the variables of interest, but does not know the relationships between those variables in order to create the network. Hence, the network must be built in a computationally viable way while still producing accurate conditional variable dependencies [1, 3]. Several researchers have tackled this problem, the most notable being

Cooper and Herskovits who developed the K2 algorithm, a greedy-hill climb algorithm [3]. This method starts with a graph and repetitively adds nodes/edges to maximize a model-selection criterion,

$$K2\,criterion = \prod_{j=1}^{q} \frac{\Gamma\left(\sum_k a_{ijk}\right)}{\Gamma\left(\sum_k a_{ijk} + \sum_k s_{ijk}\right)} \prod_{k=1}^{r_i} \frac{\Gamma\left(a_{ijk} + s_{ijk}\right)}{\Gamma\left(a_{ijk}\right)}, \tag{3}$$

where

- i, j, k are the indexes of the child node, of the parents of the child node, and of the possible values of the child node, respectively,
- q is the number of different instantiations of parent nodes,
- r_i is the number of values that the child node can assume,
- s is the number of times that the child node has the value of the k^{th} index value of the node, and
- a is the number of times that the parents and the child correlate positively in discrete cases.

This selection criterion is basically a measure of how well the given graph correlates to the data. This method requires a dataset without any gaps and a hierarchical causal ordering of nodes. This means that nodes preceding a given node can cause it while nodes after can be a result of the node. The K2 algorithm is somewhat flawed in that it can reach a local maximum and terminate the search without finding the overall global maximum [1, 3, 4]. Several methods for random restarts such as simulated annealing and best-first search have been proposed to eliminate this problem. Nonetheless, these methods are more computationally expensive but can improve the network's accuracy when dealing with large data sets [1]. A Genetic algorithm (GA) is another tool that can be used to build the desired networks [5, 6, 7]. It begins with a sample population of randomly selected network structures and their classification accuracy. Iteratively, random crossovers and mutations of networks within a population are tested and the most fit of the population is kept for future generations. As generations pass, the population evolves leaving the fitter structures while those performing poorly become extinct. This method is quite useful due to the inherent randomness that alleviates the local maximum problem and since the structure of the resulting network dynamic without regard to individual node-to-node fitness measures that have not been proven to be optimum or accurate [4].

4 Method

To use a Bayesian Network in this problem, first, one must assume that data correlation is equivalent to statistical dependence. We also must assume that the data gathered accurately portrays the system, and with such a small dataset, this can be a difficult idea to accept or cross validate. Two different methods were used to build the Bayesian Network. The first is similar to the basic K2 algorithm developed by Cooper and Herskovits. It begins with the full set of nodes with no edges between them. By assessing the utility of adding an edge between any of the given nodes, the edge with the maximum K2 scoring utility is added. The score between nodes was calculated using the Cooper-Herskovits scoring criterion mentioned in Equation (3) [1, 3, 4]. The second BN structure discovery method tested uses a GA. The variables coded in the GA as well as their ranges are the following:

- *p* - The number of parents the node of interest has [between 2 and 7].
- *pp* - The number of parents each of the above *p* parents has [between 0 and 2].
- *f* - The feature that corresponds to each of the above nodes (one of the 12 features listed in section 2).

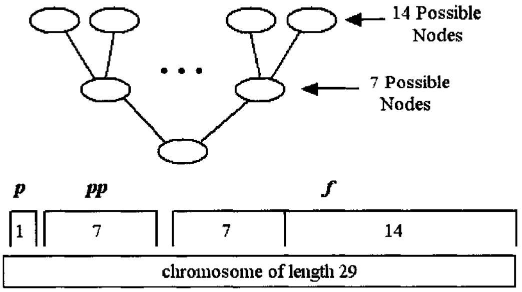


Fig. 4. The chromosome of the GA encodes with integers the number of parents and the feature which each node contains.

This chromosome information, shown in Figure 4, was coded as integers with *p* being the first in the chromosome. Then, *pp* required 7 integers to code allowing for each of the 7 possible parent nodes to have a different number of parents itself. For instance, one node can have 2 parents, while a neighboring node could have none. This allows for greater variability in the possible structures being evaluated. The features, *f*, that correspond to each of the nodes also had to account for all 21 possible node connections; 7 from the parents of the node of interest and up to 2 parents of each of them. This adds significantly to the size and complexity of the chromosome and slightly degrades the usefulness of the genetic algorithm crossovers and mutations due to some of the nucleotides being unused for the structure. But, as seen in nature, many genes of an organism stay inactive through its lifetime and are passed to further generations for later mutations or crossovers to activate, so this is seen as safeguarding diversity and consistent gene transmission, not a complexity drawback. To assess the accuracy of a network with a small sample size, a leave-one-out approach was used. This entails training the node probabilities on all but one of the patients, and testing on the remaining patient. This type of k-fold cross validation is done once for each of the patients yielding an average representation of the quality of the network building method. In order to make full use of the conditional relationship between the layers of the network, any value in the testing set has a 10% chance of exclusion. This was repeated 50 times for every trained network allowing for a fairly diverse set of testing for each network built. The metric used to determine the fitness of the network structures is the area under the curve (AUC) of the ROC curve. This is performed on the class membership probabilities output by the *age* node the network is trying to predict. This gives a numeric value to how well the network distinguishes between (classifies) the two groups.

5 Results

The network built using a greedy method similar to the K2 algorithm, performed poorly. The resulting network took on the structure shown in Figure 5, with an overall AUC of 65%. This separation between the two classes is not acceptable for medical applications.

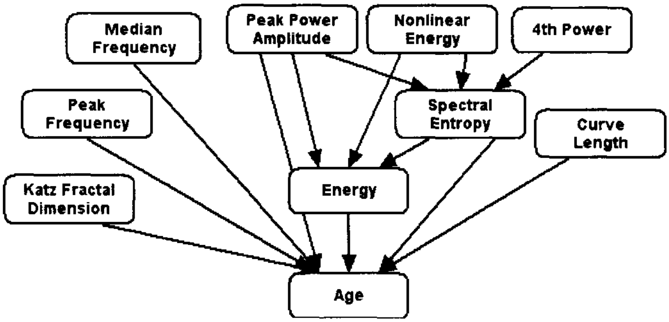


Fig. 5. Bayesian network built from modified-K2 algorithm with an AUC of 65%.

ROC plots for individual tests of the greedily built network are shown in Figure 6. The left plot shows a case with fairly good separation while, testing the same network with different missing data points, the right plot shows a case that is hardly above random guessing. This is a measure of how robust the network is to various missing data points: the greater the variance in AUC given a constant graph, the less robust the network.

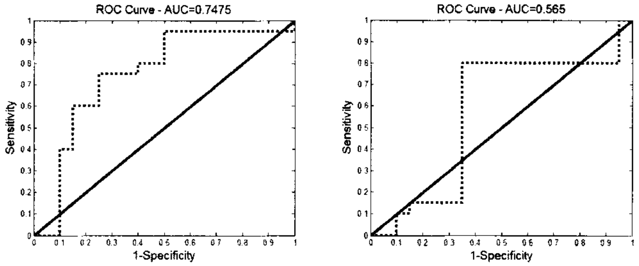


Fig. 6. Two cases (left and right) of classification of old and young patients using the network developed by the modified K2 algorithm.

The GA-evolved BN had much better results, having an AUC of 86.1% after 100 generations. This is considered good separation. Figure 7 contains the resulting network as well as the ROC curve for the resulting classification. Moreover, the resulting network is also much more robust, with most overall testing sets having between 83% and 85% AUC.

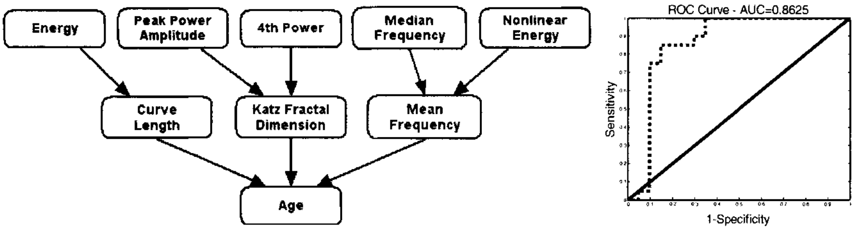


Fig. 7. Genetic algorithm evolved BN structure and the ROC curve of the *age* group classification result.

6 Discussion

When comparing the two network discovery methods presented above, the genetic algorithm developed a structure that had an overall higher accuracy and was much more robust when presented with missing data points. The networks themselves had similar composition of nodes. The GA-evolved BN contained the extra node of mean frequency while removing both peak frequency and spectral entropy, all being frequency measures that could have contained similar information. Both networks also had curve length, Katz fractal dimension and mean or median frequency as a parent of the node of interest.

Though this instance resulted in a similar number of nodes for both the greedy and evolved networks, 9 and 8, respectively, the difference in the number of connections or edges is significant, 13 and 8 respectively. Fewer parents to the node of interest make the conditional probability table more accurate and easier to build. For example, the greedily built network has *age* with 7 binary parents, making the number of data combinations 2^7-1 or 127. This requires a lot of sample data to accurately assess probabilities for each of these outcomes. The evolved network presents *age* with 3 parents, making a total of 7 possible data outcomes. This reduction in needed data is very important if the network is to be used in practice and save both funds and time through lower data collection.

7 Conclusion

This paper presents an age classification method based on the statistical presence of ECG features analyzed in a Bayesian Network evolved using a genetic algorithm. The comparison of a greedy hill-climb and the genetic algorithm-based method for network structure discovery shows a large increase in classification accuracy for the latter, as measured by the area under the curve (AUC) of the Receiver Operating Characteristic (ROC) curve.

Interesting results have come from a relatively small group of features. For instance, curve length, Katz fractal dimension as well as Hurst Parameter are highly correlated features and therefore repetitive. Future improvements will incorporate more diverse features coming from several differing domains e.g., wavelet, statistical, frequency, and nonlinear. With more accurate feature selection, further improvements of classification accuracy can be accomplished.

One of the drawbacks of this study has been the method for binary discretization used after feature extraction. Currently, the same set of data is used in this threshold determination as for the test of the final network. This is not preferred but due to a small sample size restriction. While this small data set does not allow for expansion of the network into more than binary variables, a 3-level discretized input variable set could also allow further probabilistic differentiation between the classes. Overall, a larger sample set could allow further improvements as well.

Further exploration of the encoding of the network structure should also be performed. This could enable more meaningful crossover changes to occur, allowing for better overall evolution, and determining the best network much more quickly and efficiently. Also, the fitness function should penalize for overly complex graphs that make sufficient data collection impossible. The next step is to move this technology toward use on a medical problem with complex classification problems that would benefit from feature exploitation in a Bayesian Network.

The medical community has relied on limited variable combination methods for much too long, especially while there are advanced methods of data mining and decision-making to be harnessed. The BN is an excellent method for making decisions based on collected information. The only difficulty is determining the structure of the network that gives the highest possible accuracy. With a genetic algorithm evolving the network, it is not only easy to implement, but, as it turned out, extremely accurate.

Acknowledgments

The authors gratefully acknowledge the Dana Foundation for the grant that supports this research.

References

1. Heckerman, D., *A Tutorial on Learning With Bayesian Networks.* 1995, Microsoft Research.
2. Goldberger, A.L., et al., *PhysioBank, PhysioToolkit, and PhysioNet : Components of a New Research Resource for Complex Physiologic Signals.* Circulation, 2000. 101(23): p. 215e-220. [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]
3. Neapolitan, R., *Learning Bayesian Networks.* 2004, London: Pearson Printice Hall.
4. Larranaga, P., et al., *Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996. 18(9): p. 912-26.
5. Wong, M.L. and Leung, K.S., *An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach*, IEEE Transactions on Evolutionary Computation, 2004. 8(4), p. 378 – 404.
6. Myers, J., and Laskey, K.B., *Learning Bayesian Networks from Incomplete Data with Stochastic Search Algorithms*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), 1999: p. 458-465.
7. Lee, KY, Wong, ML, Liang, Y, Leung, KS, and Lee, KH, *A-HEP: Adaptive Hybrid Evolutionary Programming for Learning Bayesian Networks*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), 2004, late breaking paper, 12 pages.

A Real-Time Hand Gesture Interface for Medical Visualization Applications

Juan Wachs[1], Helman Stern[1], Yael Edan[1], Michael Gillam[2], Craig Feied[2], Mark Smith[2], and Jon Handler[2]

[1]Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er-Sheva, Israel, 84105,
{helman, yael, juan}@bgu.ac.il.
[2]Institute for Medical Informatics, Washington Hospital Center, 110 Irving Street, NW, Washington, DC, 20010,{feied, smith, handler, gillam}@medstar.net

Abstract. In this paper, we consider a vision-based system that can interpret a user's gestures in real time to manipulate objects within a medical data visualization environment. Dynamic navigation gestures are translated to commands based on their relative positions on the screen. Static gesture poses are identified to execute non-directional commands. This is accomplished by using Haar-like features to represent the shape of the hand. These features are then input to a Fuzzy C-Means Clustering algorithm for pose classification. A probabilistic neighborhood search algorithm is employed to automatically select a small number of Haar features, and to tune the fuzzy c-means classification algorithm. The gesture recognition system was implemented in a sterile medical data-browser environment. Test results on four interface tasks showed that the use of a few Haar features with the supervised FCM yielded successful performance rates of 95 to 100%. In addition a small exploratory test of the Adaboost Haar system was made to detect a single hand gesture, and assess its suitability for hand gesture recognition.

Keywords: haar features, fuzzy c-means, hand gesture recognition, neighborhood search, computerized databases.

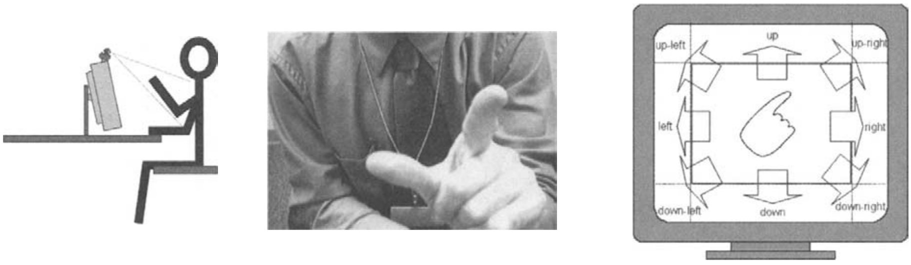
1 Introduction

Computer information technology is increasingly penetrating into the hospital domain. It is important that such technology be used in a safe manner in order to avoid serious mistakes leading to possible fatal incidents. Unfortunately, It has been found that a common method of spreading infection includes computer keyboards and mice in intensive care units (ICUs) used by doctors and nurses [7]. Many of these deficiencies may be overcome by introducing a more natural human computer interaction (HCI), especially speech and gesture. Face gestures are used in FAce MOUSe [6] whereby a surgeon can control the motion of the laparoscope. Gaze, is used as one of the diagnostic imaging techniques for selecting CT images by eye movements [12]. Here a vision-based gesture capture system to manipulate windows and objects within a graphical user interface (GUI) is proffered. Research using a hand gesture computer vision system appeared in [4]. In [13] the tracking position of fingers is used to collect quantitative data about the breast palpation process. In our work we consider hand motion and posture simultaneously. Our system is user independent without the need of a large multi-user training set. We use a fuzzy c-mean discriminator along with Haar type features. In order to obtain a more optimal system design we employ a neighborhood search method for efficient feature selection and classifier parameter tuning. The real time operation of the gesture interface was tested in a hospital environment. In this domain the non-contact aspect of the gesture interface avoids the problem of possible transfer of contagious diseases through traditional keyboard/mice user interfaces.

A system overview is presented in Section 2. Section 3 describes the segmentation of the hand from the background. Section 4 deals with feature extraction and pose recognition. The results of performance tests for the FCM hand gesture recognition system appear in Section 5. Section 6 concludes the paper.

2 System Overview

A web-camera placed above the screen (Figure. 1(a)) captures a sequence of images like those shown in Figure 1(b). The hand is segmented using color cues, a B/W threshold, and various morphological image processing operations. The location of the hand is represented by the 2D coordinates of its centroid, and mapped into one of eight possible navigation directions of the screen (see Figure 1(c)) to position the cursor of a virtual mouse. The motion of the hand is interpreted by a tracking module. At certain points in the interaction it becomes necessary to classify the pose of the hand. Then the image is cropped tightly around the blob of the hand and a more accurate segmentation is performed. The postures are recognized by extracting symbolic features (of the Haar type) from the sequence of images. The sequence of features is interpreted by a supervised FCM that has been trained to discriminate various hand poses. The classification is used



(a) (b) (c)

Fig. 1. (a)(b) Gesture Capture. (c) Screen navigation map

to bring up X-rays images, select a patient record from the database or move objects and windows in the screen. A two-layer architecture is used. The lower level provides tracking and recognition functions, while the higher level manages the user interface.

3 Segmentation

In order to track and recognize gestures, the CAMSHIFT [2] algorithm is used together with an FCM algorithm [10]. For CAMSHIFT, a hand color probability distribution image is created using a 2D hue-saturation color histogram [3]. This histogram is used as a look-up-table to convert a camera image into a corresponding skin color probability image through a process known as back propagation. A backprojected image assigns to each pixel a number between 0 and 1 as the likelihood of it being classified as a hand pixel. Thresholding to black and white, followed by morphological operations, is used to obtain a single component for further processing to classify the gestures.

The initial 2D histogram is generated in real-time by the user in the 'calibration' stage of the system. The interface preview window shows an outline of the palm of the hand gesture drawn on the screen. The user places his/her hand within the template while the color model histogram is built (Fig. 2), after which the tracking module (Camshift) is triggered to follow the hand. The calibration process is initiated by the detection of motion of the hand within the region of the template. In order to avoid false motion clues originated by non hand motion a background maintenance operation is maintained. A first image of the background is stored immediately after the application is launched, and then background differencing is used to isolate the moving object (hand) from the background.

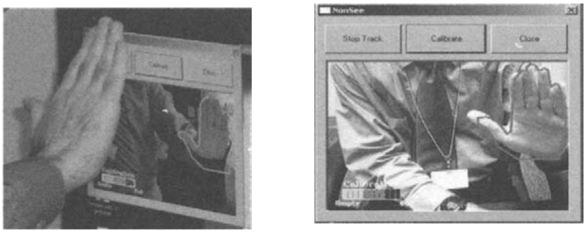


Fig. 2. User hand skin color calibration

Since background pixels have small variations due to changes in illumination over an extended period of time, the background image must be dynamically changed. Background variations are identified by a threshold applied to the absolute difference between every two consecutive frames. If the difference is under some threshold t_1, then the current images contain only a background, otherwise, an upper threshold level t_2 is checked to test whether the present object is a hand. In case that the current image is a background, the background stored image is updated using a running smoothed average.

$$Bcc_k(i,j) = (1-\alpha) * Bcc_{k-1}(i,j) + \alpha * f(i,j) \tag{1}$$

In (1) Bcc_k is the updated stored background image at frame k, Bcc_{k-1} is the stored background image at frame k-1, α is the smoothing coefficient (regulating update speed), f(i,j) is the current background image at frame k. Small changes in illumination will only update the background while huge changes in intensity will trigger the tracking module. It is assumed that the hand is the only skin colored object moving on the area of the template. The process of calibration requires only a few seconds, and is necessary as every user has a slightly different skin color distribution, and changes in artificial/daylight illumination affect the color model. A low threshold and open and close morphology operations followed by largest component selection are applied to obtain a single connected blob (Fig. 3).

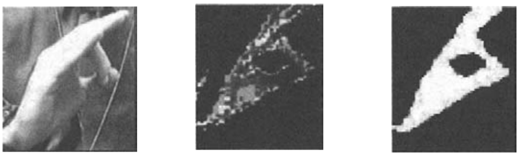


Fig. 3. Image processing of the pose

4 Feature Extraction and Pose Recognition

4.1 Hand Tracking and Pose Recognition

Hand gestures are classified using a finite state machine (Fig. 4).When a doctor wishes to move the cursor over the screen, the hand moves out of the 'neutral area' to any of 8 directional regions (Fig. 5). This interaction allows the doctor to rest the hand in the 'neutral area'.

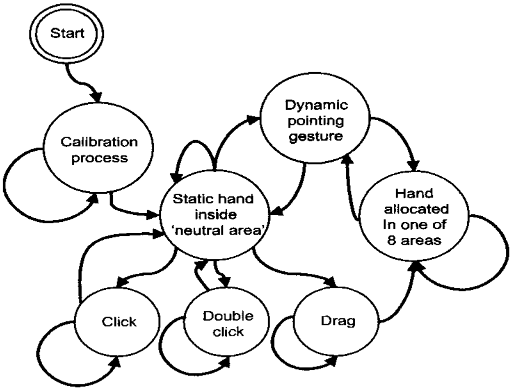


Fig. 4. State machine for the gesture-based medical browser

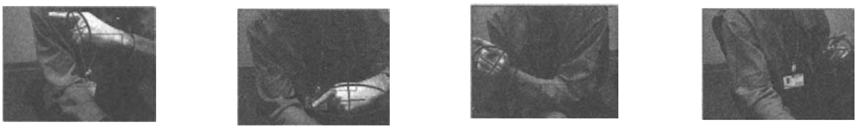
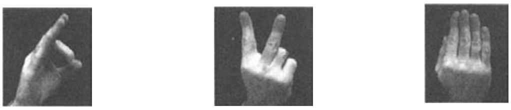


Fig. 5. Four quadrants mapped to cursor movement

To facilitate positioning, hand motion is mapped to cursor movement. Small, slow hand (large fast) motion cause small (large) pointer position changes. In this manner the user can precisely control pointer alignment.When a doctor decides to perform a click, double-click, or drag with the virtual mouse, he/she places the hand in the 'neutral area' momentarily to trigger the transition. These three poses are shown in Figure 6.



(a). Click (b). Double-click (c). Drag

Fig. 6. The gesture vocabulary

4.2 Haar Features

Basically, the features of this detector are weighted differences of integrals over rectangular sub regions. Figure 7(a)-(d) visualizes the set of available feature types, where black and white rectangles correspond to positive and negative weights, respectively. The feature types consist of four different edge-line features. The learning algorithm automatically selects the most discriminate features considering all possible feature types, sizes and locations. These features are reminiscent of Haar wavelets, which can be computed in constant time at any scale, and use the original image without preprocessing. Each rectangular feature is computed by summing up pixel values within smaller rectangles, see Eq. (2).

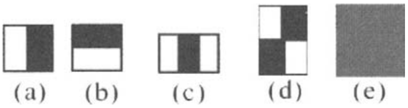


Fig. 7. Extended integral rectangle feature set

$$f_i = \sum_{i \in I = \{1,\dots,N\}} \omega_i * \mathrm{Re}\, cSum(r_i) \qquad (2)$$

In (2) $\omega_i \in \Re$ are weights, r_i is the *ith* rectangle, and N is the number of rectangles. The weights have opposite signs (indicated as black and white in Figure. 7), and are used to compensate between differences in area. Efficient computation is achieved by using summed area tables. We have added a block average feature (see Fig. 7(e)) to f_1 , f_2 , f_3 , and f_4 (see Fig. 7(a)-(d)) selected from the original feature set of Viola-Jones. A rectangle, r, in the image can be defined by the (x,y) position of its upper left corner, and by its width w and height h. We constrain the total set of rectangles in an image, by using the relation: x=w*n, and y=h*m. where n and m are integer numbers. Hence, the total number of rectangles is less than 13,334 in lieu of (>750,000) for a 100x100 resolution classifier using the full Viola-Jones set 9].

4.3 Pose Recognition

In our system we reduce the Haar rectangular positions severely to a set of 'selected' rectangles v. These rectangles are limited to lie within a bounding box of the hand tracking window, and are obtained by dividing the window into m rows and n columns. For each cell a binary variable is used to decide whether it is selected or not. A more elaborate strategy enables one to define the type of feature for selected rectangles. Therefore, a set of rectangles in a window is defined by {n,m,t}, where n, m are columns and rows; and t={t_1,...,t_i,...,t_v} represent the type of feature of rectangle i (indexed row wise from left to right). The feature type t can take integer values from 0 to 5, where 0 indicates that the rectangle is not selected, and 1,2,3,4,5 represent features of type f_1 , f_2 , f_3 ,f_4 and f_5, respectively. The hypothesis expressed in Viola and Jones is that a very small number of these features can be combined to form an effective classifier. As opposed to Viola and Jones method, our learning algorithm is not designed to select a single rectangle feature which best separates the positive and negative for each stage of a cascade of classifiers. Instead, we evaluate a set of rectangle features simultaneously, which accelerates the process of feature selection. The Haar features selected are input into the hand gesture FCM recognition system. Note, that feature sizes are automatically adjusted to fit into a dynamically changing bounding box created by the tracking system.

4.4 Optimal Feature and Recognition Parameter Selection

Feature selection and finding the parameters of the FCM algorithm for classifying hand gestures is done by a probabilistic neighborhood search (PNS) method [8]. The PNS selects samples in a small neighborhood around the current solution based on a special mixture point distribution:

$$\mathrm{PS}(\mathrm{x}\ |\,\mathrm{h}) = \begin{cases} \mathrm{h}, & x = 0 \\ h((1-h)^{|x|})/2, & x = \pm 1, \pm 2,,,,\pm(S-1) \\ ((1-h)^{|x|})/2, & x = \pm S \end{cases} \tag{3}$$

Where, S = maximum number of step increments, h = probability of no change, x_j = a random variable representing the signed (positive or negative coordinate direction) number of step size changes for parameter p_j, and $P_S(x|h) = P_r(x = s)$ the probability of step size s, given h.

Figure 8 shows an example of the convergence behavior of the PNS algorithm for 5 randomly generated starting solutions. Figure 9 shows the optimal set of features selected by this run. The features f_4 and f_5 capture characteristics of a palm based gesture using diagonal line features and average grayscale. Inner-hand regions (such inside the palms) and normal size fingers are detected through f_1, while f_3 captures the ring finger based on edge properties. Hence, this is quite different from traditional gesture classifiers which rely on parametric models or statistical properties of the gesture. Note, that the result is a set of common features for all three of our pose gestures. The optimal partition of the bounding box was 2x3 giving 6 feature rectangles. The parameter search routine found both the number of sub blocks and the type of Haar feature to assign to each.

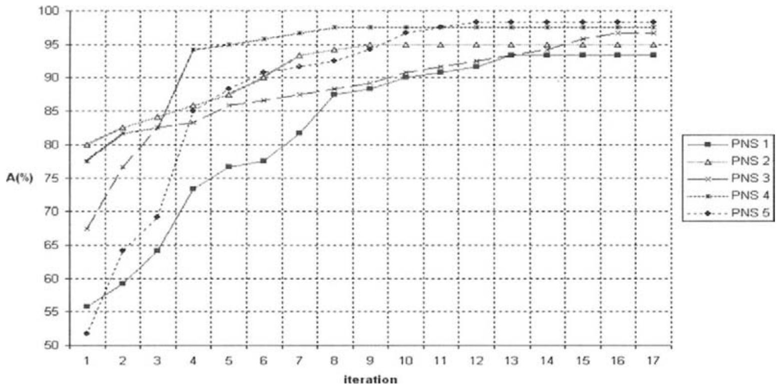


Fig. 8. Convergence curve for five solutions of the PNS algorithm.

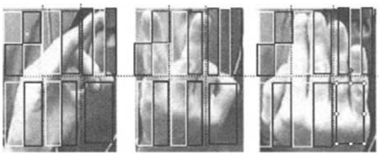


Fig. 9. Automatically selected features (f_4, f_1, f_3, f_1 ,f_1 , f_5) for the 2x3 partition

5 Test of the Hand Gesture FCM Classifier

To evaluate the overall performance of the hand gesture tracking and FCM recognition system, we used the Azyxxi Real-time Repository ™ [1], which was designed to accommodate multi-data types. The data-set consists of 20 trials of each of 4 tasks: Select Record of Patient, Browse X-ray collection, Select specific X-ray and Zoom in Damaged Area. The user was asked to perform the tasks sequentially. The total results for one experienced user are shown in Table 1. The success task rate shows the times an action (part of the task) was performed correctly without catastrophic errors. Minor errors are related to inaccurate position of the cursor due to fast movements or changes in direction, while catastrophic errors occurred as a result of misclassification of the supervised FCM. In general, the results below indicate both the ability of the system to successfully track dynamic postures; and classify them with a high level of accuracy.

Table 1. Results of medical tasks using hand gestures

Task	Steps	Trials	Success Task
Select Record of Patient	1	19	94.74%
Browse X-ray collection	2	20	100%
Select specific X-ray	1	20	100%
Zoom in Damaged Area	2	19	94.74%

6 Conclusions

In this paper, we consider a vision-based system that can interpret a user's gestures in real time to manipulate objects within a medical data visualization environment. A hand segmentation procedure first extracts binary hand blobs from each frame of an acquired image sequence. Dynamic navigation gestures are translated to commands based on their relative positions on the screen. Static gesture poses are identified to execute non-directional commands. This is accomplished by using Haar-like features to represent the shape of the hand. These features are then input to the FCM algorithm for pose classification. The PNS algorithm is employed to automatically select a small number of visual features, and to tune the FCM algorithm. Intelligent handling of features allows non discriminating regions of the image to be quickly discarded while spending more computation on promising discriminating regions. The gesture recognition system was implemented in a sterile medical data-browser environment [11] . Test results on four interface tasks showed that the use of these simple features with the supervised FCM yielded successful performance rates of 95 to 100%, which is considered accurate enough for medical browsing and navigation tasks in hospital environments. The explanation for the 5% drop in accuracy is due to the confusion between the 'double click' and 'drag' gestures, as a result of lack of samples in the training and testing sets. In a future study gestures that include shadows, occlusion and change in geometry will be used to enrich the datasets. Another issue to be addressed is false triggers as a result of a fast moving objects others than the hand. An approach to this problem is to store a generic 2D skin color distribution trained off line, and to compare it to the candidate object color histogram. Catastrophic errors due to confusion between gestures can be reduced by using the probabilities of gesture occurrences in a transition matrix based on the state machine presented in Fig. 5. An appealing alternative method for fast recognition of a large vocabulary of human gestures suggests using Haar features to reduce dimensionality in hand attention images. Future work includes recognition of dynamic two handed gestures for zooming, rotating images, and testing with larger gesture vocabularies.

Acknowledgments

This project was partially supported by the Paul Ivanier Center for Robotics Research & Production Management, Ben Gurion University.

References

1. Azyxxi Online Source (2003) Available: http://www.imedi.org/dataman.pl?c=lib&dir=docs/Azyxxi
2. Bradski GR (1998) Computer vision face tracking for use in a perceptual user interface. In Intel Technical Journal, pp 1-15.
3. Foley JD, van Dam A, Feiner SK and Hughes JF (1987) Computer graphics: principles and practice, 2 Ed, Addison Wesley
4. Graetzel C, Fong TW, Grange S, and Baur C (2004) A non-contact mouse for surgeon-computer interaction. J Tech and Health Care 12:3:245-257
5. Lienhart R and Maydt J (2002) An extended set of haar-like features for rapid object detection. In IEEE ICIP 2002 vol:1, pp 900-903
6. Nishikawa A, Hosoi T, Koara K, Negoro D, Hikita A, Asano S, Kakutani H, Miyazaki F, Sekimoto M, Yasui M, Miyake Y, Takiguchi S, and Monden M (2003) FAce MOUSe: A novel human-machine interface for controlling the position of a laparoscope. IEEE Trans on Robotics and Automation 19:5:825-841.
7. Schultz M, Gill J, Zubairi S, Huber R, Gordin F (2003) Bacterial contamination of computer keyboards in a teaching hospital. Infect Control Hosp Epidemiol 24:302-303
8. Stern H, Wachs JP, Edan Y (2004) Parameter calibration for reconfiguration of a hand gesture tele-robotic control system. In Proc of USA Symp on Flexible Automat, Denver, Colorado, July 19-21
9. Viola P and Jones M (2001) Rapid object detection using a boosted cascade of simple features. In IEEE Conf on Computer Vision and Pattern Recog, Kauai, Hawaii
10. Wachs JP, Stern H, and Edan Y (2005) Cluster labeling and parameter estimation for automated set up of a hand gesture recognition system. In IEEE Trans in SMC Part A 2005. vol. 35, no. 6, pp: 932- 944.
11. Wachs JP, Stern H (2005) Hand gesture interface for med visual app web site. Available: http://www.imedi.org/docs/references/gesture.htm
12. Yanagihara Y, Hiromitsu H (2000) System for selecting and generating images controlled by eye movements applicable to CT image display, Medical Imaging Technology, September, vol.18, no.5, pp 725-733
13. Zeng TJ, Wang Y, Freedman MT and Mun SK (1997) Finger Tracking for Breast Palpation Quantification using Color Image Features. SPIE Optical Eng 36:12, pp 3455-3461

Part V

Classification

A Hybrid Intelligent System and Its Application to Fault Detection and Diagnosis

Chee Siong Teh[1, 2] and Chee Peng Lim[1]

School of Electrical and Electronic Engineering[1]
University of Science Malaysia, Engineering Campus
14300 Nibong Tebal, Penang, Malaysia

Faculty of Cognitive Sciences and Human Development[2]
University Malaysia Sarawak
94300, Kota Samarahan, Sarawak, Malaysia

csteh@fcs.unimas.my, cplim@eng.usm.my

Abstract. This paper proposes a hybrid system that integrates the SOM (Self-Organizing Map) neural network, the kMER (kernel-based Maximum Entropy learning Rule) algorithm and the Probabilistic Neural Network (PNN) for data visualization and classification. The rationales of this hybrid SOM-kMER-PNN model are explained, and the applicability of the proposed model is demonstrated using two benchmark data sets and a real-world application to fault detection and diagnosis. The outcomes show that the hybrid system is able to achieve comparable classification rates when compared to those from a number of existing classifiers and, at the same time, to produce meaningful visualization of the data sets.

1 Introduction

Advanced methods of visualization are becoming crucial to uncover important structures and interesting correlations in data in the effort to generate useful, meaningful, and even unpredictable information from flood of data. A large number of artificial neural networks and machine learning algorithms, particularly for data projection, have been proposed [1-5]. Projection of multivariate data enables the visualization of high dimensional data in order to better understand the underlying structure, explore the intrinsic dimensionality, and analyze the clustering tendency of multivariate data [5]. Indeed, visualization can lead to better understanding of the underlying data structure as it takes advantage of human's perceptive and associative abilities to perceive clusters and correlations in data.

As pointed out in [6], unsupervised competitive learning (UCL) networks, such as the SOM, can yield grid units that are never active and these units result in less optimal usage of the map's resources. This also implies that the amount of information about the input distribution is not being transferred optimally onto the projected map. In view of this, a number of topographic map formation algorithms [7-8] have been proposed by adding a "conscience" to the weights update process, so that every grid unit is used equally.

The kMER [9] is an algorithm that formulates an equiprobabilistic map with the maximum entropy learning rule. The grids from this topologypreserving mapping have an equal probability to be active. kMER updates the prototype vectors and the radii of the kernels centered at these vectors to model the input density at convergence. This approach produces a topographic map with maximum transfer of information about the input distribution, this also known as the "faithful representation" [10]. In addition to providing the desired visualization, such map is useful for other purposes such as density estimation, clustering, and classification. In this sense, the kMER algorithm seems to be able to reduce the limitation of the SOM algorithm as mentioned earlier.

The kMER algorithm, on the other hand, suffers from limitations in terms of its computational efficiency. The use of neighborhood relations is the most essential principle in the formation of the topology-preserving mapping [9]. As a result, the kMER algorithm needs to consider a large number of active neurons simultaneously at the early stage of its learning process. This leads to computational inefficiency and a slow formation of the topographic map. On the contrary, the SOM network only considers an active neuron at a time to every input pattern and does not define any receptive field (RF) region. By exploiting the advantages as well as drawbacks of both methods, this work takes a hybrid approach that harnesses the benefits offered by both SOM and kMER.

The benefits of utilizing hybrid intelligent systems for developing useful industrial applications, such as [11], have increasingly gained recognition from many researchers. In a hybrid system, two or more techniques are combined in a manner that overcomes the limitations of individual technique [12]. Hybrid intelligent systems represent not only the combination of different intelligent techniques but also the integration of independent, self-contained intelligent processing modules that exchange information and perform separate functions to generate solutions [13].

The organization of this paper is as follows. First, the proposed hybrid model is briefly described. Then, a neural network model that implements a classical non-parametric density estimation procedure is presented. This is followed by simulation studies of two benchmarked data sets and an elaboration on the application of the hybrid SOM-kMER-PNN model to a real-world fault detection and diagnosis task in a power generation plant. The outcomes are discussed and analyzed in terms of the visualization of multivariate projection and a performance comparison between the hybrid learning system and other existing methods. A summary of concluding remarks is presented at the end of the paper.

2 The Hybrid SOM-kMER Model

Both the kMER and SOM algorithms utilize the neighborhood relation principle in the formation of a topology-preserving mapping. The SOM algorithm employs the winners-take-all (WTA) operation, in which a unique neuron (winner) is assigned to every input pattern. However, the kMER algorithm requires the definition of RF regions, in which the RFs of neighboring neurons are mutually overlapped (and more overlapping occurs at the early stage of learning). Consequently, the kMER needs to consider several "winners" at a time. This leads to computational inefficiency and slow formation of the topographic map.

Besides, the kMER algorithm requires a high processing cost and employs a small learning rate. In view of this, the hybrid SOM-kMER model proposes the utilization of the SOM batch algorithm [14] at the early stage of the learning process of the kMER to establish more robust neighborhood relations. The weighs obtained after performing some learning iterations will coincide with the generalized median of the input vectors [14], and this allows the neighboring neurons "learn" from the input vector at a faster rate. The main aim of this approach is to speed up the learning process. Once the neighborhood relations are present after the SOM batch learning process, the neighborhood function from the subsequent kMER learning is excluded in the SOM-kMER model. The purely local weight updates and mutually overlap RF regions used in the kMER is sufficient to disentangle the lattice and, hence, is able to achieve a topology-preserving mapping.

3 The Hybrid SOM-kMER-PNN Model

The Probabilistic Neural Network (PNN) is a simple but powerful nonparametric classifier, which was proposed by [15] and originated from Parzen's [16] kernel-based probability density function (pdf). The PNN algorithm uses all the samples in the training set to estimate the pdf and to perform classification. If the data samples in the training set are corrupted by noise, the classification performance may be affected. In the hybrid SOM-kMER-PNN model, instead of the original training samples (often with a large size), the prototype vectors from each class of the trained SOM-kMER map are used to estimate the pdf. The advantages of the integration are two-fold. First, SOM-kMER is used as the underlying clustering algorithm to reduce the number of pattern nodes required in the PNN; second, the PNN is used as the probability estimation algorithm to provide probabilistic prediction from SOM-kMER.

Suppose that the trained SOM-kMER map has N_{C_i} (the number of RF regions of class C_i) nodes with label C_i and the corresponding prototype vectors are $\mathbf{w}_j^{C_i}, j = 1,2,3..,N_{C_i}$, the pdf of class C_i is then estimated by using:

$$p(\mathbf{v} \mid C_i) = \frac{1}{N_{C_i}(2\pi\sigma_j^2)^{d/2}} \sum_{j=1}^{N_{C_i}} \exp\left(- \frac{\left\| \mathbf{v} - \mathbf{w}_j^{C_i} \right\|}{2\sigma_j^2} \right), \forall j = 1,2,3...,N_{C_i} \tag{1}$$

After obtaining the estimated pdf of each class, the posterior probabilities are determined by using:

$$P(C_{i*} \mid \mathbf{v}) = \frac{p(\mathbf{v} \mid C_i)P(C_i)}{\sum_{j=1}^{k} p(\mathbf{v} \mid C_j)P(C_j)} \tag{2}$$

with C_i the class labels, k the number of classes, $P(C_i)$ the a priori probability of class C_i, $p(\mathbf{v} \mid C_i)$ the conditional density of class C_i and $\mathbf{v}$ a given sample. Once the grids are labeled, we can determine the class-conditional densities and classify a given sample into class $P(C_{i*})$ when

$$p(\mathbf{v} \mid C_{i*})P(C_{i*}) > p(\mathbf{v} \mid C_j)P(C_j), \quad \forall j \neq i* \tag{3}$$

Note that $P(C_i) = N_{C_i} / \Sigma_j N_{C_i}$ since with kMER, each RF region is equiprobabilistic and σ_j are the kernel values located on each prototype weight. The proposed hybrid approach has the following advantages:

1. SOM-kMER reduces the number of pattern nodes as required in the PNN
2. SOM-kMER provides a good representative of the training samples; it makes the PNN classifier more robust in the presence of noise in data.
3. The selection of the "smoothing parameter" of the PNN is automated; in contrast to other optimization techniques, which need to be tuned using the trial-and-error [17] or cross-validation methods [18].

Since SOM-kMER allocates a kernel at each neuron weight, instead of at each input sample such as the GTM algorithms [19], it can go beyond the Parzen-window technique that utilizes fixed radii kernels. This leads to variable kernel estimation for non-parametric density estimation. The process of the hybrid algorithm is shown in Figure 1.

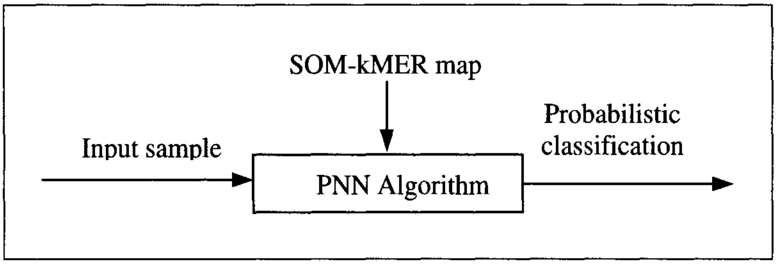


Fig. 1. The proposed hybrid SOM-kMER-PNN model

4 Benchmark Data Sets

A comparison of classification performance was made using two benchmark data sets, i.e., *Ionosphere* and *Pima Indian*, obtained from the UCI repository of Machine Learning Repository. For each performance evaluation, 80% of the data was randomly selected from the data set for training and the remaining 20% was used for testing. The data samples were converted into the required format of each algorithm. A total of five runs, each with different training and test sets, were conducted, and the results averaged. Table 1 shows the classification accuracy rates in percentages for each data set. The performance of kMER-SOM-PNN was compared with those of SOM-*k*-Nearest-Neighbor Classification (with k=5) as well as other classifiers reported in [20]

Table 1. Classification accuracy rates in (%) of (I) Ionosphere data set; and (II) Pima Indian data set for SOM-kMER-PNN and other classifiers reported in [20].

	a	b	c	d	e	f	g	h	i	j	k	l	m
I	**91.14**	86.85	91.56	91.82	93.65	86.89	90.98	88.58	88.29	83.80	85.91	89.70	87.60
II	**72.68**	70.62	71.02	71.55	73.16	73.51	72.19	71.28	50.00	68.52	71.37	68.50	70.57

Note: a=SOM-kMER-PNN, b=SOM-NNC (k =5), c=C4.5, d=C4.5 rules, e=ITI, f=LMDT, g=CN2, h=LVQ, i=OC1, j=Nevprop, k=K5, l=Q*, m=RBF

Notice that the results of SOM-kMER-PNN were better than those of SOM-NNC for both data sets. The SOM-kMER-PNN results ranked third and fourth out of thirteen different classifiers in the Pima Indian and Ionosphere benchmark tests, respectively. Nevertheless, one added advantage of the SOM-kMER-PNN model is its capability of generating visualization for the underlying data structures, in addition to performing classification.

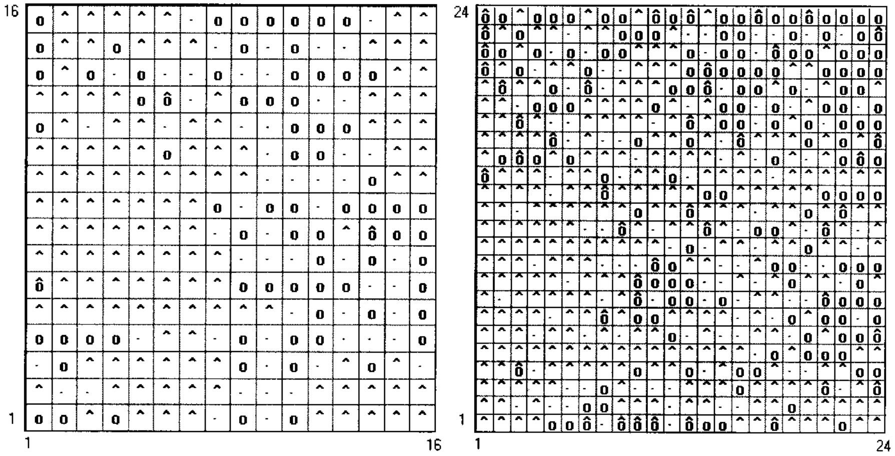


Fig. 2. Visualization of the Ionosphere data set (left – 16x16 map grid) and Pima Indian data set (right –24x24 map grid) generated using hybrid SOMkMER- PNN model. Labels “o” and “^” indicate the distinction of the two classes.

The maps in Fig. 2 clearly show that the overlapping of class boundaries for the Pima Indian data set was greater when compared to that of the Ionosphere data set. This implies that the classes of the Pima Indian data set were not as distinct as those of the Ionosphere data set. The grids of the maps were labeled using the minimum Euclidean distance technique. Such visualizations further confirmed the classification performance obtained for both data sets, in which a better performance was obtained when using the Ionosphere data set. This is the key benefit of the SOM-kMERPNN model in providing visualization of the underlying data structures that is unavailable in most classification methods, e.g. those used in [20].

5 Fault Detection and Diagnosis

In this real-world application, the focus is on fault detection and diagnosis in a power generation plant in Penang, Malaysia. The system under scrutiny is the Circulating Water (CW) system (see Figure 3) in the power plant station, with specific attentions to the conditions of the condenser performance and the detection of blockage in the system.

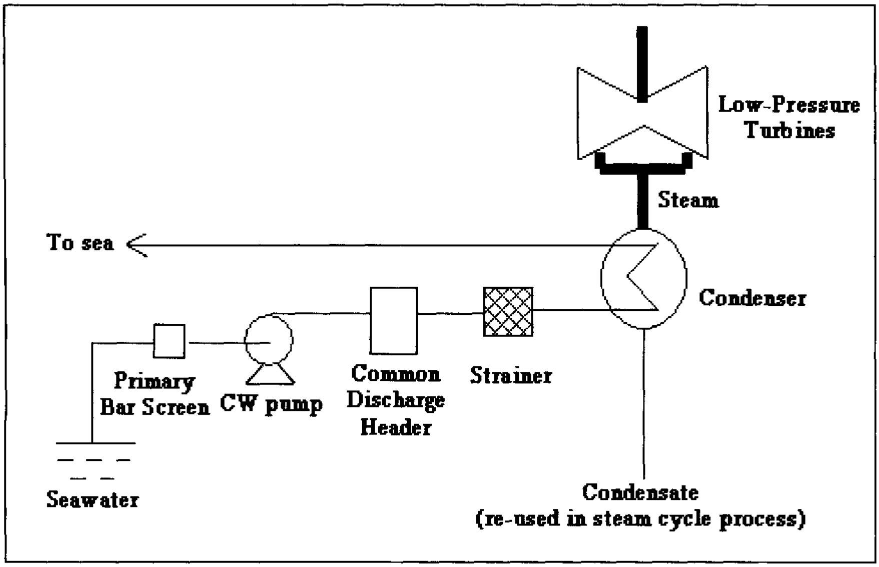


Fig. 3. The CW system (adapted from [22])

The functions of the turbine condensers are to use the circulating water to remove rejected energy (heat) from the low-pressured steam and, at the same time, to keep the turbine back-pressure (condenser vacuum) at the lowest possible yet constant level [21]. Hence, heat transfer in the condenser has a significant effect on condenser backpressure, where high efficient heat transfer will assist in maintaining condenser backpressure at a low level. With a satisfied level of condenser backpressure, it will aid in maintaining high turbine work efficiency to generating power. On that account, factors such as the pressure and temperature of the exhaust steam and the cooling water have profound influence on the performance of the condenser in the process of condensation.

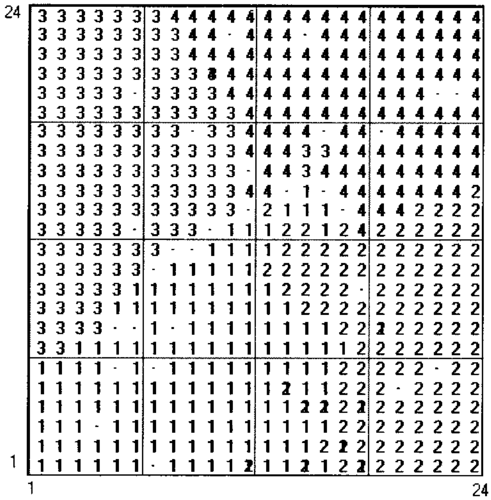
The cleanliness of the tubes in the condenser on the other hand, has a significant impact on the condenser's ability to transfer heat from the exhaust steam to the cooling water. One of the most common causes to the blockage incident is tube fouling. Occasionally, carried by the circulating water, there may be mud, and fine solid materials such as seaweed, shells, and sand, which have successfully escaped from the filters in the CW system. They may block the tubes and may affect the efficiency of the condenser in cooling the exhaust steam.

5.1 The Experiments

There were a total of 2439 samples available. These data samples were collected on 80MWatts of the targeted power generation from several cycles of light-up period of a power-generating unit. Each data sample consisted of 12 measurements of temperature and pressure at various inlet and outlet points of the condenser. This study involved a 4-class (fault-states) classification problem, i.e.,

1. Heat transfer in the condenser is efficient and there is no significant blockage in the CW system.
2. Heat transfer in the condenser is not efficient and there is no significant blockage in the CW system.
3. Heat transfer in the condenser is efficient and there is significant blockage in the CW system.
4. Heat transfer in the condenser is not efficient, and there is significant blockage in the CW system.

The available data were partitioned into a training set and a test set, each containing 1000 samples. The SOM-kMER-PNN model was used to generate visualization of the data set and to perform the classification task. Figure 4 shows the topology-preserved maps generated by the model. The minimum Euclidean distance was used to label the grid units belonging to Classes 1 to 4. It is obvious that the model was able to provide visualization of the cluster boundaries. One can observe that the cluster location for class 1 and the cluster location for class 4 were at the opposite ends. This is because the fault-state of class 1 and class 4 is the most distinctive when compared with the neighboring clusters (class 2 and class 3).



3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	4	4	·	4	4	4	·	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	·	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	·	·	4
3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	·	3	3	4	4	4	4	·	4	4	·	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3	3	4	4	4	3	3	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3	3	·	4	4	3	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3	3	4	4	·	1	·	4	4	4	4	4	4	4	2
3	3	3	3	3	3	3	3	3	3	3	·	2	1	1	1	·	4	4	4	2	2	2	2
3	3	3	3	3	·	3	3	3	·	1	1	1	2	2	1	2	4	2	2	2	2	2	2
3	3	3	3	3	3	3	·	·	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	·	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	1	1	1	1	1	1	1	1	2	2	2	2	·	2	2	2	2	2	2
3	3	3	3	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
3	3	3	3	·	·	1	·	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2
1	1	1	1	·	1	·	1	1	1	1	1	1	1	1	1	2	2	2	2	2	·	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	2	2	·	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
1	1	1	·	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	·	1	1	1	1	2	1	1	2	1	2	2	2	2	2	2	2	2

Fig. 4. Visualization of the power generation plant data set generated using hybrid SOM-kMER-PNN algorithm and labeled with fault-states of 1 to 4

The classification capability of the fault-states of the power-generating unit using the proposed hybrid system was also evaluated. Using the test data set, the error rate of SOM-kMER-PNN was found to be lower (2.6% or 26 misclassifications) when compared with that of the Ordered Fuzzy ARTMAP (FAM) model in a previous study, which produced an error rate of (4.3% or 43 misclassifications) [23].

5 Conclusion

In this paper, a hybrid intelligent system comprising SOM, kMER, and PNN has been proposed for data visualization and classification. The visualization map is useful in revealing the underlying data structure. This provides opportunities for humans to utilize their remarkable perceptive and associative abilities to perceive clusters and correlations in data. Indeed, this characteristic is particularly valuable for constructing intelligent systems that require an integration of humans' subjective intelligence with the objective computation of machines in order to attain more accurate interpretation and prediction when dealing with complex and voluminous data.

From the benchmark studies, the classification performance of the proposed hybrid intelligent system is better than the SOM model and is comparable to a variety of existing classifier models. In addition, the practicality of the hybrid model has been demonstrated using a real-world fault detection and diagnosis problem in a power generation plant. Most importantly, the visualization maps generated by the hybrid model have been shown to be useful in revealing the underlying structures of the data sets, as evidenced from the two benchmark case studies as well as the real world fault detection and diagnosis task presented in this paper.

Nevertheless, as SOM batch algorithm is used at the early stage of the proposed hybrid model, this model is employing batch learning. Thus, some interesting future work would be devising an incremental learning system that could adapt to changing environments and comparing the effectiveness of the visualization generated by the hybrid SOM-kMER model with other methods.

References

1. Siedlecki W, Siedlecka K, Sklansky J (1988) An overview of mapping techniques for exploratory pattern analysis. Pattern Recognition 21(5): 411-429
2. Jain AK, Mao J (1992) Artificial neural network for nonlinear projection of multivariate data. Proc. IEEE Int. Joint Conf. Neural Networks 3:335-340

3. Mao J, Jain AK (1993) Discriminant analysis neural networks. Proc IEEE Int. Conf. Neural Networks 1:300-305
4. König A (1998) A survey of methods for multivariate data projection, visualization and interactive analysis. Proc. of the 5th Int. Conf. on Soft Computing and Information/Intelligent Systems, IIZUKA'98, pp. 55-59
5. König A, Michel T (2003) DIPOL-SOM - a distance preserving enhancement of the self-organizing map for dimensionality reduction and multivariate data visualization. Workshop on Self-Organizing Maps - Intelligent Systems and innovational Computing (WSOM&acute'03)
6. Rumelhart DE, Zipser D (1985) Feature discovery by competitive learning. Cognitive Science 9:75-112
7. Bauer H-U, Der R, Herrmann M (1996) Controlling the magnification factor of self-organizing feature maps. Neural Computation 8:757-771
8. Van Hulle MM (1995) Globally-ordered topology-preserving maps achieved with a learning rule performing local weight updates only. Proc. IEEE NNSP95, Cambridge, MA, pp. 95-104
9. Van Hulle MM (1998) Kernel-based equiprobabilistic topographic map formation. Neural Computation 10(7): 1847-1871
10. Lin JK, Grier DG, Cowan JD (1997) Faithful representation of separable distributions. Neural Computation 9: 1305-1320
11. MacIntyre J, Smith P and Harris T (1995). Applications of Hybrid Systems in the Power Industry, in: Medsker LR, Intelligent Hybrid Systems, Kluwer, USA.
12. Medsker LR (1995) Hybrid Intelligent systems. Kluwer, USA.
13. Goonatilake S and Khebbal S (1995). Intelligent Hybrid Systems. John Wiley & Sons, New York.
14. Kohonen T, Somervuo P (2002) How to make large self-organizing maps for nonvectorial data. Neural Networks 15: 945-952
15. Specht DF (1990) Probabilistic neural networks. Neural Networks, 3: 109-118
16. Parzen E (1962) On estimation of a probability density function and mode. Annals of Mathematical Statistics 33: 1065-1076
17. Chen J, Li H, Tang S, Sun J (2002) A SOM-based probabilistic neural network for classification of ship noises. IEEE Int. Conf. On Communication, Circuits and Systems and West Sino Expositions 2:1209-1212
18. Ma F, Wang W, Tsang WW, Tang Z, Xia S, Tong X (1998) Probabilistic segmentation of volume data for visualization using SOM-PNN classifier. IEEE Symposium on Volume Visualization 169:71-78
19. Bishop CM, Svensén M, Williams CKI (1996) GTM: A principled alternative to the self-organizing map. Proc. ICANN'96, pp. 165-170
20. Hoang A (1997) Supervised classifier performance on the UCI database. M.Sc Thesis, Department of Computer Science, University of Adelaide, Australia
21. System Description and Operating Procedures (1999). Prai Power Station Stage 3, 14
22. Tan SC (2001) Classification and rule extraction using artificial neural networks. M.Sc Thesis, School of Electrical & Electronic Engineering, University of Science Malaysia

23. Tan SC, Lim CP (2000) Application of an Adaptive Neural Network to Fault Diagnosis and Condition Monitoring. Proc. of the 5th Online World Conf. On Soft Computing in Industrial Applications, 4 Sept 2000

Evolutionary Multidimensional Scaling for Data Visualization and Classification

Kuncup Iswandy[1] and Andreas König[2]

[1] Institute of Integrated Sensor Systems, TU Kaiserslautern, 67663 Kaiserslautern
kuncup@rhrk.uni-kl.de
[2] koenig@eit.uni-kl.de

Summary. Multidimensional Scaling (MDS) is a well established technique for the projection of high-dimensional data in pattern recognition, data visualization and analysis, as well as scientific and industrial applications. In particular, Sammons Nonlinear Mapping (NLM) as a common MDS instance, computes distance preserving mapping based on gradient descent, which depends on initialization and just can reach the nearest local optimum. Improvement of mapping quality or reduction of mapping error is aspired and can be achieved by more powerful optimization techniques, e.g., stochastic search, successfully applied in our prior work. In this paper, evolutionary optimization adapted to the particular problem and the NLM has been investigated for the same aim, showing the feasibility of the approach and delivering competitive and encouraging results.

1 Introduction

Multidimensional Scaling (MDS) is a common dimensional reduction method and widely used in exploratory data analysis and design of pattern recognition and integrated sensor systems for extraction of essential information from multivariate data sets [1]. Extraction of essential information by dimensionality reduction is required for various reasons. For instance, it can avoid the curse of dimensionality and thus improve the ability of classification. The analysis of unknown data by projection and ensuing interactive visualization is another common and important application of dimensionality reduction [2, 3].

MDS represents data in a smaller number of dimensions preserving the similarity structure of the data as much as possible. One common MDS instance is Sammon's nonlinear mapping (NLM) that will be emphasized and discussed in this paper. In this method, a criterion denoted stress (error) function is defined and optimization takes place by gradient descent techniques [4]. These optimizations have known drawbacks, which strongly depend on the initialization, can get more easily trapped in a less fortunate local minimum and saturate on an undesirable high error value. Implications of this behavior can be data visualizations of low reliability, which incorporate twists and misleading neighborhoods. To overcome this problem, there are some methods that have been proposed, e.g., simulated annealing [5] or using neural network [6]. In similar cases and applications, stochastic search optimization in our prior work has been successfully applied to overcome such a problem [7]. In this paper, we investigated another stochastic method, that draws inspiration from the process of natural evolution, i.e., evolutionary computation. In several research activities, MDS has been applied to extract useful information and visualize the progress of evolutionary optimization [8]. Additionally, some recent publications have introduced the application of genetic algorithm for optimizing MDS itself [9, 10]. Similar to those, we have applied evolutionary optimization. However, we adopted from our work on stochastic search [7] the concept of the mutation operator, rather than employing gradient factor, since this approach pursued in [9] takes significant computational effort. Regarding to the goals of mapping improvement, i.e., the reliability (error reduction) and speed computation, we examine the feasibility and assessment of applicability or competitiveness of such approaches adapted to MDS.

The next section will be briefly summarized the NLM and stochastic search technique according to the previous work, the third section will describe the developed evolutionary optimization for NLM, the fourth section will describe our experiments and results. Finally, the fifth section presents the conclusion.

2 Multidimensional Scaling

In the visualization and data analysis of high dimensional data, a method transform multidimensional data to a lower dimension is needed, preferably to 2 or 3 dimensions due to the human visual limitation to three dimensions. This method is referred to as Multidimensional Scaling (MDS) [1]. This transformation provides a lower dimensional mapping where the dissimilarities (similarities) between the data points of the high dimensional domain corresponds with the dissimilarities of the lower dimensional domain. To measure the dissimilarity (similarity), the distance between pairs of data points is used, for instance the Euclidean distance. MDS is to try to position the feature vectors in the reduced dimension space so that the distance values among them are preserved as much as possible. A measurement has been employed for this purpose, known as stress or error function, that is iteratively minimized. One of the well known methods of multidimensional scaling is Sammon's nonlinear mapping or NLM.

2.1 Sammon's Nonlinear Mapping

The Sammon's nonlinear mapping or NLM is a standard mapping algorithm used for reducing higher-dimensional data space, $\mathbb{R}^p$, to lower-dimensional data space, $\mathbb{R}^d$ (here $d = 2$). The idea is to preserve all interpoint distances in the p-space to the d-space. This can be achieved by minimizing an error or cost function [4]

$$E(m) = \frac{1}{\sum_{i=1}^{N-1}\sum_{j=1+i}^{N} d_{X_{ij}}} \sum_{i=1}^{N-1}\sum_{j=1+i}^{N} \frac{(d_{X_{ij}} - d_{Y_{ij}})^2}{d_{X_{ij}}}, \tag{1}$$

where d_{Xij} is distance between X_i and X_j in $\mathbb{R}^p$, and d_{Yij} is distance between Y_i and Y_j in $\mathbb{R}^d$.

To minimize the cost function, a gradient descent approach is used and the new d-space are determined by

$$Y_{iq}(m+1) = Y_{iq}(m) - MF * \triangle Y_{iq}(m), \tag{2}$$

where

$$\triangle Y_{iq}(m) = \frac{\partial E(m)}{\partial Y_{iq}(m)} \Bigg/ \left| \frac{\partial^2 E(m)}{\partial Y_{iq}(m)^2} \right|. \tag{3}$$

The *Magic Factor* MF serves to scale the gradient steps, $0 < MF \leq 1$ and was determined to be $MF \approx 0.3$ or 0.4. This method has known-drawbacks, which can be tackled by stochastic methods.

2.2 Stochastic Search

In our prior work, a stochastic search technique have been proposed to optimize the error function (1). This technique can be specified into two searching strategies according to the search direction vector [7]. The first strategy, called Global Random Search (GRS), seeks the optimum with moving all the lower-dimensional data together at once and it has a disadvantage, which makes many iterations and effort in computation. The other successful local variant, denoted Local Random Search (LRS), that can reach more competitive results to gradient descent and is faster than GRS in the search for a suiting minimum, seeks the optimum by just moving the points one by one, i.e., when a point is adapted and displaced, the remaining points are frozen. According to the disadvantage of GRS [7], its approach has been dropped and only LRS has been used.

The algorithm generates a minimizing sequence $\mathbf{Y}_i(m), i, m = 1, 2...$ as:

$$\mathbf{Y}_i(m+1) = \mathbf{Y}_i(m) + c(m) \times L_{\mathbf{Y}_i(m)} \times \mathbf{Vn}_i, \tag{4}$$

where c is step size, $L_{\mathbf{Y}}$ is the length of vector $\mathbf{Y}(m)$ and $\mathbf{Vn}$ is a random search direction (normalized-vector).

This algorithm runs in an iterative way and the error value of each moved point is evaluated by its previous value. If the moved point has smaller error than old one, then the new moved point will be saved, otherwise keep the old point. When no any points in one iterative get improving then the step size c will be decreased by multiplying 0.5, otherwise keep the step size.

3 Evolutionary Multidimensional Scaling

The cost function has been optimized previously by using gradient and stochastic search methods. The main characteristic of evolutionary algorithm is the intensive use of randomness and genetic-inspired operations, e.g., selection, recombination and mutation to evolve a set of candidate solutions (chromosomes). The chromosomes are represented as real-valued vectors and each chromosome corresponds to a solution of points in the low dimensional space. The quality function (1) is used as a fitness measure - the lower the better. To attain our target regarding to the mapping quality and speed computation, the concept of mutation operator has been adopted from stochastic search. The main steps of the evolutionary Multidimensional Scaling are described as follows:

1. *Initialization*: Generate an initial population using a random mechanism. Denote the population as $\mathbf{P}_i = \{\mathbf{p}_1, ..., \mathbf{p}_N\}$, where N is the population size and $\mathbf{p} = \{\mathbf{Y}_1, ..., \mathbf{Y}_n\}$, where n is data size.
2. *Selection for recombination*: Roulette Wheel Selection is used to select candidates into mating pool and the selection probability of each individual is based on its proportional fitness value. Selected candidates represent an intermediate population, $\mathbf{P}_s$, that has same size with current population.
3. *Recombination*: Here, discrete recombination takes place for search process and information exchange between two chromosomes from parent population and this operator is one of recombination types in evolution strategies used for recombining two floating-point strings [11]. This generates a next intermediate population $\mathbf{P}_r$ from $\mathbf{P}_s$, where each two individuals take the recombination process and produce two offsprings.
4. *Mutation*: After recombination process, the mutation generates the next intermediate population $\mathbf{P}_m$. Each individual from $\mathbf{P}_r$ produce an offspring. The mutation procedure is adapted and works similarly as stochastic search technique (4). This mutates one by one pattern $\mathbf{Y}_i$ in one individual $\mathbf{p}_k$.

   $$\mathbf{Y}_i(m+1) = \mathbf{Y}_i(m) + s \times D_i \times \mathbf{Vn}, \tag{5}$$

   where s is step size, $\mathbf{Vn}$ is a random search direction (normalized-vector) and

   $$D_i = \frac{1}{n-1} \sum_{j=1, i \neq j}^{n} |d_{X_{ij}} - d_{Y_{ij}}|, \tag{6}$$

   The step size s will be decreased by 0.9, if the percentage of patterns getting improvement in one generation is less than 10%.
5. *Reproduction*: Offsprings from $\mathbf{P}_m$ will be selected into the next generation. The best 10% individuals of parent will be taken and competed with offsprings based on their fitness values. If the parent is better than the

offspring, then the parent will be reproduced for next generation $\mathbf{P}_{i+1}$ and otherwise, they will be discarded.

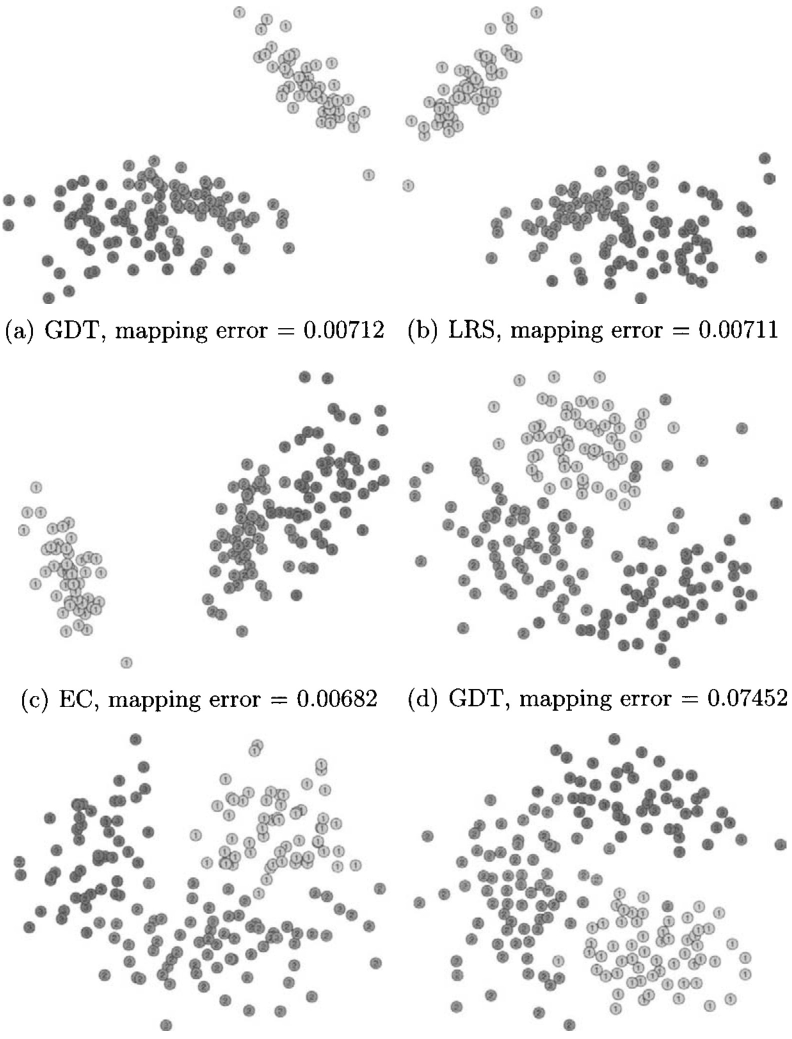
6. *Termination condition*: The termination condition of this algorithm is referred to a pre-specified number of generation and criterion of fitness value. The algorithm will stop, if one of these two conditions are reached; otherwise go to step 2.

4 Experiments and Results

The experiments use three types of data sets, namely Iris, Wine and Mechatronic data for evaluating the optimization algorithms [7]. The maximum iteration or generation is bounded as much as 100 for each algorithm. The parameters of optimization algorithm will be set same for all data sets during the experiments. For a population in this evolutionary optimization, we used 20 individuals. A brief description of each data set and the results will be discussed as follows:

1. *Iris Data*: This data set is well-known, since it was used by many researchers to demonstrate the proposed mapping technique. The data consist of 4 features and 150 patterns. The features are petal and sepal width of the iris flowers, and the three classes correspond to species setosa, virginia and versicolor.
2. *Wine Data*: The data is available from a machine learning repository (www.ics.uci.edu/~mlearn/MLRepository.html) and is also standard data commonly used. The data serves the purpose to distinguish three types of wines, which correspond to the classes, based on 13 features. The continuous attributes are in the order of presentation Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline. A total of 178 patterns are included, where 59 are affiliated to class 1, 71 are affiliated to class 2, and 48 are affiliated to class 3.
3. *Mechatronic Data*: This data set comes from the domain of mechatronics, where the compressor of a turbine jet engine was monitored with regard to its operating range [2]. The 24-dimensional data, which has also high intrinsic dimension, was generated from Fourier spectra obtained from the compressor. Four operating regions were defined as classes. In summary, 1775 samples were drawn from a compressor setup and recorded as a data set including attribute values. The objective of the underlying research was the development of a stall margin indicator for optimum jet engine operation. This medium-sized data set provides the first realistic application of the proposed methods.

According to the stochastic nature of the applied methods, we carried out extensive simulations for three types of data sets. In our investigation, ten different, independently generated random initialization as starting points for



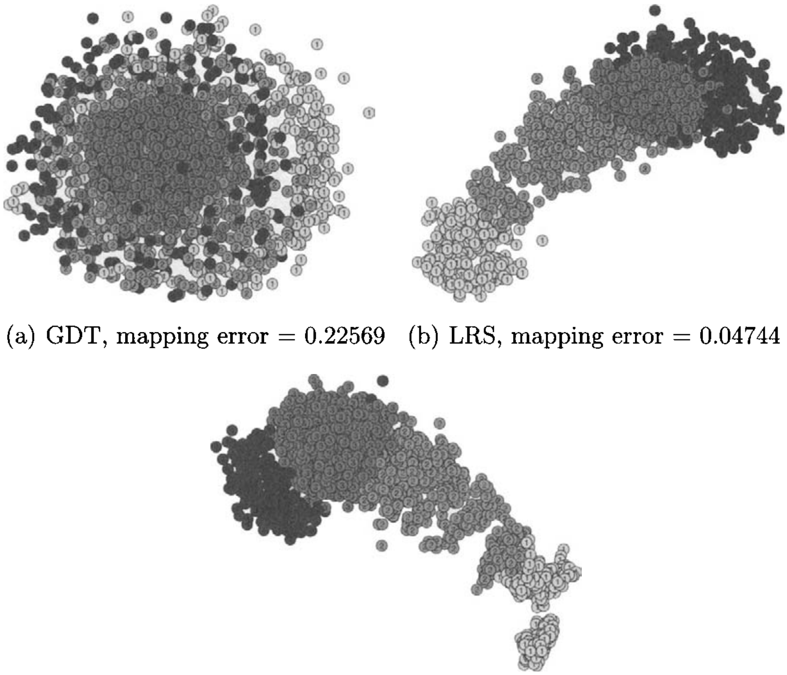
(a) GDT, mapping error = 0.00712 (b) LRS, mapping error = 0.00711

(c) EC, mapping error = 0.00682 (d) GDT, mapping error = 0.07452

(e) LRS, mapping error = 0.05752 (f) EC, mapping error = 0.05863

Fig. 1. Visualization 2D of Iris data (a)-(c) and Wine Data (d)-(f)

GDT, LRS and EC. Since the LRS depends additionally on random mechanism in its searching directions, we carried out simulations based on ten different random seeds for a balanced comparison. The corresponding results of ten runs are summarized in each row by mean and best error value for LRS, except Mechatronic data. The results are given in three tables in ten different rows. Here, we have extended the results of previous work and put the new results into perspective in each of the tables. In Table 1, 2 and 3 the obtained results show that EC optimization in all cases is more robust than LRS and



(a) GDT, mapping error = 0.22569 (b) LRS, mapping error = 0.04744

(c) EC, mapping error = 0.02794

Fig. 2. Visualization 2D of Mechatronic data

Table 1. Mapping Quality of GDT, LRS and EC for Iris Data

initial no.	E_{GDT}	E_{LRS} (mean / best)	E_{EC}
1.	0.01207	0.00833 / 0.00717	0.00682
2.	0.01632	0.01011 / 0.00689	0.00673
3.	0.01380	0.00946 / 0.00687	0.00670
4.	0.01064	0.00888 / 0.00682	0.00676
5.	0.00983	0.00834 / 0.00712	0.00671
6.	0.01189	0.00817 / 0.00682	0.00676
7.	0.01132	0.00832 / 0.00711	0.00682
8.	0.01113	0.00816 / 0.00719	0.00670
9.	0.00712	0.00826 / 0.00690	0.00686
10.	0.01083	0.00780 / 0.00684	0.00675
mean	0.01149	0.00858	0.00676
σ	0.00241	0.00070	0.00005

superior to the LRS and GDT, especially for large dimension and data size. The projection results achieved by GDT, LRS and EC for three types of data set are shown in Fig. 1 and 2.

Table 2. Mapping Results of GDT, LRS and EC for Wine Data

initial no.	E_{GDT}	E_{LRS} (mean / best)	E_{EC}
1.	0.07721	0.05982 / 0.05775	0.05878
2.	0.07495	0.06048 / 0.05925	0.05966
3.	0.07452	0.05987 / 0.05764	0.05785
4.	0.07871	0.06158 / 0.05810	0.05760
5.	0.06472	0.05891 / 0.05752	0.05863
6.	0.07610	0.06187 / 0.05834	0.05904
7.	0.07799	0.05822 / 0.05793	0.05818
8.	0.06623	0.05902 / 0.05790	0.05874
9.	0.06510	0.06045 / 0.05846	0.05764
10.	0.07871	0.05855 / 0.05765	0.05912
mean	0.07342	0.05987	0.05852
σ	0.00575	0.00123	0.00068

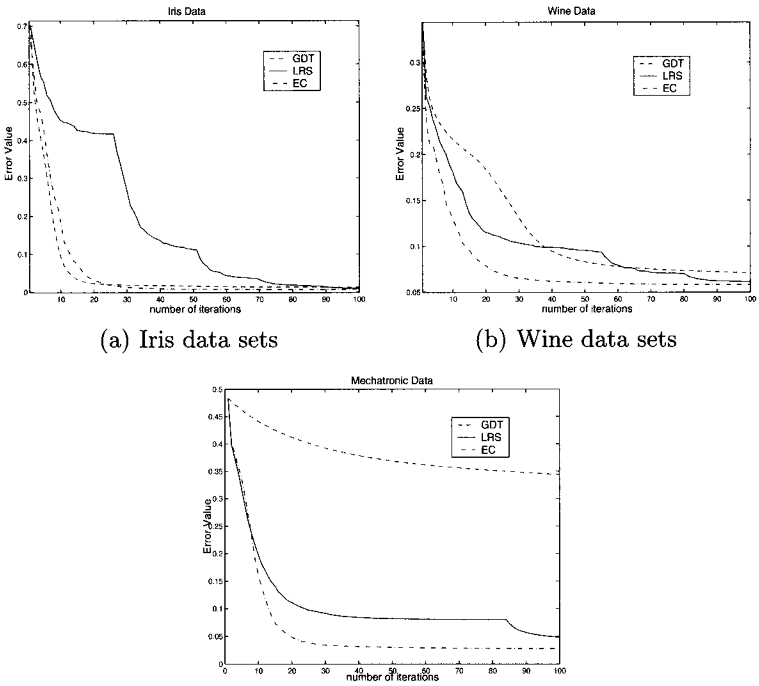
Table 3. Mapping Results of GDT, LRS and EC for Mechatronic Data

initial no.	E_{GDT}	E_{LRS}	E_{EC}
1.	0.24241	0.07100	0.02902
2.	0.30169	0.07780	0.02856
3.	0.31531	0.07100	0.02965
4.	0.27968	0.07013	0.02821
5.	0.28320	0.05145	0.02753
6.	0.33068	0.05143	0.02836
7.	0.29981	0.06624	0.02865
8.	0.27665	0.06849	0.02745
9.	0.22569	0.04804	0.02794
10.	0.33963	0.04744	0.02855
mean	0.28944	0.06230	0.02839
σ	0.03613	0.01138	0.00066

Table 4. The attained Goals of GDT, LRS and EC

	EC	GDT	LRS
speed	-	+	+
error	++	-	+

Figure 3(a) - 3(c) show that the EC's error curves can reach more lower error values than both LRS and GDT. Also, EC optimization needs only less than 100 iterations to achieve good solution for three types of data sets. Extending the number of iterations up to 500 did not give any more improvement for EC. Expending more time and iterations can improve GDT and LRS



(a) Iris data sets

(b) Wine data sets

(c) Mechatronic data sets

Fig. 3. Error curves of GDT, LRS and EC

results. However, GDT curve saturates (fall to less suited local minimum) in many cases. Although the obtained results show that EC optimization is superior to LRS and GDT, the evolutionary optimization needs large computational effort in computing for one whole generation. The EC optimization for Iris data set requires 183 seconds/100 iterations in average, whereas GDT and LRS need respectively 16 and 22 seconds/100 iterations.

5 Conclusions

In this paper, we have investigated the combination of evolutionary optimization with dedicated mutation operator derived from prior work on Stochastic Search for multidimensional scaling, in particular Sammon's distance preserving nonlinear mapping, with regard to the goals of mapping improvement in terms of error reduction and/or speed improvement. The obtained EC results have been compared with previous work and optimization techniques employing representative standard data sets. With regard to mapping quality (error), the investigated method scaled well with the data size and proved to be competitive or even superior to existing techniques. Although there is no improvement for the computational speed with regard to our own prior work, the introduced mutation operator appears to be much simpler to implement than state of the art published methods. In any case our method is in particular beneficial, when reliable mappings are needed.

The focus of the work in this paper was on mapping high dimensional data sets for visualization purposes. But, similar to the work reported in [2], the evolutionary concept can straightforward be adapted to the NLM recall, that allows both hierarchical mapping for large data sets as well as separate mapping of training, test, and application data and for classification purposes. Further, the evolutionary multidimensional scaling could be applied to extract and visualize data produced by evolutionary algorithms themselves [8].

References

1. J. B. Kruskal and M. Wish. Multidimensional Scaling. Sage University Paper Series on Quatitative Application in the Social Sciences, Sage Publications, Beverly Hills, no. 07-011, 1978.
2. A. König. Interactive visualization and analysis of hierarchical neural projections for data mining. *IEEE Trans. on Neural Networks*, 11(3):615–624, January 2000.
3. J. W. Sammon. Interactive pattern analysis and classification. *IEEE Trans. on Computers*, C-19(7):594-616, July 1970.
4. J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. on Computers*, C-18(5):401-409, May 1969.
5. H. Klock and J. M. Buhmann. Multidimensional Scaling by deterministic annealing. In M. Pellilo and E. R. Hancock, editors, *Proceedings EMMCVPR'97*, volume 1223 of Lecture Notes in Computer Science, pages 401-409, Springer Verlag, 1997.
6. J. C. Mao and A. K. Jain. Artificial Neural Networks for Feature Extraction and Multivariate Data Projection. *IEEE Trans. on Neural Networks*, Vol. 6(2):296-317, March 1995.
7. K. Iswandy and A. König. Improvement of Non-Linear Mapping Computation for Dimensionality Reduction in Data Visualization and Classification. In *Proc. of the 4th Int. Conf. on Hybrid Intelligent Systems HIS04*, IEEE CS, pages 260-265, Kitakyushu, Japan, 5-8 Dec., 2004.
8. H. Pohlheim. Visualization of Evolutionary Algorithms - Set of Standard Techniques and Multidimensional Visualization. In Banzhaf, W., eds.: *GECCO'99 - Proc. of the Genentic and Evolutionary Computation Conference*, San Fransisco, CA, Morgan Kaufmann, pages 533-540, 1999.
9. S. Etschberger and A. Hilbert. Multidimensional Scaling and Genetic Algorithm: A Solution Approach to Avoid Local Minima. In: Arbeitspapiere zur mathematischen Wirtschaftsforschung 181, 2002.
10. L. D. Chambers. The Practical Handbook of Genetic Algorithms: Applications, Second Edition. CRC Press, 2000.
11. T. Bäck, U. Hammel and H.-P. Schwefel. Evolutionary Computation: Comments on the History and Current State. *IEEE Trans. on Evolutionary Computation*, Vol. 1(1):3-17, April 1997.

Extended Genetic Algorithm for Tuning a Multiple Classifier System

Ramon Soto C. and Julio Waissman V.

Center for Research on Information Technologies and Systems Autonomous University of Hidalgo State, Pachuca, Hgo., Mexico rsotocc@hotmail.com

Summary. A widely accepted idea in the pattern recognition field is that a multiple classifier system use to show superior performance than individual classifiers when dealing with complex problems. Most multiple classier systems are built up from classifiers developed completely independent of each other and combined in a last step, generating possible decisions conflicts among individual classifiers. In this paper, a non standard genetic algorithm for tuning multiple classifier systems is presented. This algorithm is based on a set of concepts that extends the genetic metaphor: coevolutionary diversity, collective fitness, suitable behavior, phylogenetic evolution and ontogenetic evolution.

Keywords: Genetic algorithms, multiple classifier systems, ontogenetic evolution.

1 Introduction

A widely accepted idea in the pattern recognition field is that a multiple classifier system (MCS) use to show superior performance than individual classifiers when dealing with complex problems.

The main problems to be solved when developing a multiple classifier system are: 1) the procedure used to choice the best individual models; and 2) the way to combine the individuals models in a whole system. Most multiple classier systems are built up from classifiers developed completely independent of each other and combined in a last step, generating possible decisions conflicts among individual classifiers. Finding the best model for a system whose dynamics is incompletely known is a hard task due to the combinatorial explosion problem that arises when the number of variables grows up.

Genetic algorithms (GA) are a useful and straightforward method for solving hard optimization problems, which have been used successfully for training diverse types of classifiers. But must genetic algorithms work is based on the assumption of homogeneous populations.

In this paper, a non standard genetic algorithm for tuning multiple classifier systems is presented. This algorithm is based on a set of concepts that extends the genetic metaphor beyond the standard genetic approach: coevolutionary diversity, collective fitness, suitable behavior, phylogenetic evolution and ontogenetic evolution.

2 Basic Genetic Concepts

The importance of evolution as a way to improve the potential success of the species in nature is a broadly accepted idea in biology. Along several generations, the natural populations evolve according to the principles of natural selection and survival of the most skilful. Individuals compete to each other for resources like food, water and refuge. Similarly, they compete to attract and to mate the most outstanding individuals in the population. Individuals with more success to survive and to mate will have more numerous descendants, whereas individuals with low performance will have scarce or any descendant. The combination of good characteristics of different predecessors is expected to produce descendants with better aptitudes that any of its parents. This way, the species evolve to become better adapted to their environments.

The idea of applying the evolution concepts to an artificial system led to the evolutionary computing paradigm. The evolutionary computing approach includes different techniques for search and optimization problems based on diverse analogies with the evolutionary processes in nature. The main research lines in this field are those of genetic algorithms, evolution strategies and evolutionary programming.

Genetic algorithms are the evolutionary technique that has received the largest attention from the artificial intelligence community. The genetic algorithms try to develop appropriate solutions for computational problems imitating the genetic processes used by nature for evolving living organisms, with a high degree of adaptation to their environment. Potential solutions to the problem play the role of individuals in a population, while the environment is modeled by a fitness function that determines the quality of each candidate solution. The individuals with high aptitudes receive larger opportunities to reproduce by means of crossover with other individuals of the population. This produces new individuals that share some characteristics taken from each parent. A new population of possible solutions takes place when the best individuals in the actual generation are selected for transferring their characteristics to the new population throughout new individuals that replace the worse individuals of the actual generation. The process is repeated until an organism is produced which gives a solution that is good enough.

By favoring the mating and surviving of the most capable individuals, it is possible to explore the most promising search regions without having to carry out an exhaustive search. The population will converge to a good solution for the problem provided the genetic algorithm has been well designed.

3 Extended Genetic Concepts

The genetic programming algorithm presented in this paper is based on the following set of concepts: coevolutionary diversity, collective fitness, suitable behavior, phylogenetic evolution and ontogenetic evolution. These concepts extend the genetic metaphor beyond the standard genetic algorithms approach.

3.1 Coevolutionary Diversity

Most applications in evolutionary computation suppose certain population homogeneity along the complete evolutionary process. In this way, all individuals are selected for evolution at the same time. But diversity is one of most desirable feature of a multiple classifier system, as the variety of models improves the possibility of obtaining a good solution [4, 10]. However, such heterogeneity also makes difficult to obtain the collective model [2]. While some models, like k-nearest neighbors classifiers, have simple configuration procedures, models like neural networks require of large training processes. The differences in the tuning complexity represent a form of coevolutionary diversity [12].

In order of undertake the problem of selection of individuals, granting enough time for tuning complex models, but avoiding to maintain simple models in an idle state, three periods of evaluation are used, that are defined as follows:

Team. The individual classifiers must show a good performance in solving a given problem while they collaborate in diverse teams throughout his life cycle. Along the training process, different teams of classifiers are formed from the entire population. The performance of the classifier is recorded while they participate in each one of the work teams in which it is assigned.

Generation. This is the period of time throughout a fixed set of individuals of a given species lives, cohabits and competes by the resources available. This concept describes the basic change in the composition of the population of the MCS. At the end of each generation a new population is generated, based on the performance shown by the agents in the present population throughout the diverse work teams in which they have participated. The generations of diverse species can last different.

Age. This concept describes a global change in the population of a MCS, involving generational changes on all the species in the system.

These concepts describe and emulates the evolutionary diversity shown by nature. Diverse species in a natural ecosystem show diverse rates of evolution, depending on their organic complexity and the natural resources available.

3.2 Advisable Behavior and Collective Fitness Evaluation

Two goals of the method for design of multiple classifier systems, developed in this work are: 1) to simulate a complex dynamic system by a combination of simple models, and 2) to have the capacity of replacing on-line those models which are no longer useful. In order to achieve the first goal, a genetic programming process, based on the concept of collective fitness evaluation is used. The performance of each classifier is partially evaluated in terms of its impact in the global prediction, so that, very simple and even erroneous individual models can be considered useful if they improve the performance of the MCS. The second goal is reached by conserving those agents which have shown the best average performance after collaborating in diverse teams.

Due to the strategy of collective evaluation, it is expected that only the collective model would be appropriated for the description of the physical system. Therefore, the optimal solution looked for by the genetic programming algorithm consists of a collection of individual models with the capacity to collaborate in the solution of a given class of problems, instead of an individual able to solve the complete problem by itself. The search of the optimal global model corresponds to selecting those classifiers who, when working in different teams, lead to a good collective performance.

Definition 1. *The behavior of a classifier is advisable if and only if such behavior leads to improve the performance of the species.*

Definition 2. *A collective evaluation of aptitude is a process of aptitude evaluation that favors to the classifiers with an advisable behavior.*

The aptitude of the classifier A_j is evaluated from two error measures: the direct error (e_j), that describes the individual quality of the model and the induced error ($\tilde{e}_j$), that describes its ability to collaborate in the collective prediction. These measures of error are defined as:

$$e_j = \frac{1}{N_T} \sum_{n=0}^{N_T} \min \left(\left| \frac{O_n - \hat{O}_{nj}}{d} \right|, 1 \right) \tag{1}$$

and

$$\tilde{e}_j = \frac{1}{N_T} \sum_{n=0}^{N_T} \left[\min \left(\left| \frac{O_n - \hat{O}_n}{d} \right|, 1 \right) - \min \left(\left| \frac{O_n - \hat{O}_n^{(-j)}}{d} \right|, 1 \right) \right] \tag{2}$$

being N_T the number of prediction assignments in which the classifier has taken part; O_n is the goal value in the n^{th} prediction task; $\hat{O}_{nj}$ is the prediction made by the classifier A_j; $\hat{O}_n$ is the collective prediction for the same objective value and $\hat{O}_n^{(-j)}$ is the collective prediction of O_n when the opinion of A_j is discarded. d is defined as:

$$d = \begin{cases} O_n \text{ if } O_n \neq 0 \\ 1 \quad \text{a.o.c.} \end{cases} \tag{3}$$

The global error (E_j) for each classifier A_j is computed as:

$$E_j = \begin{cases} e_j \cdot \tilde{e}_j & \text{if } \tilde{e}_j > 0, e_j \neq 0 \\ \tilde{e}_j / e_j & \text{if } \tilde{e}_j < 0, e_j \neq 0 \\ \min(\tilde{e}_j, e_j) & \text{a.o.c.} \end{cases} \tag{4}$$

The best individuals in each generation are those which accumulate the less global error.

These concepts highlight the importance of the context for evaluating the fitness of individuals. Due to this approach to fitness evaluation, the organization of the global population also changes throughout the genetic process.

3.3 Phylogenetic Evolution v.s. Ontogenetic Evolution

The main limitation of genetic algorithms (and the other evolutionary computing approaches) is the use of a very narrow interpretation of the concept of evolution. In a broad sense, evolution refers to any process of change over time. When related to living beings, evolution has a meaning that is closely tied to that of adaptation: the main goal of evolution is to get individuals more adapted to their environment.

Nature uses diverse adaptation mechanisms that differ in time and space, but which are complementary for improving the success expectations of individuals [3, 9, 8]:

- The first evolution mechanism takes place at phylogenetic level. At the fertilization, the chromosomes of the parents merge to create a new genotype which comprises genetic information from its antecessors. In this stage, the new genotype can experiment several mutations, giving place for a further genetic change: Along many generations phylogenetic evolution produces new individuals that are expected to be more and more apt to interact with their environment than their predecessors were. Due to this process, the structure of the individuals of a species is gradually modified [3, 8], being the species the main beneficiary of this kind of adaptation: This idea is the basis of standard genetic algorithms
- The second mechanism of evolution occurs at the ontogenetic level. Following the fertilization, the zygote undergoes a complex process, called embryogenesis, to become a mature phenotype, involving several steps of multiplication, growth, movement and differentiation of the cells. This process has inspired the so called artificial embryogenesis [1, 15].
- The third mechanism of adaptation is learning, which occurs also at the ontogenetic level and that consists in the evolution that each individual experiences throughout its life cycle [11, 5, 1, 3, 8]. After the birth, the individual must acquire and adjust a set of abilities in order of being

able to survive in its environment. Learning has important consequences in improving the potential success of an individual. In this case, the main beneficiary of the adaptation process is the individual and only a very small part of these learned resources is inherited to the following generations. This concept has been widely used in automatic learning [5], specially in the scope of artificial neuronal networks.

Nevertheless, and although diverse authors have been outstand the importance and complementariness of these adaptation mechanisms, very few experiments to explicitly combine such evolutionary concepts have been realized [3].

3.4 Extended Genetic Algorithm

The concepts presented in the previous sections can be used for outline an extended genetic algorithm, whose scheme is shown in Sect. 1, Figs. 1 and 2.

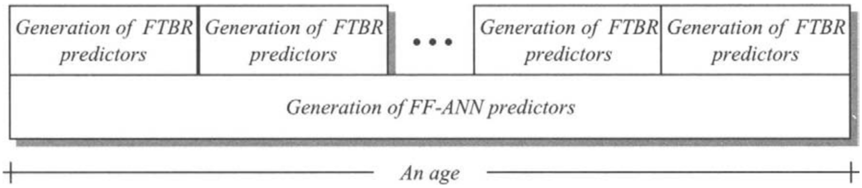

Generation of FTBR predictors
Generation of FTBR predictors
• • •
Generation of FTBR predictors
Generation of FTBR predictors
Generation of FF-ANN predictors
An age

Fig. 1. Relation between the duration of generation and Age.

The extended genetic algorithm starts by generating an initial hybrid population, composed by classifiers from different species. Next, parallel evolutionary processes at generational level for each species are carried out. Independent classifier teams from each classifier type subpopulation are gathered. Each classifier participating in a team is subject of a training process, which is specific of the classifier species. Then, the whole team is evaluated on the basis of a collective task. A set of such collective training-evaluation tasks are performed, in order of assure that each classifier has participated in several teams along a generation. At the end of a generation, a change on the classifier type subpopulation is realized and a new generational cycle is initialized. An age comprises one or several generations for each classifier type. If a classifier type has a complex learning process, like backpropagation algorithm for neural networks, the age comprises as few as just one generation. If the classifier training is very simple, like that of k-means, then several generations take place throughout an age. This variability allows spending large training time for complex classifiers, while training and testing large sets of simple classifiers.

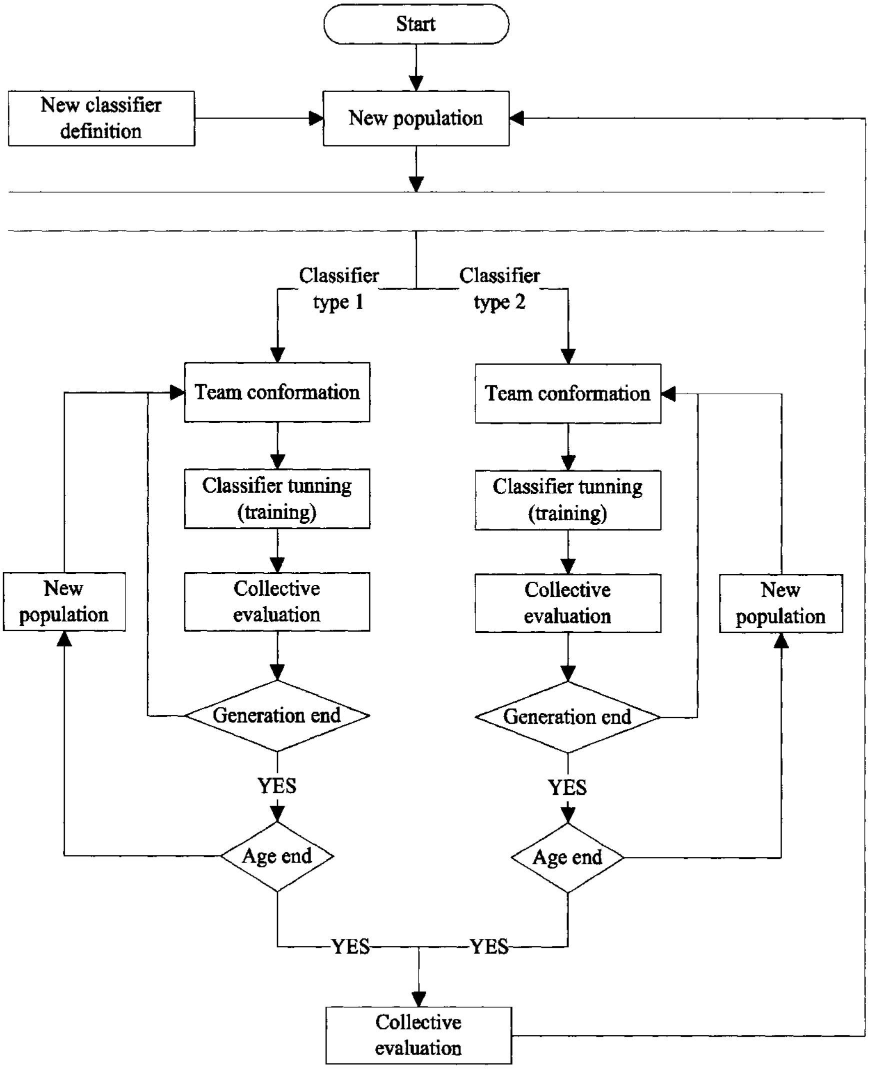


Fig. 2. Extended genetic algorithm.

At any time along the genetic process, new classifier types can be added to the system.

4 Results

In order of test the extended genetic algorithm, prediction in the Foreign Exchange Rates market was addressed. The variables to be predicted in this study case are the exchange rates related to the U.S. dollar of the next currencies: the Pound Sterling, the Canadian Dollar, the German

Mark, the Japanese Yen and the Swiss Franc. The data used in this experiment were taken originally from the Monetary Yearbook of the Chicago Mercantile Exchange, and published in Internet by Andreas S. Weigend (http://www.cs.colorado.edu/~andreas/Time-Series/Data/). This database contains 3512 records covering the time period from June 1st, 1973 to May 21st, 1987.

A time series database consisting of 3384 records was used; the first 3000 records were used as knowledge base while the next 256 records were used as target points in the training. The last 128 points are used as target points in a final test phase.

To assure reproducibility, 4 different MCS were trained for solving this problem. The prediction techniques used in this experiment were: a fuzzy KNN-type classifier called Fuzzy Time-series Based Reasoning (FTBR) [13], the standard feedforward neural network trained with the backpropagation algorithm (BNN) [16] and a fuzzy variant of backpropagation neural network (FBNN) [14]. The population in each MCS consisted of 100 FTBR predictors, 15 BNN predictors and 15 FBNN predictors.

The available data to the predictors during each prediction task were just the cases base, and the exact input values for the first prediction step. The predicted values are added to the cases base and used as input in later prediction steps.

The performance of the MCS is evaluated in terms of the error ratio when predicting the change tendency in the 5 variables considered in the experiment. As can be seen from Table 1, the results are similar o superior to those obtained with general techniques, and even comparable to those based on specific information (see, for example, [6, 7, 17]).

Table 1. Error ratio of 4 multiple classifier systems when predicting the change tendency of 5 variables in foreign exchange market.

	Output variable				
MCS	Pound Sterling	Canadian Dollar	German Mark	Japanese Yen	Swiss Franc
1	63.49206	63.49206	58.73016	60.31746	58.73016
2	66.66666	60.31746	47.61905	58.73016	53.96825
3	63.49206	52.38095	58.73016	65.07937	57.14285
4	79.36508	76.19047	76.19047	73.01588	77.77778
Prom	68.25397	63.09523	60.31746	64.28571	61.90476

5 Conclusions

An extended genetic algorithm has been presented, based on the concepts of coevolutionary diversity, collective fitness, suitable behavior, phylogenetic evolution and ontogenetic evolution.

The core concepts of this algorithm allows to configure heterogeneous multiple classifier systems. The results on the foreign exchange market shows a good performance of the MCS generated by the extended genetic algorithm.

As can be seen from Table 1, the results are similar o superior to those obtained with general techniques.

References

1. J. Bongard and R. Pfeifer. Evolving complete agents using artificial ontogeny. In *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258. Springer-Verlag, Berlin, 2003.
2. R.P.W. Duin. The combining classifier: To train or not to train? In C. Suen R. Kasturi, D. Laurendeau, editor, *Proceedings 16th International Conference on Pattern Recognition ICPR16*, volume II, pages 765–771, Los Alamitos, 2002. IEEE Computer Society Press.
3. D. Floreano and F. Mondada. Evolution of plastic neurocontrollers for situated agents. In P. Maes, M. Mataric, J. Meyer, J. Pollack, H. Roitblat, and S. Wilson, editors, *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cambridge, MA., 1996. MIT Press-Bradford Books.
4. V. Gorodetski, O. Karsayev, and V. Samoilov. Multi-agent data fusion systems: Design and implementation issues. In *Proceedings of the 10th International Conference on Telecommunication Systems - Modeling and Analysis, Monterey, CA*, pages 762–774, 2002.
5. D. Harter. Ontogenetic development of skills, strategies and goals for autonomously behaving systems. In *Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001)*, pages 178–181, Orlando, FL, 2001.
6. S. Lawrence, A. Chung-Tsoi, and L. Lee-Giles. Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference. Technical Report UMIACS-9627, University of Maryland, 1996.
7. L. Lee-Giles, S. Lawrence, and A. Chung-Tsoi. Rule inference for finantial prediction using recurrent neural networks. In *Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering*, 1997.
8. S. Nolfi and D. Floreano. Learning and evolution. *Autonomous Robots*, 7(1), 1999.
9. S. Nolfi and D. Parisi. Learning to adapt to changing environments in evolving neural networks. *Adaptive Behavior*, 5(1):75–98, 1997.
10. J. Ortega, M. Koppel, and S. Argamon. Arbitrating among competing classifiers using learned referees. *Knowledge and Information Systems*, 3(4), 2001.
11. J. Piaget. *The psychology of intelligence.* New York: Routledge, 2001.
12. M. Ridley. *Evolution, 3rd edition.* Blackwell Publishers, 2003.
13. R. Soto and G. Nez. Soft modelling of financial time series. In *IASTED International Conference on Modelling and Simulation (MS 2003)*, 2003.
14. R. Soto and J. Waissman. Fuzzy backpropagation neural networks for nonstationary data prediction. Technical report, Universidad Autnoma del Estado de Hidalgo, 2005.

15. K. O. Stanley and R. Miikkuleinen. A taxonomy for artificial embryogeny. *Artificial Life*, 9:93–130, 2003.
16. A. Weigend, B. Huberman, and D. Rumelhart. Predicting sunspots and exchange rates with connectionist networks. In S. Eubank and M. Casdagli, editors, *Proceedings of the 1990 NATO Workshop on Nonlinear Modeling and Forecasting*. Addison-Wesley, 1991.
17. J. T. Yao, H. L. Poh, and T. Jasic. Foreign exchange rates forecasting with neural networks. In *International Conference on Neural Information Processing*, pages 754–759, 1996.

Development of Fuzzy Expert System for Customer and Service Advisor Categorisation within Contact Centre Environment

Satya Shah, Rajkumar Roy and Ashutosh Tiwari

{s.shah, r.roy, a.tiwari}@cranfield.ac.uk. Enterprise Integration, Cranfield University, Cranfield. Bedfordshire. MK43 0AL UK

Abstract. In this paper, we describe the research and development of a fuzzy expert system methodology for categorising customer and customer service advisor (CSA) within customer contact centre (CCC) environment. On the basis of data collected through case studies carried out within customer contact centre, two step clustering analysis within SPSS was used to derive the categories for customers and advisors based on demographic, experience, business value and behavioural attributes. The fuzzy expert system assigns a new customer or advisor to the pre-defined categories and provides the corresponding membership values given into the system using fuzzy logic. The author has explained the steps which were followed for the development of the fuzzy expert system. A prototype system has been designed and developed to identify the type of customer and CSA based on the demographic, experience and behavioural attributes. Experimental results are provided and the methodology is validated within the case study approach.

1. Introduction

In a time of fast growing technology and communication systems, it is very important for the industry and the corporations to develop new customer contact centre (CCC) environment technologies for better customer contact requirements. The integration of customer contact centre into day-to-day organisational operations represents one of the most promising trends in the 21st century economy. The impact is such that contact centres are expected to affect almost all aspects of society from the private sector to public sector in all parts of the world. Whatever the nature or point of contact, customers want a seamless interaction throughout their experience with the company. Customers want to contact companies at their convenience, using the most convenient means. The companies are aware that it is easier to lose a customer than to gain one; it also knows that it is easier to sell additional services and products to customers who are satisfied with the service provided upon contact; and that a minority of the customer base accounts for the majority of an organisation revenues. For identifying the type of information required, we needed to develop a fuzzy expert system which would identify the type of customer and advisor based on the demographic, experience, business value and behavioural attributes.

2. Related Research

Understanding and adapting to changes of customer behaviour is an important aspect of surviving in a continuously changing environment [Chiu et al. 2002]. Research in understanding customer preferences, known as 'consumer behaviour study', has been the subject of investigation in psychological marketing area for few decades. It is necessary to understand individual customers from designer side, as well as from the customer's side to provide guidance for customers to find what they want. Customer choice of a product depends on explicit requirements, implicit requirements, available options and latent requirements implied by the product [Zeelenberg and Pieters. 2004]. Studies have also shown that complete understanding of service advisor satisfaction requires knowledge of the customers situation before the communication begins [Heckman and Guskey, 1998]. Although research has suggested that customer service advisor (CSA's) performance is critical to create customer satisfaction, little has been done to analyse which employee behaviours influence customer encounter satisfaction and which behaviours influence relationship satisfaction. There are five dimensions of CSA's behaviour that influence customer's perceptions: mutual understanding, authenticity, extra attention, competence, and meeting minimum standards [Dolen et al. 2004]. Modeling the users can include statements of how the users within a specific user group behave in certain situations or perform certain functions. A system can be designed to accommodate the behavioural diversity of the user groups that most strongly contribute to meeting business goals [Bushey et.al 1999]. The technological change is not simply transforming the methods by which the organisation operates, but is impacting the level of skill and education required by both CSA's and management within the contact centre environment. Many telecoms service sector are subjected to failures in service delivery and better customer satisfaction values because they much depend on customer service advisor (advisors) to deliver service to their customers. Because of the delivery of the service occurs during the interaction between contact advisors and customers, the attitudes and behaviours of advisors can influence customer's perceptions of the services [Hartline and Ferrell, 1996].

Soft computing differs from hard (conventional) computing in that it is tolerant of imprecision, uncertainty and partial truth [Zadeh, 1996]. Soft computing technologies provide an approximate solution to all ill-defined problems and can create user models in an environment, such as behaviour modeling, in which users are not willing to give feedback on their actions and/or designers are not able to fully define all possible interactions [Martinez et.al 2004]. Fuzzy logic has proved useful for developing many practical applications, especially in the field of engineering, as it can handle inexact and vague information. Soft computing technologies provide an approximate solution to an ill-defined problem and can create user models in an environment, such as contact centre environment. The elements that a user model captures (goals, plans, preferences, common characteristics of users) can exploit the ability of soft computing of mixing different behaviour and capturing human decision processes. Fuzzy logic provides a mechanism to mimic human decision making that can be used to infer goals and plans [Martinez et.al 2004]. Mamdani-type inference, as defined it for the Fuzzy Logic Toolbox, expects the output membership functions to be fuzzy sets. Sugeno-type systems support this type of model. In general, Sugeno-type systems can be used to model any inference system in which the output membership functions are either linear or constant [Mathworks, 2005].

3. Proposed Methodology

The proposed methodology of the research was to develop fuzzy expert system model for categorizing customer and service advisor within contact centres with the use of Matlab fuzzy logic toolbox. The steps followed for the development of the expert system model for categorizing customer and service advisor are shown in Figure 1 and explained later in the paper.

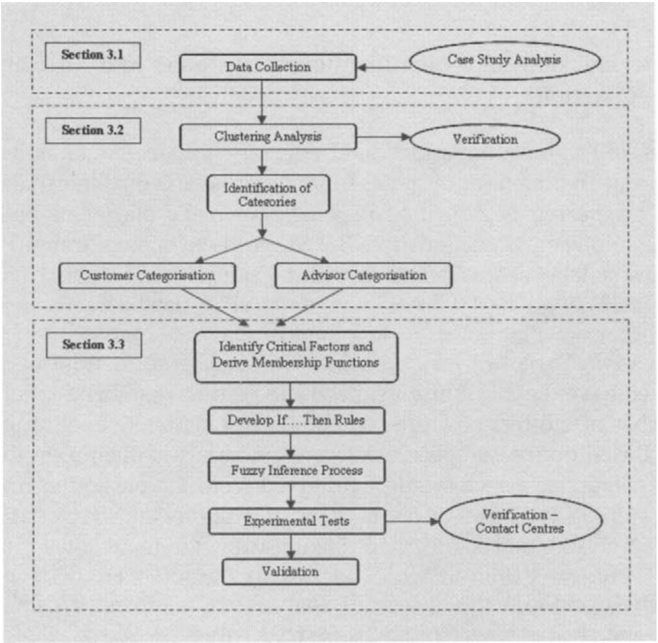


Fig. 1. Flowchart for the proposed methodology

3.1 Data Collection

Data was collected with the help of open set of questionnaires for customer service advisor (CSA) and team leaders/managers with respect to their demographic variables, experience and behavioural variables within five customer contact centre focusing on fault and sales and looking on single to multi profile business customers. A total of 84 advisors were interviewed and assessed, 60 customer calls were heard and monitored, and total of 19 team leaders and managers were interviewed through the questionnaires. For CSA data collection, the author had gone through the questions and also monitored any change in their behaviours during or after the call conversation with the customers at the centre. For customer data collection, the collection was done on the basis of the information provided on the screen of the advisor when the call conversation was in progress, and also the author's hearing to the calls to identify the behavioural aspect of the customer, before the call and once the call was finished. Based on the data collection and the analysis of data the attributes derived for customer and advisor are as follows:

> Customer – age, education, financial status, time with company, business value and behavioural analysis
> Advisor – age, education, experience, previous experience, IT speed, and behavioural analysis

3.2 Clustering Analysis – Identification of Customer and Advisor Categorisation

Classification is a most important and frequently used technique in data mining. It is a process of finding a set of models that describe and distinguish data classes or concepts. ***Clustering*** is a method in which we make cluster of objects that are some how similar in characteristics. Based on the data structuring done from the case studies, a data set was designed with 60 samples of customer records and 84 samples (cases) of advisor (CSA's) within the SPSS database. Because of the total number of cases available was of less number and the complexity of the customer data, *Two Step Clustering Process* within SPSS was used. With two step cluster analysis, we have the flexibility to specify the cluster numbers, specify the maximum number of clusters or let the technique automatically choose the number of clusters. Based on the samples, clustering analysis was then used to identify the groups of categories which would be derived from the clustering results [Johann *et.al* 2001] [SPSS Inc]. Ten different types of experiments were carried out within the two step cluster analysis method ranging from automatic clustering to a maximum of 10 clusters within SPSS. A total of six clusters were derived for customers and advisors. From the clustering analysis the samples of categorisation for customers and customer service advisors (CSA) are:

- Customer Category C1 (Angry Customer) – Female, 18-25, School (education), Poor (financial status), 1-5 yrs (time with company), Low (business value), Angry and Aggressive (behaviour).

- Advisor Category A1 (Novice Advisor) – Female, 18-25 (age), School (education), >1yrs (experience), Low (IT speed), none (previous exp.), Angry and Unaware (behaviour).

3.3 Development of Fuzzy Expert System

Fuzzy logic has proved useful for developing many practical applications, especially in the field of engineering, as it can handle inexact and vague information. This section discusses the steps followed for the development of the fuzzy expert system for customer and CSA categorisation.

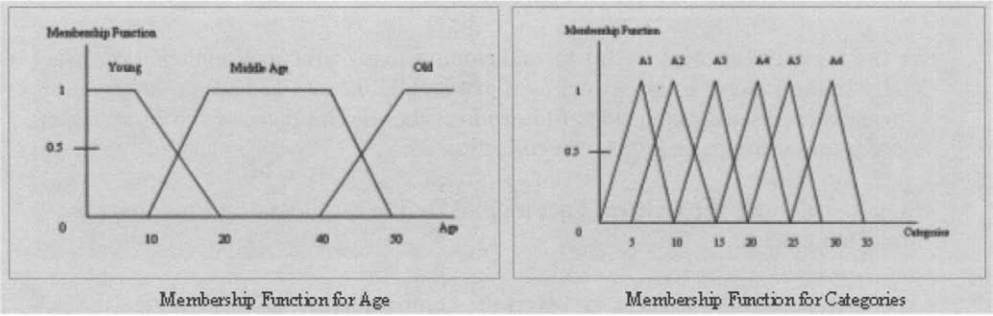
Step 1 – Identify the Critical Factors and Define Membership Functions and Fuzzy Sets

The first step of the process involved the combination of a list of critical factors based on the literature review and in-depth interviews with the advisor, team leaders, centre managers and systems expert within the environment. The critical factors were the input variables of the fuzzy ES which were as age, gender, and education, and financial background, time with the company, business value and behaviour from the customer side which would identify the type of category they belong.

Table 1. Linguistic Terms for Membership Functions in Fuzzy Model

Customer Service Advisor (CSA)	Customer
1. Age – young, middle age, old	1. Age – young, middle age, old
2. Education – school, college, graduate, professional	2. Education – school, college, graduate, professional
3. Experience – novice, medium, senior	3. Financial Status – poor, average, good
4. IT Speed – slow, medium, fast	4. Time with company – low, moderate, high
5. Previous Exp – low, moderate, extensive	5. Business value – low, medium, high
6. Positive Behaviour – attentive, friendly, customer focus	6. Positive Behaviour – joyful, co-operative understanding
7. Negative Behaviour – unaware, annoyed angry	7. Negative Behaviour – angry, annoyed, aggressive

The development of the model was done by author's own understanding for the current contact centre and from the literature studies. Once the selection was done and the model was developed, it was validated with expert judgment from the team leaders at the centre through nine team leaders and managers at three of the case study contact centres. The simplest membership functions are formed using straight lines. Of these, the simplest is the *triangular* membership function, and it has the function name trimf. The *trapezoidal* membership function, trapmf, has a flat top and really is just a truncated triangle curve.



Membership Function for Age

Membership Function for Categories

Fig. 2. Sample Membership Functions for Input (Age) and Output (Categories) of the System

Step 2 – Construct the Fuzzy Rules

Fuzzy ES make decisions and generate output values based on knowledge provided by the designer in the form of IF {condition} THEN {action} rules. The rule base specifies qualitatively how the output of the system "Category" for the advisor and the customer is determined for various instances of the input variables of Age, Education, Financial Status, and Time with Company, Business Value, Experience, and Behavioral attributes. Based on the initial results from the clustering of customers and advisors, the author identified the important rules from the results. A sample of the rules derived for the fuzzy logic expert system are shown below and explained later in the section (Table 2 and 3).

Table 2. Sample of Advisor Fuzzy If….Then Rules

Age	Education	Experience	Previous Experience	IT Speed	Positive Behaviour	Negative Behaviour	Category
Young	School	Novice	Low	Slow	Friendly	Unaware	A1
Middle	Graduate	Medium	Moderate	Medium	Attentive	Annoyed	A3
Old	Profess.	Senior	Extensive	Medium	Focus	None	A5
Young	College	Novice	Moderate	Fast	Focus	Unaware	A6
Young	Graduate	Novice	Low	Fast	Attentive	Annoyed	A2
Old	Graduate	Senior	Extensive	Fast	Friendly	None	A4
Young	Graduate	Medium	Moderate	Fast	Attentive	None	A2

Table 3. Sample of Customer Fuzzy If….Then Rules

Age	Education	Financial Status	Time with Company	Business Value	Positive Behaviour	Negative Behaviour	Category
Young	School	Poor	Low	Low	None	Aggressive	C1
Middle	Graduate	Good	Moderate	Low	None	Annoyed	C2
Old	Graduate	Average	Moderate	Medium	Understanding	Angry	C6
Young	College	Poor	Low	Medium	Co-operative	None	C3
Middle	Professional	Good	Moderate	High	Joyful	None	C5
Old	Professional	Average	High	High	Joyful	Annoyed	C4
Middle	School	Poor	High	Medium	None	Aggressive	C1

The above given rules are the sample rules derived for the system. Few rules were written which covered all aspects of the inputs and had a range of output categories for service advisors as A1-A6 categories and C1-C6 categories for customers. The rules within the expert system model were developed from the literature studies and from the understanding the author had developed while doing the case studies and the data collection and analysis phase of the research. Once the rules were derived, the author validated with that of the team leaders from the contact centre, whether the rules were of significance in the real world environment.

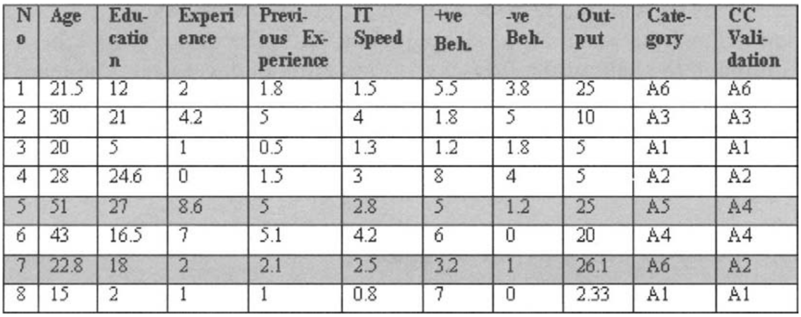
4. Experimental Examples and Results

With respect to the model, the author carried out few experiments with the fuzzy expert system model by changing the input variable values and monitoring the change in the output which showed the change in the category for customer and advisors. The summary of the experiments which were carried out are shown later in the Section 4.1 and 4.2 (refer Table 4 and 5). The results derived from the experiments carried out within the expert system model were validated within the contact centre environment with the team leaders and managers. The experiments which were carried out within the model as explained above were done on the basis of the changes made within the rule viewer by changing the input values of the model and observing the output selection of assignment of the categories for customers/advisors.

4.1 Customer Advisor (CSA) Experiments

Ex. 1 - If Age = 21.5, Education = 12, Experience = 2, IT Speed = 1.5, Previous Exp = 1.8, Positive Behaviour = 5.5, Negative Behaviour = 3.8. Then Advisor Category output is 25 which determines that the category for advisor is A6.
Ex. 2 - If Age = 30, Education = 21, Experience = 4.2, IT Speed = 5, Previous Exp = 4, Positive Behaviour = 1.8, Negative Behaviour = 5. Then Advisor Category output is 10 which determine that the category for advisor is A3.

Table 4. Experimental Results for Advisor Expert System Model Summary



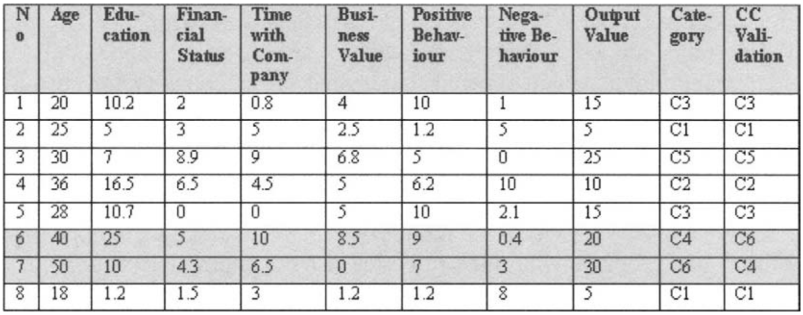
No	Age	Education	Experience	Previous Experience	IT Speed	+ve Beh.	-ve Beh.	Output	Category	CC Validation
1	21.5	12	2	1.8	1.5	5.5	3.8	25	A6	A6
2	30	21	4.2	5	4	1.8	5	10	A3	A3
3	20	5	1	0.5	1.3	1.2	1.8	5	A1	A1
4	28	24.6	0	1.5	3	8	4	5	A2	A2
5	51	27	8.6	5	2.8	5	1.2	25	A5	A4
6	43	16.5	7	5.1	4.2	6	0	20	A4	A4
7	22.8	18	2	2.1	2.5	3.2	1	26.1	A6	A2
8	15	2	1	1	0.8	7	0	2.33	A1	A1

4.2 Customer Experimental Results

Ex. 1 - If Age = 20, Education = 10.2, Financial Status = 2, Time with company = 0.8, Business Value = 4, Positive Behaviour = 10, Negative Behaviour = 1. Then Customer Category output is 5 and the category is C3

Ex.2 - If Age = 25, Education = 5, Financial Status = 3, Time with company = 5, Business Value = 2.5, Positive Behaviour = 1.2, Negative Behaviour = 5. Then Customer Category is C1.

Table 5. Experimental Results for Customer Expert System Model Summary



No	Age	Education	Financial Status	Time with Company	Business Value	Positive Behaviour	Negative Behaviour	Output Value	Category	CC Validation
1	20	10.2	2	0.8	4	10	1	15	C3	C3
2	25	5	3	5	2.5	1.2	5	5	C1	C1
3	30	7	8.9	9	6.8	5	0	25	C5	C5
4	36	16.5	6.5	4.5	5	6.2	10	10	C2	C2
5	28	10.7	0	0	5	10	2.1	15	C3	C3
6	40	25	5	10	8.5	9	0.4	20	C4	C6
7	50	10	4.3	6.5	0	7	3	30	C6	C4
8	18	1.2	1.5	3	1.2	1.2	8	5	C1	C1

Based on the model, the author identified that the results derived from the model, assigned a customer with the pre-determined category which were derived from the clustering. These results were also validated with the team leaders at the contact centre to verify that the given selection of the pre-determined categories for customer was properly justified.

4.3 Validation

The information and the results from the model were verified within industry expert judgment through team leaders and managers at three of the contact centres where the case studies were carried out. A total of nine team leaders and managers were interviewed with the help of an open set questionnaire, showing the categories derived and the assignment of a particular customer or advisor to these categories through the help of the fuzzy expert system tool developed. The team leaders at the contact centre were shown the possible combinations of the customer and advisor categories, and on what basis these categories were derived (Table 4/5)

5. Discussion and Future Research

The authors have demonstrated the steps which were followed for thee development of a fuzzy expert system to assign the customer and advisor to the pre-determined category. The experimental results in Table (4) and (5) shows that

80% of results are as expected, and were assigning a particular customer and advisor to the categories which were derived from clustering. Based on experiment 5, the expert system assigned category A5 to the advisor. However from validation with team leaders it revealed that the category should be A4. On the basis of the validation the changes were made with respect to behavioural attributes from friendly behaviour to customer focus behaviour. The reasons for this swift change in selection of category were due to (a) Education level to be high (b) Positive behaviour to be attentive and (c) Less amount of negative behaviour.

For customer categorisation, the results from the expert system for experiment 6 and 7 did not match that to the validation from the team leaders at CC (Table 5). As seen in experiment 6, the changes made were education level was changed from graduate to college level to assign customer with C6 category. The changes made within the expert system were: (a) Customer time within company (b) Positive attitude towards the advisor and (c) Less amount of negative attitude shown from the customer.

The next step of this research is to develop an information required framework which would identify the type of information required by the advisor to serve the customer. On the basis of assignment of any customer or advisor through the expert system tool to that of pre-determined category we can identify the type of caller (customer) and the random advisor assigned to that customer.

6. Conclusions

This paper is focused on the development of methodology for designing fuzzy expert system for customer and advisor (CSA) categorisation within contact centre environment. The authors identified the initial attributes to be used within the model for customer and advisor through five case studies carried out at the customer contact centres within UK ranging from telecoms to help desk and government; and 84 customer advisor and 60 customers. Based on the data collection and analysis of data, the author identified the attributes for advisor and customer respectively.

A fuzzy expert system was developed to assign any customer or advisor to that of the pre-determined category from the clustering analysis. Experimental tests were carried out to check the assignment of the customer and advisor to the categories derived from the clustering analysis were same and related to each other. The results showed that over eighty percent of the time the assignment from the expert system for the categorisation of the customer and advisor was correct; which was validated with the team leaders at the case study contact centres.

Acknowledgements

The author wish to acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC), BT Telecommunication Plc, and Decision Engineering Group (DEG) Cranfield University; UK.

References

Bushey, R.; Mauney, J. M. and Deelman, T. (1999). The development of behaviour-based user models for a computer system. 7th International Conference on User Modeling (UM99). 20th -24th June 1999. Banff Centre, Banff. Canada. Pp 109-18

Chiu, C (2002). A case based customer classification approach for direct marketing. Expert Systems with Applications. Vol 22 (2002). Pp 163-168. Elsevier Science

Dolen, W; Ruyter, K and Lemmink, J. (2004). An empirical assessment of the influence of customer emotions and contact employee performance on encounter and relationship satisfaction. Journal of Business Research. Vol. 57. Pp 437-444.

Goff, B. G; Boles, J. S; Bellenger, D. N and Stojack, C. (1997). The influence of salesperson selling behaviours on customer satisfaction with products. Journal of Retailing. Vol 32. No 2. pp 171-183.

Harline, M. D and Ferrell, O. C (1996). The management of customer contact service employees: an empirical investigation. Journal of Marketing. Vol 60 (1996). Pp 52-70.

Heckman, R and Guskey, A (1998). Sources of customer satisfaction and dissatisfaction with information technology help desks. Journal of Market Focused Management. Vol 3. pp 59-89.

Hu, T. and Sheu, J. (2003). A fuzzy based customer classification method for demand responsive logistical distribution operations. Fuzzy Sets and Systems. Vol 139. No.2 October 2003. PP 431-450

Johann, B; Knut, W. and Melanie, V. (2001). SPSS Two Step Cluster – A first evaluation. http://www.statisticalinnovations.com/products/twostep.pdf

Martinez, F.E; Magoulas G.D.; Chen S., and Macredie R. (2004). Recent soft computing approaches to user modeling in Adaptive Hypermedia. Proceedings of 3rd Int. Conference Adaptive Hypermedia-AH 2004, Eindhoven, The Netherlands, Aug. 2004, Lecture Notes in Computer Science, vol. 3137, Springer, 104-113.

Mathworks Fuzzy Logic Toolbox. Model based design software. Available at: http://www.mathworks.com/products/fuzzylogic/ (accessed on: 15/01/05)

Ngai, E.W.T and Wat, F.K.T (2003). Design and development of a fuzzy expert system for hotel selection. International Journal of Management Science. Vol 31. PP 275-86.

Song, H. S; Kim, J. K. and Kim, S. H. (2001). Mining the change of customer behaviour in an Internet shopping mall. Expert Systems and Applications. Vol 21. pp 157-168. Elsevier Science Ltd.

SPSS Inc (2005). Statistics & Data Mining Software: SPSS.

Swinyard, W. R (2003). The effects of salesperson mood, shopper behaviour and store type on customer service. Journal of Retailing and Consumer Services. Vol 10 (2003). Pp 323-333. Elsevier Science Ltd.

Zadeh, L. (1996). The role of soft computing and fuzzy logic in conception, design and deployment of intelligent systems. Proceedings International Workshop on Soft Computing in Industry, Muroran, Japan, April 1996, pp 136-137.

Zeelenberg, M. and Pieters, R. (2004). Beyond valence in customer dissatisfaction: A review and new findings on behavioural responses to regret and disappointment in failed services. Journal of Business Research. Vol 57 (2004). Pp 445-455. Elsevier Science Ltd.

Soft Computing for Intelligent Information Management

Ben Azvine, Trevor Martin[1] and Marcus Thint

Intelligent Systems Lab, BT Research & Venturing, Adastral Park, Ipswich, UK.
ben.azvine@bt.com, marcus.2.thint@bt.com, trevor.martin@bris.ac.uk

Abstract. This paper focuses on the application of advanced soft computing methods to text content and meta-data in order to integrate and organise both structured and semi-structured information. The primary aim is to reduce the time spent in collecting information from different sources. The paper covers two aspects of the process, namely the recognition of similar information from multiple sources and the integration of hierarchical classifications from different sources.

Introduction

It has been predicted that most, if not all, significant human creative output will be available online within 5-10 years (together with a huge volume of less significant material). Approximate figures give a rough idea of the problem scale arising from this volume of data - Landauer [1] estimated that a human brain can hold about 200 megabytes of "information from experience", and a report by the Berkeley School of Information Management and Systems in 2000 estimated the world's total yearly production of print, film, optical, and magnetic content would require roughly 1.5 billion gigabytes of storage. This is roughly 250 megabytes per person for each man, woman, and child on earth [2], and by 2003 the estimate had risen to 800 megabytes of new data per person per year [3]. Clearly the "brain capacity" is not directly comparable to these data production figures - human recall of a photograph, for instance, does not require "storage" of every pixel. However it is equally clear that the majority of stored data will never be directly accessed by a human being. Human attention is the scarcest resource in the processes of information creation, transmission, storage and use, and machine assistance is needed to focus human attention in an efficient way.

[1] Permanent address: AI Group, University of Bristol, BS8 1TR UK

In order to deal effectively with this problem, we need intelligent information management. Our position is that this depends on the availability of adequate meta-data to indicate the nature of the information. Such meta-data may be created in advance of use or computed dynamically from the content (this is a broad view of meta-data which includes any features or summaries derived from, or describing, the content). There are few standards for meta-data of this sort, and there may be even less agreement on how the meta-data should describe the content. Even a straightforward schema such as the Dublin Core allows for free text in parts of the description, and hence there is a degree of subjectivity. Although this can be restricted to some degree by use of controlled vocabularies, free text almost inevitably leads to a problem in matching user queries with a data source, or in combining data from more than one source. Although meta-data is frequently not consistent between different sources, the delivery of intelligent information management requires the capability to automatically integrate information from different sources.

In this paper we outline parts of a research programme into the use of soft computing for intelligent information management, addressing the problem of information integration. Previous work [4] has described how domain-dependent vocabularies can be extracted, allowing term equivalences and near equivalences to be built. Here, we focus on higher-level aspects of the problem - the need to integrate responses from multiple sources into a single response, and the combination of different hierarchical classifications.

Integrating Query Responses

The need for information fusion exists in the semi-structured and unstructured domains – for example, to integrate responses from multiple sources into a unified response. Although technology falls far short of being able to compose a unified response anew, it is feasible to recognise similar/duplicate information and to formulate an aggregate response.

Operation scenario

Key functions of an intelligent information management system include analysis and/or re-formulation of a query, routing to appropriate information sources, and returning a filtered and integrated response (see Fig. 1). Query processing and identification/discovery of suitable knowledge sources are performed by the Smart SourceFinder module. This section focuses on the "Response Integrator" module and the current work assumes (simulates) five data sources, all returning responses to a query. The sample queries are about films, and responses are film reviews from:

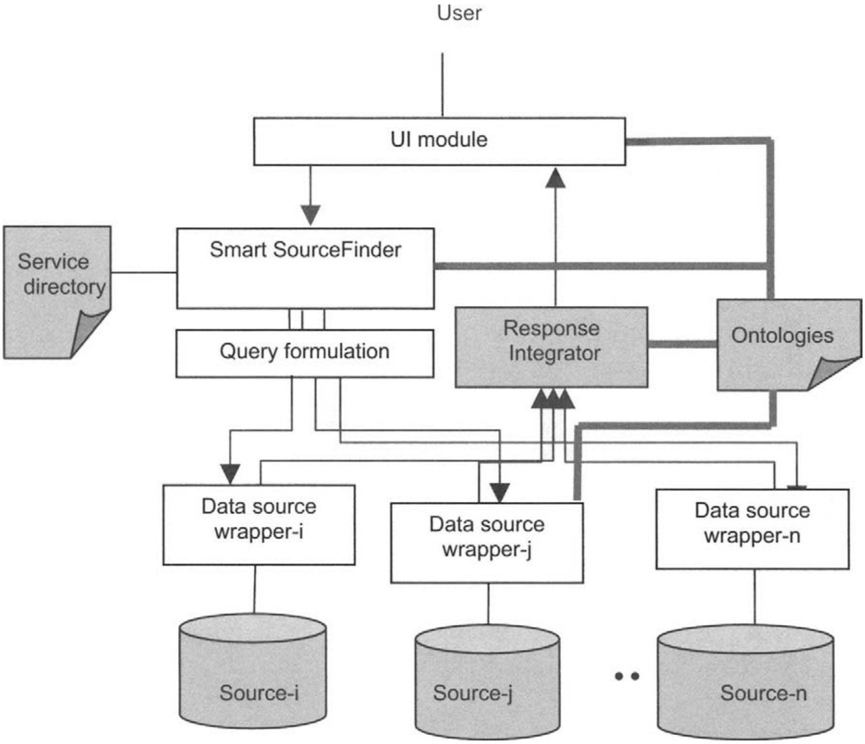


Fig. 1. Response Integrator module in an information management system

- Chicago Tribune (http://metromix.chicagotribune.com/movies),
- Film Journal International (http://www.filmjournal.com/filmjournal/index.jsp),
- Roger Ebert (http://rogerebert.suntimes.com/apps/pbcs.dll/frontpage),
- Reel Views (http://movie-reviews.colossus.net), and
- The New York Times (http://www.nytimes.com/pages/movies/index.html).

From these sources, a local knowledge base for 20 Oscar-nominated films of 2004 was compiled and used for testing. Note that even for a human, after reading through responses from all sources, it is a difficult task to identify similarities and differences in these text passages and perform the integration process.

Response Integrator Module

Components of the Response Integrator module are shown in Fig. 2. It receives multiple unstructured text segments from several (N=5) sources, which are first processed in the preprocessor module. Different information sources use diverse tags or delimiters to identify parts of their content, and this preprocessor assumes/allows some supplemental information (source name, film title, film year, critic name) in XML syntax. Copies of the preprocessor output (text body) are used by the Summarizer and Keyphrase Extractor modules. The Summarizer condenses original text by extracting important sentences, to a final length approximating a parameter specified as a total number of words or percentage of the original document length. Sentence selection is based on its position, TF-IDF values, word overlap and bigram overlap. Some Keyphrase Extractors employ a domain-specific approach whereby they are first trained with a representative document set, but the version in the response integrator is not domain-specific (i.e. keyphrases are relative to general English word statistics and TF-IDF). Hence, keyphrase quality may be slightly inferior, but this choice is more appropriate for general use, because some application realms (e.g. films) cover a wide-range of domains/topics. The Summarizer and Keyphrase Extractor employed in the Response Integrator are BT-developed.

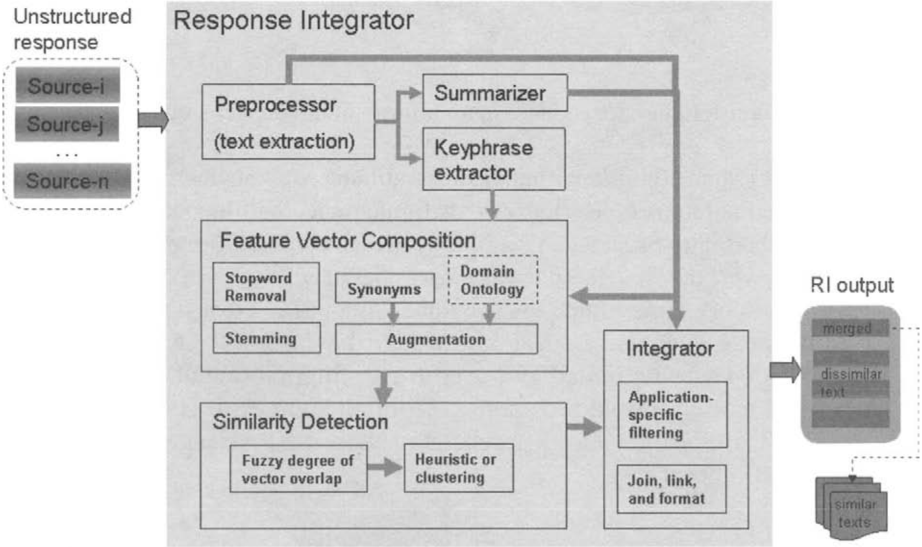


Fig. 2. Response Integrator components

The Feature Vector Composition module takes the condensed text and keyphrases as input, and produces final symbolic vectors to be compared among the different text summaries. Its operations include Stopword removal, Stemming, concept Augmentation, and removing duplicates to produce a document feature vector. Common synonyms are obtained from Wordnet, and optionally, a domain-specific lexicon could be used as an additional resource. Whereas an application example in medicine, law, or technology domains would have required a domain-specific ontology server, content for the present example (film reviews) do not. Concept Augmentation increases the chance of matching/recognizing similar statements when synonymous expressions are used.

Document feature vectors contain the keyphrases and stemmed words of the summary and their synonyms. The vector lengths are not uniform, neither are their corresponding summaries. The symbolic vector elements are compared against each other in nested loops, to assess similarities in the summarized text. Relative similarity measures (to each query response) are currently used to evaluate inter-document similarity. It may be possible to use an absolute threshold but it is doubtful that a value applicable across all content style and length could be determined, and the sample knowledgebase (approx. 100 documents) is too small to suggest a conclusion. A ratio or percentage of matched terms in the feature vectors (high/medium/low) are used, and the following heuristic is applied to produce the outputs discussed below:

- The document pair (i,j) with the highest similarity SH is marked 'merged' and excluded from further processing
- Document pair (p,q), $[p \neq \{i, j\}$ and $q \neq \{i,j\}]$ with the lowest similarity SL are marked 'distinct' for definite output
- For each remaining documents ($k, k \neq \{i,j,p,q\}$), merge with $\{i,j\}$ if $SH - S(k,i) < \varepsilon 1$ or $SH - S(k,j) < \varepsilon 1$ and merge with p or q if $SH - S(k,p) < \varepsilon 2$ or $SH - S(k,q) < \varepsilon 2$, where $\varepsilon a = \lambda\, a\, (SH - SL^*)$. (It may be that true minimum similarity value $SL^* = SL$, but in cases where SL^* applies to pair {i, m) or (j, m) where i and j have been excluded after the first step above, $SL^* \neq SL$). λ is a design parameter and in a simple case $\varepsilon 1 = \varepsilon 2$ (i.e. all $\lambda\, a = \lambda$ and $\varepsilon a = \varepsilon$).

Examination of the similarity values also revealed that a simple one-dimensional clustering could be employed to classify the metrics into fuzzy groups (high/medium/low) to determine which text passages should be grouped together, but the performance gain/loss is a subjective evaluation. Clustering, as illustrated in Fig. 3, is likely to merge more documents as similar - thus in applications where it is more critical not to "miss" a

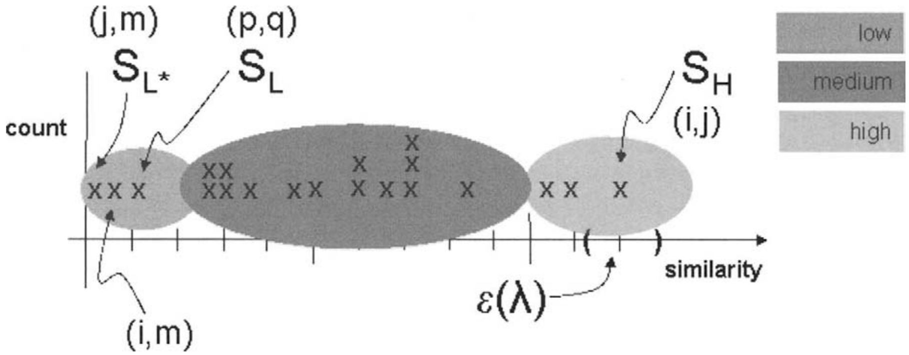


Fig. 3. Sample similarity profile and cluster regions

document by being classified as similar/duplicate, the heuristic approach with tight ε is preferable, whereas applications that seek higher information compression would favour the clustering approach. (Note that 'merged' documents need not be irretrievably discarded – they could be accumulated in the 'background' and user given an option to view them).

Output Results

The Integrator module composes the Response Integrator output via (i) retrieving the original or summary of documents corresponding to the selected feature vectors (ii) applying application specific filters, and (iii) joining and formatting and linking the selected outputs for the end user.

Once a set of documents are identified as similar/duplicates, it is desirable to show one representative sample and discard or archive the rest. But how should the "representative" or "best" document be determined? At least in part, that depends on the application, and thus an application-specific filtering is used. In this work, it is assumed the user wants a "useful" film review, and those containing more critical vocabulary (positive examples:. *great, must see, superb, wonderful, dazzling, remarkable, finest…* and negative examples: *bad, terrible, disappoint, superficial, flawed, fails to…*) are selected over the others. In a different application, this module can be replaced with alternative logic.

Subsequently, joining the various texts and formatting for display completes the Response Integrator module. The quality of "decisions" (passages selected as similar/duplicate) is difficult to assess even by a human. Due to the nature of the subject domain (film review) the tone of some prose is ambiguous (e.g. reviewers praise some aspects of a film and criticize other aspects). Additional "controlled" (viz. input text passages) testing would be useful to fully evaluate the efficacy of this approach.

Correspondence Between Meta-Data Hierarchies

The second aspect of this work is a method to find correspondences between hierarchies, which are described by different meta-data schemata. Such correspondences may also be used to improve the identification of equivalent instances. Some successful initial tests of the method on large movie databases are reported below.

Meta-data often contains useful hierarchical categorisation but this is frequently not consistent between different sources. Furthermore, the hierarchies may not reflect an individual's preferred way of organising information. The classification structure and attributes (properties) of the objects (i.e. the values associated with meta-data tags) can be used to guide searching and integration of multiple sources. Even if different hierarchies use different categories, there is likely to be a degree of correspondence, and objects placed within similar categories are likely to have similar properties. For example, a digital library and an online bookseller refer to the same (structured) objects but may differ in categorisation and details stored about each book.

The properties can be used to group the objects into classes, which may in turn form some sort of hierarchical structure. This leads to the "ontology alignment" or "schema matching" problem, when different sources classify the same set of objects according to two different hierarchies. The method outlined here uses "instance-matching" initially, to determine that objects from different sources are the same; this is extended to compare and/or predict the hierarchical classification.

To illustrate, we consider two specific meta-data sources which describe films using different sets of tags and (importantly) different genre hierarchies to classify the films. Given mappings between meta-data attributes, previous work [5] has shown how we can automatically identify instances which are equivalent on the basis of their attributes.

We use the equivalence of instances from different sources to learn a soft mapping between genre hierarchies, allowing us to compare the hierarchical classification of instances as well as their attributes.

Class Matching

Given two (or more) hierarchically organised sets of instances, we can use the SOFT method [5] to determine which instances are equivalent by comparing their attributes. Having determined equivalent instances from the two sources, we can look for correspondences between the different classification structures. For example, online music sources are typically organised hierarchically, but one site's

music > rock > classic rock > 70's classics
section may correspond (or correspond mostly) to another's
music > rock&pop oldies
because both contain mostly the same tracks (or albums) where the notion of "same" is determined by SOFT.

The > symbol indicates that all tracks in a sub-category belong to the broader "parent" category e.g. anything classified as classic rock also belongs to rock and hence also to the category music. For convenience we define X > X for any category X.

In general, we consider two sets of entities A and B with corresponding sets of labels LA and LB each of which has a hierarchical structure i.e. there is a partial order defined on the labels.

Each label $l_i \in L_A$ denotes a subset of A i.e. we have a denotation function

$$den : L_A \to A$$

such that

$$l_i > l_j \quad \Leftrightarrow den(l_j) \subseteq den(l_i)$$

(and similarly for B)

For example, if A and B are sets of films then L_A and L_B could be genres such as *western, action, thriller, romance*, etc

In previous work we have shown how to derive a soft mapping on the sets of entities A and B

$h : A \to \tilde{P}(B)$ where $\tilde{P}(B)$ is the set of all fuzzy subsets of B

It may not be possible to say in all cases that an element of A corresponds to a specific element of B or that it does not correspond to any element of B. This mapping is used to determine a (soft) correspondence between any pair of labels l_i and l_j from the label sets L_A and L_B

$$g : L_A \to \tilde{P}(L_B)$$

Given a label $l_i \in L_A$ we consider its denotation *den*(l_i) under the mapping *h* and compare it to the denotation of $l_j \in L_B$

In the ideal case if the two labels are equivalent,

$$h(den(l_i)) = den(l_j)$$

Given that *h* is approximate and that the correspondence between labels may not be exact, we use semantic unification to compare the sets.

$$\Pr(l_i \to l_j) = \Pr(h(den(l_i)) = den(l_j))$$

This gives an interval-valued conditional probability which expresses the relation between a pair of labels; we then extract the most likely pair to give a crisp relation

$$g_c : L_A \to L_B$$

Ideally, it should be possible to map such categories into a user's personal hierarchy – here, we concentrate on extracting rules from the overlap between categories in different classification structures based on a sample and use the derived rules to predict likely categorisations of new examples.

Application to Film Databases

The two film websites "rotten tomatoes"[2] and the internet movie database[3] are "user-maintained" datasets which aim to catalogue movie information. The databases denoted *dbT* and *dbI* below are derived from these sites, respectively containing 94,500 and 94,176 film records, and were used in experiments. Since *dbT* and *dbI* are produced by two different movie web sites, there is inevitable "noise" existing in the film data; i.e. different tag sets, different genre names and missing elements.

dbI	
TITLE	Gentleman B.
YEAR	2000
DIRECTED_BY	Jordan Alan,
GENRE	Thriller
Country	USA

dbT	
TITLE	Gentleman Bandit
Year	2000
CAST	Ed Lauter, Peter Greene, Justine Miceli, Ryan O'Neal, Charlie Mattera
GENRE	Dramas, Drama
MPAA_RATING	NOT Rated

Instance and Genre Matching

In order to match instances by means of their attributes, some very simple string matching functions were used, as follows:

(i) String S1 is an approximate substring of S2 if S1 is shorter than S2 and most words in S1 are also in S2.

(ii) String S1 is an approximate permutation of S2 if they have a high proportion of common words, i.e. degree of match = proportion of common words, which must be at least two.

Both ignore "stop" words such as the, and, etc. We note also that it is possible to obtain better results for people's names (attributes such as cast, director, etc) using a more structured approach which extracts first name and surname and then matches on that basis.

The average matches between domains are given in Table 1

[2] www.rottentomatoes.com

[3] Information courtesy of The Internet Movie Database (http://www.imdb.com). Used with permission. Results from Martin and Shen, AMR 2005

Table 1. Average degree of match between attributes in dbI and dbT

dbI attributes	*dbT* attributes	average match
TITLE	TITLE	41%
DIRECTED_BY	DIRECTOR	27%
AKA	TITLE	21%

On the basis of the three attributes, the system identified movies from *dbI* dated 1976-1990 which were also in *dbT*, and compared the genre classification. The similarity threshold between two film records was set to 0.5 giving a total of 14124 movies which are found to be identical.

The similarity between two genres is relatively hard to decide from text string matching. For example, "*animation*" is not similar to "*children*" from the point of view of text matching, but the extension of the sets of films in these two categories shows considerable overlap. Some examples of interesting genre mappings are

dbT Genre	*dbI Genre*
ANIMATION	*CHILDREN*
HORROR	*SUSPENSE*
SCI-FI	*FANTASY*

Results on Unseen Data

The attribute and genre mappings were applied to a new set of 24,839 entries from *dbI* (calendar years 2000-05), trying to find matches in *dbT*. For comparison, a manually produced ground truth established 1274 true matches – this figure is low due to the relatively large number of entries for TV series, "foreign" movies etc in *dbI* which are not included in *dbT*. Using the SOFT algorithm without genre mapping, we find 861 pairs of matching film entries when the similarity threshold between two films is set to 0.44. With the presence of the ground truth, 261 film matching pairs out of 382 film pairs in 2000 are missing, 102 out of 364 in 2001 are missing, 87 out of 330 in 2002 are missing, 60 out of 142 in 2003 are missing, and 3 out of 8 in 2004 are missing.

This represents a recall of 67 % and a precision of 100%. Incorporating the genre mapping as well produces a much better (100%) recall, with a loss in precision (65%).

Summary

We see the availability of good meta-data as crucial to the development of intelligent information management systems, and the use of soft methods for approximate matching as a vital component in such systems. We have presented methods for determining similarity of content and for finding identifying similar categories within hierarchical classifications.

Initial results are promising. Further testing is necessary to fully establish the robustness of this work, involving a wider variety of datasets. The use of this method with a personal hierarchy would enable the user to adapt source meta-data to his/her own preferences and hence improve the retrieval process.

References

1. Landauer, T.K., How much do people remember? Some estimates of the quantity of learned information in long-term memory. Cognitive Science, 1986. **10**: p. 477-493.
2. Lyman, P. and H.R. Varian, *How Much Information.* 2000, http://www.sims.berkeley.edu/how-much-info
3. Lyman, P. and H.R. Varian, *How Much Information (2003).* 2003, http://www.sims.berkeley.edu/how-much-info-2003
4. Martin, T.P. and B. Azvine, *Acquisition of Soft Taxonomies for Intelligent Personal Hierarchies and the Soft Semantic Web.* BT Technology Journal, 2003. **21**(4): p. 113-122.
5. Martin, T.P. and B. Azvine, *Soft Integration of Information with Semantic Gaps*, in *Fuzzy Logic and the Semantic Web*, E. Sanchez, Editor. 2005, Elsevier.

Soft Computing in Intelligent Data Analysis

Ben Azvine, Detlef Nauck, and Martin Spott

Intelligent Systems Research Centre, BT Research and Venturing
Adastral Park, Orion Building pp1/12
Martlesham, Ipswich, IP5 3RE
{ben.azvine,detlef.nauck,martin.spott}@bt.com

Summary. In today's dynamic business environment it is paramount for businesses to turn data first into information and then into action quickly. If data is available, business decisions should be based on knowledge derived from the data. However, analysis results are often not available and decisions are merely based on assumptions about data. One of the reasons for missing analysis results is the fact that available tools are still technology centered and require analysis experts as users which are often not at hand. We introduce three of our software tools that help business users analyse data at different levels: a dedicated intelligent data analysis tool, a problem specific solution, and a platform integrated in an operational system. Thereby, soft computing techniques proved useful since they can deal with vague and uncertain information which is common in our applications and also present information in a way that is easy to understand for business users.

Keywords: Soft computing applications, intelligent data analysis, what-if analysis, rule discovery

1 Introduction

In today's dynamic business environment it is paramount for businesses to turn data first into information and then into action quickly. If data is available, business decisions should be based on knowledge derived from the data. However, analysis results are often not available and decisions are merely based on assumptions about data. But we can only truly expect good decisions if they are based on proper data analysis and sensible combination of analysis results with business knowledge. The analysis process is often hampered by missing tools and insufficient analysis skills. Software platforms that automate the analysis process and implement results in a controlled way can help businesses in making good use of their operational data and improve their processes.

In order to transform data into valuable information, an intelligent approach to data analysis is required. Intelligent data analysis (IDA) goes one step further than todays data mining approaches and also considers the suitability of the created solutions in terms of usability, comprehension, simplicity and cost. The intelligence in IDA comes from the expert knowledge that can be integrated into the analysis process, the knowledge-based methods used for analysis, and the new knowledge created and communicated by the analysis process.

Whatever strategy businesses pursue today, cost reduction is invariably at the heart of it. In order to succeed they must know the performance of their processes and find means to optimise them. IDA provides means to find and combine process knowledge with the collected data. Learning systems based on IDA methods can continuously optimise processes and also provide new knowledge about the business. IDA is therefore an important aspect in modern knowledge management and business intelligence.

In the following, we present three of our applications that help business users analyse data at different levels: a dedicated intelligent data analysis tool that can be used by analysts who are not necessarily data analysis experts, a problem specific solution for analysing customer satisfaction, and a tool integrated in an operational system that automatically analyses data in the background. All projects feature soft computing techniques like fuzzy approaches for specifying domain knowledge, drawing fuzzy conclusions or summarising data in terms of fuzzy rules.

2 Towards the Automation of Intelligent Data Analysis

Traditionally, experts are required to run an IDA process. However, the ever-growing need for business automation requires either support for non-experts or data analysis tools that can run largely unsupervised. In the future businesses will not have the time anymore to deploy a team of experts who assess and analyse the requirements for improving a product like, for example, an Internet service, find an optimal solution and implement it. The data about service usage must be analysed online and used immediately to improve the service.

We cannot expect that all analysts are experts for the data analysis methods they apply to problems in their domains. In the same way we do not expect a driver to be capable of repairing his car or a computer user to understand the function of a CPU. Geologists or physicians, for example, are not interested in the mathematical theories behind some analysis method they apply to their data but in the answers to questions, like, where to drill for oil, or which treatment is best for a certain disease. The point is that data analysis has become a practical area and data analysis methods nowadays are used as tools. This pragmatic view on data analysis requires IDA tools that support users and prevent them from making errors or from using methods the wrong way.

From a practical point of view, certain restrictions have to be imposed on models obtained in a data analysis process. Thanks to the computer it is possible to create almost arbitrarily sophisticated models that fit every subtle aspect of a data set or an underlying process. Not only are such subtleties usually irrelevant in practical applications, complex models also tend to overfit the data, i.e. they fit the noise and uncertainties contained in the data. From the viewpoint of a user, a model must also be *comprehensible, interpretable* and *inexpensive.* In several application areas, e.g. medicine or financial services, security requirements demand that models can only be trusted, if they can be understood by the user. For example, an artificial neural network that was created from medical data will probably not be simply accepted as a decision authority, if it recommends an amputation based on the data of a patient. It is therefore important that IDA tools support such pragmatic approaches to model generation.

2.1 Main Features of SPIDA

Based on the requirements set out above we developed SPIDA (Soft computing Platform for Intelligent Data Analysis). Essentially, SPIDA is a data analysis tool comprising access to different data sources (text files, databases), a set of data analysis methods, data filters for pre- and post-processing and visualisation capabilities. Featured analysis algorithms are machine learning techniques like neuro-fuzzy models, neural networks, support vector machines, decision trees and fuzzy clustering as well as traditional techniques like linear regression. Filters include basic transformations like scaling or normalisation up to more sophisticated modules like outlier detection/filtering and principal component analysis.

SPIDA offers different user interfaces for the creation of data analysis processes. Firstly, experts can design IDA processes in a graphical editor. Following the data flow, the user drops blocks for data access, filters, data analysis algorithms and visualisation in co-called workspaces, see Fig. 2 on the left. Blocks will be connected and can be configured by opening dialog windows. Finally, an IDA process can be run by executing a workspace.

The Wizard is the interface for non-experts. In a sequence of dialogs the user specifies the data source, the analysis problem (prediction, clustering or dependency analysis) and preferences regarding the solution. Preferences include balancing accuracy and simplicity of a solution, desired and unacceptable accuracy (if known), if the user requires an adaptable model or an explanation like a rule set etc. Many of the preferences can be given as fuzzy terms, which are easy to understand. Based on this information, the Wizard produces a ranked list of applicable analysis methods with an indication of their suitability. More details about the fuzzy ranking mechanism can be found in [5] and about the underlying fuzzy inference mechanism in [4]. Figure 1 shows an example dialog for specifying preferences and the list of applicable models with their suitability.

The user can then choose methods from the list and ask the Wizard to create them. Creation incorporates building workspaces, i.e., picking the right pre-processing filters depending on data and analysis methods, configure the actual analysis block (setting parameters) and set up visualisation blocks. The workspace will be run automatically and the accuracy of the model and simplicity of an explanation (if exists) will be measured after execution. Taking into account these values, the suitability of the method will be updated. If accuracy or simplicity do not meet the expected figures the Wizard tries to adapt parameter settings of the respective algorithm and re-run the workspace. This procedure will be re-iterated until requirements are met or further improvements cannot be achieved. For instance, if the user required a simple explanation in terms of rules and a decision tree or a fuzzy model consists of too many rules, pruning parameters will be changed in order to force the learning algorithms to produces less rules. If accuracy is too low, the granularity of fuzzy partitions or the number of neurons or layers in a neural network can be changed, for example. Figure 2 on the right gives information about different decision trees that have been created in order to meet the user requirements. In this case a simple explanation was required, so the maximum height of the tree was gradually reduced until it was small enough.

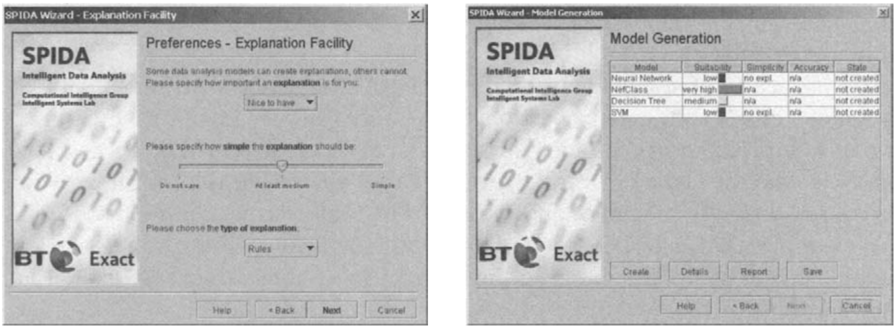
Finally, a HTML report can be generated that shows information about the data set, summarises the user requirements, the suitabilities of different methods and test results including generated explanations. Furthermore, a created workspace can be wrapped up as stand-alone application and called from other applications. For example, a user might want to learn a classifier from data and then use it in his own application to classify new data.

SPIDA is implemented as an open client/server architecture in Java. The server runs IDA processes and the client functions as a graphical user interface (GUI) and can connect both to local and remote SPIDA servers. A SPIDA server provides a plug-in API that can be used to connect basically any data analysis method or software to SPIDA. Connection is possible, for example, by direct method invocation, or by automatically creating control files that are then executed by a connected software tool in a separate thread. After a new method has been described in the knowledge base, SPIDA is ready to use it.

SPIDA has been used for several internal projects, e.g. parts have been integrated in an application for mobile workforce management, see Section 4 and [1].

3 Data Analytics in Customer Relationship Management

Customers are at the heart of most businesses, in particular, service providers. In order to serve their customers better, they have to introduce reliable processes and procedures for the interaction with their customers, which is


SPIDA Wizard - Explanation Facility
SPIDA
Intelligent Data Analysis
Computational Intelligence Group
Intelligent Systems Lab
BT Exact
Help
< Back
Next
Cancel
SPIDA Wizard - Model Generation
Create
Details
Report
Save

Preferences - Explanation Facility

Some data analysis models can create explanations, others cannot. Please specify how important an **explanation** is for you:

Nice to have

Please specify how **simple** the **explanation** should be:

Do not care — At least medium — Simple

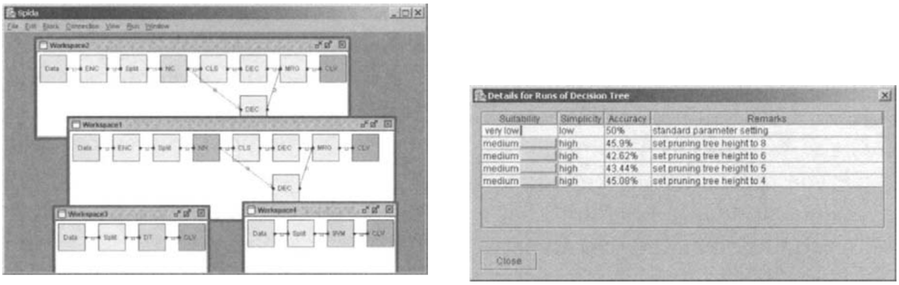
Please choose the **type of explanation**:

Rules

Model Generation

Model	Suitability	Simplicity	Accuracy	State
Neural Network	low	no expl.	n/a	not created
NefClass	very high	n/a	n/a	not created
Decision Tree	medium	n/a	n/a	not created
SVM	low	no expl.	n/a	not created

Fig. 1. Choosing preferences in the Wizard, left: the dialog for an explanation facility of methods, right: ranking of data analysis methods according to user preferences before model creation



Suitability	Simplicity	Accuracy	Remarks
very low	low	50%	standard parameter setting
medium	high	45.9%	set pruning tree height to 8
medium	high	42.62%	set pruning tree height to 6
medium	high	43.44%	set pruning tree height to 5
medium	high	45.08%	set pruning tree height to 4

Fig. 2. Left: Workspaces automatically created by the wizard for four different data analysis methods. The workspaces are usually not shown by the wizard. This view also represents the GUI an IDA expert would use to configure data analysis processes manually. Right: automatic re-runs of the decision tree module

known as Customer Relationship Management (CRM). Typically, organisations collect and keep customer data as part of their processes. Therefore, data forms the core information source for CRM. Additionally, customer surveys are being conducted to listen to the voice of the customers, aiming at improving customer satisfaction and loyalty. In order to achieve improvements, first the main drivers of satisfaction, loyalty or other important indicators have to be identified and then processes must be adapted in such a way that the main drivers are positively influenced. For further discussion, we focus on analysing customer satisfaction based on survey data. The techniques described below, however, can obviously be applied to many other problems.

In order to measure customer satisfaction BT regularly conducts customer surveys, for example by interviewing customers or in form of online questionnaires. The survey results are analysed statistically to highlight customer satisfaction in different areas. Those reports can pinpoint areas where customer satisfaction must be improved, but they do not provide information about dependencies between different influence factors and they cannot be used as a planning tool.

A typical statistical analysis looks at the frequency distributions of all possible replies to a given question. From that we can learn in which areas we have a problem in customer satisfaction. We can then drill down and look at the replies of certain customer groups based on demographical information or the replies to particular questions. From that we may be able to learn, for example, that users that complain about the layout of our web portal are more likely to complain about not finding the hyper-links than customers that are happy with the layout.

Another typical analysis is to look at correlations between the replies to different questions. Often, we find that many questions are highly correlated meaning the higher the satisfaction or dissatisfaction in one area the higher the satisfaction or dissatisfaction in another area is likely to be. A correlation analysis, however, assumes a linear relationship and its result is therefore only relevant, if the assumed linear dependence actually exists.

In order to understand the quantitative influence of the result for one question on the overall satisfaction we can use a functional model. In statistical analysis we typically use linear regression for this purpose. However, linear regression assumes that the individual variables (questions of the survey) are independent of each other and that a linear dependency between the independent variables and the overall satisfaction level actually exists. Because we can typically not uphold the independence assumption we can first minimise the correlation of inputs to the regression analysis with a principal component analysis.

A regression model can provide some insight of how a selected number of independent variables (questions, drivers of (dis)satisfaction) influence a particular dependent variable (overall satisfaction level) in a linear fashion. However, the model cannot model the mutual dependencies between the variables and thus cannot take compensatory or reinforcing effects into account.

If we want to understand the effects changes in a particular set of drivers of (dis)satisfaction have not only on overall satisfaction levels but also on all the other drivers then we can use a multi-dimensional probabilistic model which is not restricted by linear dependencies or global independence assumptions. A suitable probabilistic model is a Bayesian network that can represent arbitrary probabilistic influences between any number of variables.

As a software solution to tackle the problems mentioned above, we have developed iCSat – a platform for intelligent customer satisfaction modeling. The system offers Bayesian networks and linear statistics to analyse customer surveys or any other table-based, structured data. The Bayesian module analyses the dependencies between all satisfaction indicators and automatically learns a probabilistic model of customer satisfaction based on Bayesian networks. At the core of iCSat is an algorithm to learn the structure of Bayesian networks which we implemented following [3], and the commercial library NETICA by

Norsys Software Corporation to represent the network, learn the probabilities from data and propagate probabilities through the network.

In order to enable business users who are not familiar with Bayesian networks to analyse data iCSat comes with a simple user interface that hides as much of the technology as possible and focuses on the solving business problems instead. Regarding network generation it allows the user to load and edit questionnaires, data, and build Bayesian networks (learning structure and probabilities). Afterwards, the model can be analysed using functionality like

- sensitivity analysis: which answers influence others to which degree
- what-if analysis: experiment with the potential answers to a particular question or set of questions and observe the predicted impact on the rest of the survey
- compare the impacts of different what-if scenarios

iCSat is being used to plan changes in the areas of selected satisfaction indicators and to understand the influence on other indicators. This allows users to plan projects for improving customer satisfaction and to tackle the most promising indicators for improvement. Since iCSat is using Bayesian networks, it can be applied to many other problems that involve multi-dimensional probabilistic relationships.

4 Data Analytics in Resource Management

Any organisation with a large mobile workforce needs to ensure efficient utilisation of its resources as they move between tasks distributed over a large geographical area. BT employs around 15 000 engineers in the UK who provide services for business and residential customers such as network maintenance, line provision and fault repairs. In order to manage its resources efficiently and effectively, BT uses a sophisticated dynamic scheduling system to build proposed sequences of work for field engineers. A typical schedule for a field engineer contains a sequence of time windows for travel and task. To generate accurate schedules the system must have accurate estimates for the time taken between consecutive tasks (travel time) and estimates for task duration.

We have implemented a system called ITEMS (Intelligent Travel time Estimation and Management System) that substantially improves the accuracy of travel time estimates compared to the previous system. Where the old system was based on manual definitions of travel times using routing software and local knowledge, which was difficult and labour-intensive to maintain and update, ITEMS is based on data collected from technicians and automatically updates estimates on a daily basis. Also, under the old system, many engineers, mainly due to underestimation of travel time, were not able to arrive on time for their next task resulting in knock-on effects and inefficient schedules.

Note that the travel time is calculated as the difference between the time a task is issued and the arrival time on site. This comprises the time required to leave the current site, walk to the car park, start the car, drive to the destination site, park the car, and arrive at the doorstep of the next customer. In most cases the largest part of the travel time consists of driving from one site to another and the car park is within the premises of the site. Unfortunately this may not be the case in urban areas such as London, or city centres, where the engineers may require a substantial amount of time to find a place to park and to walk from the car park to the customers premises. Consequently, routing software that typically estimates pure travel times is not suitable for this problem.

Travel time is typically treated as a fixed overhead when scheduling jobs and is extremely difficult to quantify, because factors such as road conditions, weather, vehicle type, route disruption, driving behaviour, traffic peak periods, etc. all contribute to journey times, making it difficult to prescribe an expected inter-job journey time.

ITEMS was customised such that it can interface to BT's proprietary Work Manager (WM) system which manages the technicians' work schedules amongst other things. Generally, WM estimates the travel time of a journey by dividing the straight line distance between start and end point by expected speed on the route. The expected speed value is called General Travel Factor (GTF) and usually differs for different regions. Each region is further divided into smaller areas around exchange buildings. For a route between any two exchange areas in a region WM can hold a special Travel Factor (TF) as some kind of exception from the GTF. These TFs are stored in a table that keeps the TF for each pair of exchange areas. That means in particular that the TF is based only on start and end point of a journey but not on other information like time of day or day of week. Additionally, a second value called NRT (Non-Road Travel time) can be defined which is basically the offset in the linear regression equation. As the name suggests, it can be interpreted as non-travel time for parking, accessing premises etc. since it does not depend on the distance.

In order to find a good estimation model we have experimented with different approaches like linear regressions, neural networks, neuro-fuzzy techniques and regression trees. Table 1 shows the performance of four approaches, whereby the first two are linear regressions using only start and end location as required by WM. The difference between these regressions is the way we aggregated the data. For the neural network and the neuro-fuzzy model we additionally took into account the time of day, the day of week and the number of job (like first, second, third job of the day). The root mean square error (RMSE) is not necessarily an indicator for prediction accuracy, which is measured by a ±15min window. This is due to the non-normal distribution of travel time.

Additional to the estimation of travel times, ITEMS features a web-based interface for exploring actual travel patterns. It supports resource managers

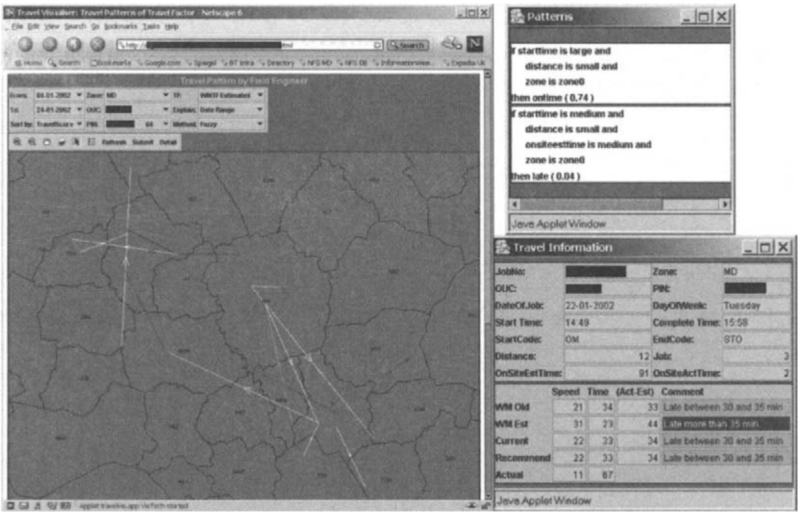
Table 1. Performance of different inter-job time estimators

Model	RMSE	Accuracy
TF + NRT	19.6	67.5
Lin. Regression	18.2	64
Neural Net	16.1	69.6
Neuro-Fuzzy	14.1	74

in analysing travel patterns in order to improve job schedules and distribute resources.

Figure 3 shows travel patterns of a particular engineer. Each line on the map represents a journey, the colour of the line the actual travel time compared to the estimated one. For each of the journeys detailed travel and job information can be shown by clicking the respective line. The interface allows the user to select date ranges for the journeys, the region, units and technicians. In this way, the user can drill down to the desired level of detail. Other views are available that show journeys by day of the week or time of the day.

The window at the top right displays fuzzy rules that match the selected journey. The degree of match tells the user how useful a rule is to explain the travel pattern, i.e. why the technician was late, early or on time. It provides resource managers with means to generally understand travel patterns as the rules summarise the data in a way that is easy to understand. Furthermore, it tells managers if a specific pattern is common or not. ITEMS induces decision trees and Neuro-Fuzzy systems (NEFCLASS, see [2]) from data on the fly in order to generate the rules.


Travel Visualiser: Travel Patterns of Travel Factor - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://[redacted]html
Search
Home Search Bookmarks Google.com Spiegel BT Intra Directory NFS MD NFS OB Informationview Expedia UK
Travel Pattern by Field Engineer
Refresh Submit Detail
Patterns
Java Applet Window
Travel Information
Java Applet Window

From:	04-01-2002	Zone:	MD	TF:	WMTF Estimated
To:	24-01-2002	OUC:	[redacted]	Explain:	Date Range
Sort by:	TravelScore	PIN:	[redacted] 64	Method:	Fuzzy

If starttime is large and
distance is small and
zone is zone0
then ontime (0.74)

If starttime is medium and
distance is small and
onsiteesttime is medium and
zone is zone0
then late (0.04)

JobNo:	[redacted]	Zone:	MD
OUC:	[redacted]	PIN:	[redacted]
DateOfJob:	22-01-2002	DayOfWeek:	Tuesday
Start Time:	14:49	Complete Time:	15:58
StartCode:	OM	EndCode:	STO
Distance:	12	Job:	3
OnSiteEstTime:	91	OnSiteActTime:	2

	Speed	Time	(Act-Est)	Comment
WM Old	21	34	33	Late between 30 and 35 min
WM Est	31	23	44	Late more than 35 min
Current	22	33	34	Late between 30 and 35 min
Recommend	22	33	34	Late between 30 and 35 min
Actual	11	67		

Fig. 3. Visualisation of actual journeys, details of a single journey, and rules that explain the pattern

ITEMS is operational for a couple of years now. Analysis shows that the travel time predictions are 7-12% better than with the old systems. Due to improved job scheduling, this typically resulted in a 5-10% reduction in average distance per journey and average travel time. The consequences are considerable cost savings for BT every year, increased customer satisfaction since appointments can be planned better and improved satisfaction of the technicians, because the estimates for their journeys are more realistic.

5 Conclusions

Many organisations are still far from leveraging data to their advantage. An important reason is that most available data analysis tools are still to difficult to use. Part of our research is therefore dedicated to the question of how we can take the latest research results to the business user by hiding technology behind simple user interfaces.

As successful examples, we presented three of our applications that support business users at different levels. The data analysis platform SPIDA enables non-experts to analyse data using modern machine learning techniques without the need to understand them. iCSat has been very successful within BT as a tool for understanding drivers of customer satisfaction because of its intuitive interface. It is now being used in other problem spaces, as well. Finally, ITEMS is a good example for a data analysis and exploration system that has been integrated in an operational system and is therefore used on a daily basis. All three platforms feature state-of-the-art machine learning or data analysis algorithms which are rarely seen in commercial applications.

Soft computing helped in all applications since we quite often deal with uncertain or vague information. The use of fuzzy techniques is a very natural way of presenting information to business users and also helps them communicate their knowledge or requirements as in SPIDA.

References

1. C. Ho, B. Azvine, D. Nauck, and M. Spott. An intelligent travel time estimation and management tool. In *Proc. of 7th European Conference on Networks and Optical Communications (NOC 2002)*, pages 433–439, Darmstadt, 2002.
2. Detlef Nauck. Fuzzy data analysis with NEFCLASS. *International Journal of Approximate Reasoning*, 32:103–130, 2002.
3. M. Singh and M. Valtorta. Construction of bayesian network structures from data: a brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12:111–131, 1995.
4. M. Spott. Efficient reasoning with fuzzy words. In *Proc. of FSKD 2002*, Singapore, 2002.
5. M. Spott and D. Nauck. On choosing an appropriate data analysis algorithm. In *Proc. IEEE Int. Conf. on Fuzzy Systems 2005*, 2005.

Neural Network-Based Expert System to Predict the Results of Finite Element Analysis

Onkar Pradeeprao Bhise[1] and Dilip Kumar Pratihar[2]

[1] PG student
opbhise@yahoo.com
[2] Associate Professor
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur-721 302
West Bengal, India
dkpra@mech.iitkgp.ernet.in

Summary. Realizing the fact that the performance of a finite element (FE) analysis depends on the type of elements, mesh topology, mesh density, node numbering and others, an attempt is made in the present work, to develop a neural network-based expert system to predict stress analysis results of an FEM package, within a reasonable accuracy. A rubber cylinder is compressed diametrically between two plates, whose induced stresses and deformed shape are to be determined using an FE analysis. By varying two parameters, namely element size and shape ratio, results (obtained through FE analysis) in terms of induced stress and deformed shape of the cylinder are recorded, which are utilized to train a neural network (NN)-based expert system, by using a back-propagation algorithm and a genetic algorithm, separately. Results of two NN-based expert systems are compared, in terms of accuracy in prediction of the results. It is interesting to note that the expert system can predict the results within a fraction of a second, whereas an actual FE analysis may take several seconds depending on the complexity of the problem.

Keywords: Expert system, FEM, Neural network, Back-propagation, Genetic Algorithm.

1 Introduction

Expert system (ES) is nothing but a computer program developed by injecting the human reasoning capability in it, which can achieve an expert-level competence for solving the problems in a specific area. The performance of an ES depends on its knowledge base (KB), the development of which is the prime task, before using it for a quick answer.

The successful application of an FEM package requires a substantial amount of experience and expertise to make the computation feasible and the results accurate at the same time. It is so, due to the fact that there is an enough fuzziness in the performance of an FEM package, which depends on the selection of element type, element size, mesh density, mesh topology, node numbering, and others. Thus, design and development of an ES is necessary, for an efficient use of an FEM package, to solve a particular physical problem.

Several attempts were made by various researchers to analyze the fuzziness of an FEM package and for development or modification of FEM programs, so as to reduce the computational time and increase the accuracy. Umar and Abbas [1] studied the error in FEM results, due to the modification of mesh. They presented an extrapolation technique to predict the results more precisely by carrying out the analysis for two mesh configurations, which have a different number of elements in each case, but possess a similar discretization pattern. As the accuracy of an FEM analysis depends on mesh refinement, Kittur and Houstan [2] discussed about the mesh refinement criteria for stress analysis. A step-by-step procedure for mesh selection and refinement was presented. Manevitz and Givoli [3] argued that FEM serves as a rich test bed appropriate for the serious use of soft computing techniques and had developed an initial framework of an intelligent FEM package. The areas where expert knowledge is essential while using the FEM were listed and some possible soft computing techniques were suggested. Rank and Babuska [4] developed an ES framework, which provides the domain knowledge for the optimal mesh design. To find a proper combination of mesh refinement and polynomial degree, they implemented mathematical and heuristic rules in the finite element ES. Manevitz and Malik [5] used neural networks (NNs) to the problem of mesh placement for FEM. They adopted the self-organizing algorithm of Kohonen, to solve the problem of automatically assigning the coordinates from a 2-D domain to a given topological grid of nodes. Porto [6] investigated three potential NN training algorithms, namely back-propagation, simulated annealing and evolutionary programming in processing active sonar returns. He claimed that although all three methods can be used for processing the sonar results, the stochastic methods of simulated annealing and evolutionary programming can outperform the back-propagation. For an efficient analysis, a lot of research is yet to be done, to develop an intelligent FEM package.

In the present work, an attempt is made to predict FE analysis results for a particular problem, by developing an NN-based expert system. In this paper, we have developed the KB of an expert system, through modeling the fuzziness in FE analysis, due to variation in element size and shape ratio, keeping the other parameters unaltered. Moreover, a comparison is made of different learning algorithms for an NN.

The rest of the text is organized as follows: Section 2 explains the mathematical formulation of the problem and its possible solution using an FE analysis. Section 3 describes the proposed techniques to develop NN-based ES. Results are discussed in Section 4 and some concluding remarks are made in Section 5. Section 6 indicates the scope for future work.

2 Mathematical Formulation of the Problem and Its Solution Using an FE Analysis

The fuzziness of an FEA is studied by taking the example of rubber cylinder compression problem [7]. The problem is defined as follows: A long rubber cylinder is pressed between two rigid plates to a maximum imposed displacement $\delta_{max} = -0.2$ m (refer to Fig. 1). We will have to determine the induced stress and deflected shape of the section.

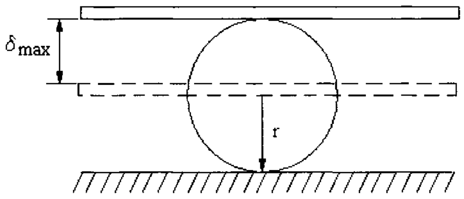


Fig. 1. Sketch of the rubber cylinder compression problem.

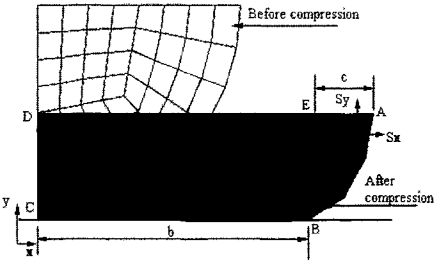


Fig. 2. Model for finite element analysis.

2.1 Results of ANSYS 7.0 Package

It is a static analysis problem, in which we have used Mixed U-P Hyper-elastic solid 2D-4 node element for representing the rubber material. The Young's modulus and Poisson's ratio are taken to be equal to 2.82 MPa and 0.49967, respectively. The Monney-Rivlin constants C_{10} and C_{01} are assumed to be equal to 0.293 and 0.177 MPa, respectively. A rubber cylinder of radius $r = 0.2$ m is subjected to a maximum compression $\delta_{max} = -0.2$ m (it is downward displacement, hence negative sign is shown). A plane strain solution is assumed based on the geometry of the problem. Due to geometric and loading symmetry, the analysis is performed using one quarter of the cross section (refer to Fig. 2). All nodes on the left edge (X = 0) are constrained, UX = 0. All nodes on the top edge (Y = 0) are coupled in UY. An imposed displacement of -0.1 m acts upon the coupled nodes. The quarter of cylindrical cross section is divided into 70 approximately equal areas. We can use either triangular or quadratic elements for meshing of each small area. Shape ratio is defined as the ratio of the number of small regions consisting of triangular elements only to the total number of small regions, i.e., 70. It is to be noted that the number of elements present in the small area (either triangular or quadrilateral) after meshing, depends on the element size.

2.2 Basics of FEM Formulation

We will discuss the mathematical model used by Khandia [8], to solve the problems involving rubber or rubber-like materials using an FEM.

Mathematical model based on Mooney-Rivlin theory for a rubber-like material

Let us consider a rectangular block of elastomeric material having L_1, L_2 and L_3 as length along three axes and λ_1, λ_2 and λ_3 denote the length ratios along the edges of the block. For a Mooney-Rivlin material, the stress values can be determined as the following.

$$\sigma_1 = \frac{2}{\lambda_1^3}[(C_{10} + C_{01})\lambda_1^4 - (\frac{C_{10}}{\lambda_2^2} - C_{01})] \tag{1}$$

$$\sigma_2 = \frac{2}{\lambda_2^3}[(C_{10} + C_{01})\lambda_2^4 - (\frac{C_{10}}{\lambda_1^2} - C_{01})] \tag{2}$$

$$\sigma_3 = 0 \tag{3}$$

where C_{10}, C_{01} are the constants.

The deformed shape after compression, as represented by the curve AB (refer to Fig. 2), can be expressed as $y = \frac{0.1}{c^2}x^2$, after shifting the origin from point C to point B and applying the end conditions – (i) y = 0 at x = 0 and (ii) y = 0.1 at x = c.

2.3 Results of FE Analysis Using ANSYS 7.0 Package

Two parameters - shape ratio and element size are varied in their respective ranges by keeping the other parameters fixed and results are noted while carrying out FE analysis on the said problem, by using ANSYS 7.0 package. It is important to mention that only two input parameters have been considered in the present work, for simplicity but the performance of an FE analysis depends on some other parameters, namely mesh density, mesh topology, node numbering etc., which have been kept unaltered. The package is run for 275 different combinations of element size and shape ratio, by varying them within the ranges of (0.003 to 0.015) m and (0 to 1), respectively. The outputs, i.e., maximum induced stress and deformed shape are recorded, for each of these 275 different combinations. It is interesting to note that the maximum values of compressive and tensile stresses along X-direction are found to vary in the ranges of (1.456 to 22.837) MPa and (1.355 to 17.784) MPa, respectively. Similarly the maximum compressive and tensile stresses along Y-direction are seen to vary in the ranges of (3.61 to 24.908) MPa and (0.244346 to 16.185) MPa, respectively. The variations are also observed in the values of the parameters - b and c, which are used to determine the deformed shape of the rubber cylinder.

3 Proposed Techniques to Develop NN-based ES

The performance of an NN depends on its architecture, connecting weights, bias values, activation functions, and others. A proper training (learning) is to be provided to the network, so that it can predict the output accurately and thus, reduce the error in prediction. In this paper, NN-based expert systems have been developed by using both back-propagation as well as genetic algorithm-based learning algorithms, which are discussed below.

3.1 Approach 1: NN-based ES Using Back-propagation Learning Algorithm

We consider a back-propagation NN [9] consisting of three layers, namely input, hidden and output layers. Both the input as well as output layers contain the fixed number of neurons, whereas the number of neurons can be varied in the hidden layer. Back-propagation algorithm works based on the principle of a steepest descent method and it is nothing but a supervised learning, where the task is to learn to map the input vectors to desired output vectors. The back-propagation algorithm modifies feed-forward connections between the input and hidden units, and the hidden and output units, so that when an input vector is presented to the input layer, the output layer's response becomes the desired output vector. During training, the error caused by the difference between the desired output vector and the output layer's response to an input vector propagates back through connections between layers and adjusts appropriate connection weights, so as to minimize the error. Initially, the weights are created at random. Depending on the weights and transfer functions, a set of outputs of the network is determined, for a set of inputs. Each output is then compared to its corresponding target value to calculate the error as $E = \frac{1}{2}(T-O)^2$, where T and O represent the target and calculated values, respectively. This error E is then back-propagated to modify the weight values, so that it becomes minimum. The weights are updated in each backward pass as $\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) - \eta \frac{\partial E(n)}{\partial W_{ji}(n)}$, where $\Delta w_{ji}(n)$ is the correction applied to the synaptic weight connecting output of neuron i (lying in a layer) to the input of neuron j (lying on another layer), at iteration n, α denotes the momentum term, η represents the learning rate.

The back-propagation algorithm is simple to implement and computationally efficient in the sense that its complexity is linear in synaptic weights of the network. However, a major limitation of the algorithm lies in the fact that being a gradient descent method, it is plagued by the local minima problem.

3.2 Approach 2: NN-based ES Using GA-based Learning Algorithm

Genetic algorithms (GAs) are population-based search and optimization techniques, which work based on the mechanics of natural genetics and natural selection [10]. As it is a population-based technique, the chance of its solutions for getting trapped into the local minima, is less. We have used a binary-coded GA, in which 10-bits are assigned to represent each of the weight values, the bias value and the constant of each activation function. The fitness of a GA-string is calculated as $F = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{T_i - O_i}{T_i}\right| \times 100$, where N represents the total number of training cases.

4 Results and Discussion

Training data are collected by solving the said physical problem using an ANSYS package, and noting down six different outputs for each of 275 input combinations involving two parameters, namely element size and shape ratio. For the development of an NN-based ES, the batch mode of training is adopted, using both back-propagation as well as a GA, separately. Results of these two NN-based ESs are compared among themselves and with those of ANSYS package.

Initially, we tried to develop a back-propagation NN, which can determine all six outputs together, but it was unable to predict all the outputs within a reasonable accuracy limit. Hence, we have developed a separate NN, for predicting each output. Thus, we have six different BPNNs for predicting six required output values, namely maximum compressive stress along X and Y - directions, maximum tensile stress along X and Y - directions and deformation parameters – b and c.

To determine a set of optimal parameters for BPNN, experiments are carried out by varying one parameter at a time and keeping the others unaltered. The similar experiments are also carried out for the remaining five outputs. Table 1 shows the sets of optimal parameters for all six BPNNs.

Table 1. Optimal parameters for six BPNNs.

Output	n	a_j	a_k	η_j	η_k	α
Sx (C)	8	47.0	23.0	0.1	0.04	0.84
Sy (C)	8	33.0	12.9	0.07	0.04	0.84
Sx (T)	10	38.8	10.2	0.07	0.06	0.84
Sy (T)	10	38.8	10.2	0.13	0.05	0.84
b	8	18.5	7.9	0.13	0.12	0.78
c	7	26.0	5.0	0.12	0.08	0.81

Table 2. Optimal GA-parameters for six different GA-NNs.

Output	n	p_m	Pop-size	Gen.
Sx (C)	10	0.006	100	1184
Sy (C)	8	0.006	100	674
Sx (T)	5	0.008	100	953
Sy (T)	7	0.006	100	738
b	10	0.008	100	733
c	6	0.006	100	994

In Approach 2, the back-propagation algorithm of NN is replaced by a GA, whose fitness is defined as the average absolute percentage error in prediction. In GA, we have used a tournament selection scheme, uniform crossover with a probability of 0.5, and a bit-wise mutation. Initially, the GA-string representing the connecting weights, bias value, constants of the activation functions of the hidden and output layers are generated at random. These strings are further modified by using different operators. It is to be noted that the connecting weights, bias value, and the sigmoidal constant for hidden and outer layers are generated in the ranges of (-1.0 to +1.0), (-0.005 to +0.005), and (1 to 71), respectively.

Table 2 shows the results of parametric study for six GA-trained NNs developed for carrying out the stress analysis of the said problem.

Table 3. Results of two NN-based ESs

Type of NN	Error	Sx (C)	Sy (C)	Sx (T)	Sy (T)	b	c
BPNN	Avg. abs. % error	15.35	12.98	31.70	79.08	1.00	3.71
BPNN	Mean	-0.22539	-0.21296	-0.32215	0.45891	0.00072	-0.00056
BPNN	Std. dev.	1.53707	1.60641	1.94444	1.77641	0.00387	0.00406
GA-NN	Avg. abs. % error	19.69	16.56	23.76	36.03	1.06	3.93
GA-NN	Mean	0.32549	-0.10370	0.60102	-0.50972	0.00088	-0.00143
GA-NN	Std. dev.	2.14064	2.01182	1.62068	1.43949	0.00418	0.00444

Results of the test cases (refer to Table 3) indicate that both the NN-based ESs are following the trend of ANSYS results. The performances of these two ESs for a particular test case, are shown in Figs. 3 through 8. Comparisons are made of the two NN-based ESs, in terms of their accuracy in prediction.

We can observe from Table 3 that BPNN has shown better performance compared to the combined GA-NN, in the case of predicting the compressive stress values and deformation parameters. On the other hand, GA-NN has proved its supremacy over the BPNN, while predicting the tensile stress values. It could be due to the fact that input-output relationships, in the case of compressive stress and deformation parameters may not be so much non-linear, as in the case of that of the tensile stress. The combined GA-NN may have a wider search space compared to that of the BPNN but the former could be slower compared to the latter, in terms of its search speed. Thus, in case of predicting the compressive stress values and deformation parameters, BPNN has reached the optimal solutions much earlier compared to the GA-NN, as the input-output relations may not be so much non-linear in these cases. On the other hand, GA-NN has shown the better performance compared to BPNN, in case of predicting the tensile stresses and it could be due to the reason that the input-output relationships are highly non-linear, which could not be modeled accurately using the BPNN. Moreover, the performance of both BPNN and GA-NN are dependent on their learning patterns. The aim of providing the learning is to minimize the error in prediction, i.e., to reach the minimum point of the error surface. Thus, the performance of BPNN and GA-NN will depend on the nature of the error surface, which is

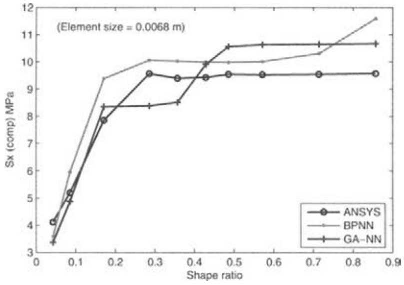


Fig. 3. Prediction of Sx (comp.)

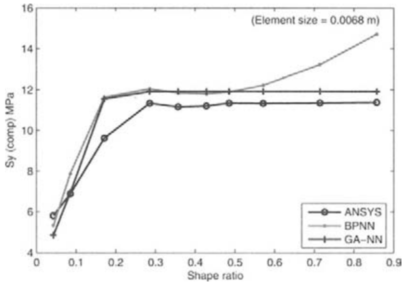


Fig. 4. Prediction of Sy (comp.)

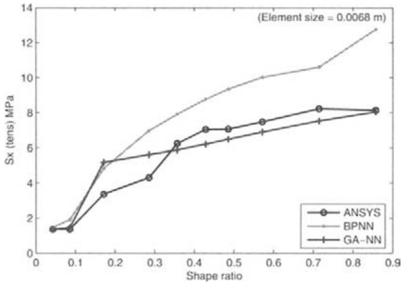


Fig. 5. Prediction of Sx (tension)

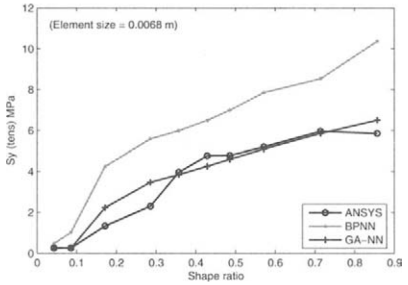


Fig. 6. Prediction of Sy (tension)

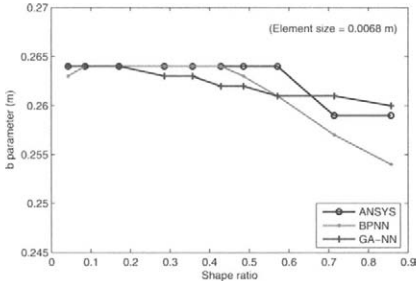


Fig. 7. Prediction of b parameter

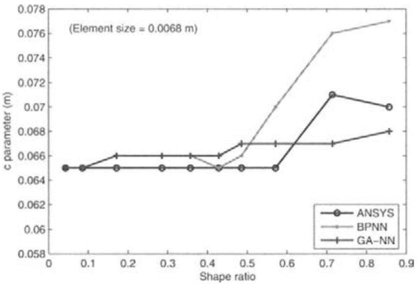


Fig. 8. Prediction of c parameter

multi-dimensional in nature. If the error surface is uni-modal in nature, BPNN will perform better than the GA-NN, but for a more complicated error surface, such as multi-modal etc., GA-NN may perform better than the BPNN.

Two NN-based ESs are compared in terms of their computational complexity, while carrying out training using the same scenarios, on a P-IV PC. The BPNN requires a slightly more computational time per iteration, compared to the GA-NN, because BPNN in each iteration performs both the forward-propagation as well as backward-propagation, which makes it computationally a slightly more expensive per iteration. On the other hand, only the forward-propagation of the NN is involved, while developing the combined GA-NN. However, BPNN is found to be computationally faster than the combined GA-NN, for achieving the same accuracy. It may be due to the fact that BPNN follows the steepest descent algorithm.

To compare the performances of two NN-based ESs with that of ANSYS package, two aspects, namely accuracy in prediction of results and computational complexity are considered. Regarding the accuracy, it has been observed that both the NN-based ESs are following the trend of ANSYS results. Two NN-based ESs are compared with the ANSYS package, in terms of their CPU time also. Fig. 9 shows the results of this comparative study, which indicates that the CPU time of both the NN-based ESs is independent of their parameters, whereas the CPU time of ANSYS package is highly dependent on the parameters. For example, as the element size decreases, the CPU time of the ANSYS package will go on increasing. As the CPU time of the ESs is only a fraction of a second, they might be suitable for making on-line prediction of stress analysis results.

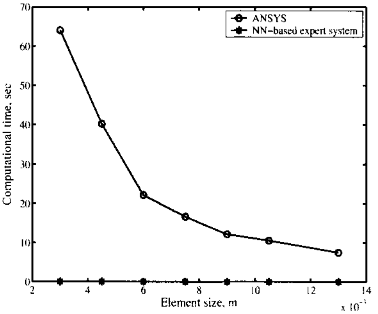


Fig. 9. Computational time vs. element size for ANSYS package and NN-based ESs.

5 Concluding Remarks

From the above study, concluding remarks have been made as follows.

- Fuzziness in the performance of an FEM package is established from the variations of stress analysis results given by it. Both the NN-based ESs are able to predict the stress analysis results, within a reasonable accuracy limit.
- Expert system based on BPNN has performed better than GA-NN approach for predicting most of the outputs but not all. When the input-output relationship is highly non-linear, the GA-learned NN has performed

better than BPNN. The performances of BPNN and GA-NN depend on the nature of error surface, which is to be minimized during their learning.

- Computational complexity of the ANSYS package is dependent on its input parameters. On the other hand, CPU time of the NN-based ESs is independent of their input parameters and they are computationally much faster than the ANSYS package.
- BPNN is computationally slightly more expensive than the GA-NN, as far as the computational complexity per iteration is concerned. However, BPNN is seen to be computationally faster than the GA-NN, for achieving the same accuracy.

6 Scope for Future Work

In the present work, we have modeled the fuzziness of an FEM package by varying the element size and shape ratio. In an FE analysis, fuzziness also depends on the selection of mesh density, mesh topology, node numbering, and others. A suitable model may be developed by considering all these parameters together. Moreover, a GA with varying string length might be used in future, to optimize the NN considering its variable structure.

References

1. Umar A, Abbas H (1996), Prediction of error in finite element results, *Computers and Structures*, 60: 471–480.
2. Kittur M G, Houstan R L (1990), Finite element mesh refinement criteria for stress analysis, *Computers and Structures*, 34: 251–255.
3. Manevitz L M, Givoli D (2003), Towards automating the finite element method: A test bed for soft computing, *Applied Soft Computing*, 3: 37–51.
4. Rank E, Babuska I (1987), An expert system for the optimal mesh design in the hp-version of the finite element method, *International Journal for Numerical Methods in Engineering*, 24: 657–682.
5. Manevitz L M, Malik Y (1997), Finite element mesh generation using self-organizing neural networks, *Microcomputers in Civil Engineering*, 12: 233.
6. Porto V W (1995), Alternative neural network training methods, *IEEE Expert: Intelligent Systems and Their Applications*, 22: 16–22.
7. ANSYS 7.0 Verification Manual, Vol. 6.
8. Khandia Y A (1990), Finite element analysis of rubber problems, *IOP Short Meetings of the Stress Analysis Group of the Institute of Physics - London*, 24: 11–31.
9. Haykin S (2001), *Neural Networks* Pearson Education (Singapore) Pvt. Ltd., New Delhi, India.
10. Goldberg D E (1989), *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, Mass., USA.

Part VI

Identification and Forecasting

Modular Neural Networks with Fuzzy Integration Applied to Time Series Prediction

Patricia Melin, Ileana Leal, Valente Ochoa, Luis Valenzuela, Gabriela Torres, Daniel Clemente

Dept. of Computer Science, 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 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.

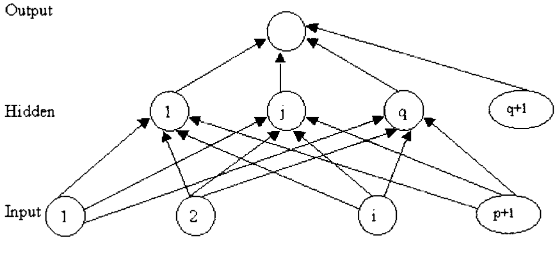


Fig. 1. Single hidden layer feedforward network.

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

$$Y_t = \sum_{j=1}^{p+1} \pi_t x_{it} + \sum_{j=1}^{p+1} \theta_j F\left(\sum_{i=1}^{p+1} w_{ij} x_{it}\right) \tag{1}$$

Here π_t 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:

$$E_l = \tfrac{1}{2} \sum_{l=1}^{n} (d_l - y_l)^2 \tag{2}$$

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.

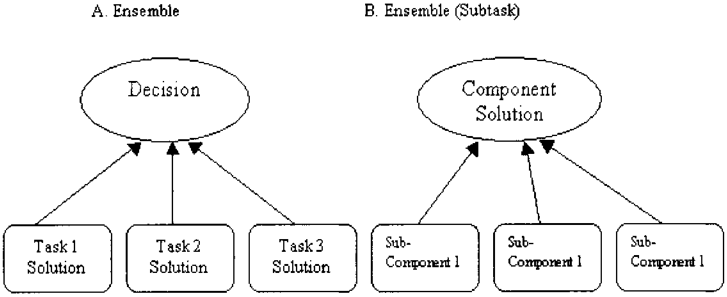


Fig. 2. Ensembles for one task and subtask.

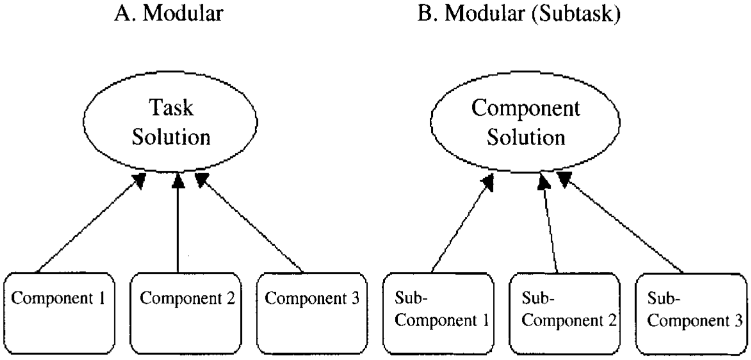


Fig. 3. Modular approach for task and subtask.

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, $D_k(z)=G_k(A_1(z), A_2(z), \ldots ,A_n(z))$. 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 $h:X\rightarrow[0,1]$ 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,

$$\begin{aligned} h(x) \circ g\,(x) &= \max_{E\subseteq X} [\,\min (\,\min_{x\in E} h(x), g(E))] \\ &= \sup_{\alpha\in[0,1]} [\min(\,\alpha\,, g(h_\alpha\,))] \end{aligned} \tag{6}$$

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.

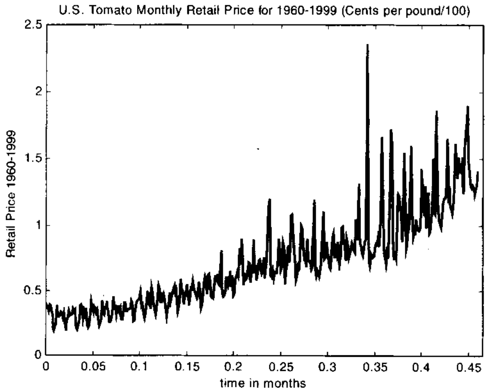


Fig. 4. Prices in US Dollars of tomato from January 1960 to December 1999.

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.

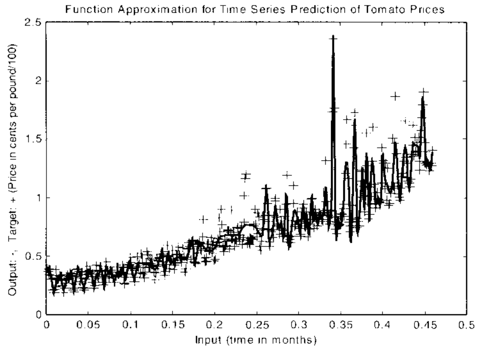


Fig. 5. Modular network for tomato prices with Levenberg-Marquardt algorithm.

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.

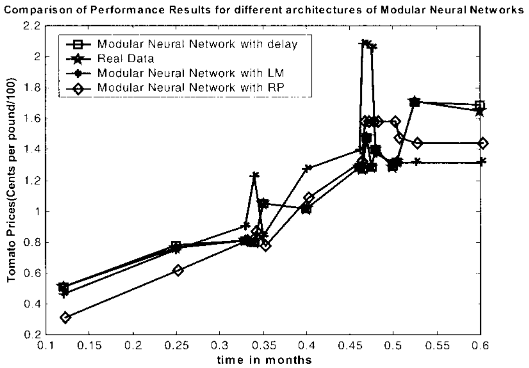


Fig. 6. Comparison of performance results for several modular neural networks.

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.

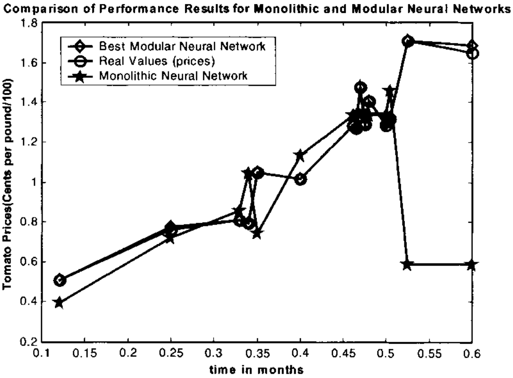


Fig. 7. Comparison of monolithic and modular neural networks.

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.

Fuzzy Model Identification for Rapid Nickel-Cadmium Battery Charger through Particle Swarm Optimization Algorithm

Arun Khosla[1], Shakti Kumar[2], K.K. Aggarwal[3] and Jagatpreet Singh[4]

[1] Dr B R Ambedkar National Institute of Technology, Jalandhar – 144011. India. khoslaak@nitj.ac.in
[2] Haryana Engineering College, Jagadhari – 135003. India.
[3] GGS Indraprastha University, Delhi – 110006. India. kka@ipu.edu
[4] Infosys Technologies Limited, Chennai – 600019. India. jagatpreet@yahoo.com

This paper presents the fuzzy model identification for rapid Nickel-Cadmium (Ni-Cd) battery charger by applying Particle Swarm Optimization (PSO) algorithm on the input-output data. Models generated through this approach provide the flexibility of black-box approach like neural networks, since it does not need to know any information regarding the process that generates the data. The PSO method is a member of the broad category of swarm intelligence techniques for finding optimized solutions. The motivation behind the PSO algorithm is the social behavior of animals viz. flocking of birds and fish schooling and has its origin in simulation for visualizing the synchronized choreography of bird flock. The data for the batteries charger was obtained through experimentation with an objective to charge the batteries as fast as possible. The implementation of the approach is described and simulation results are presented to illustrate its effectiveness.

1 Introduction

The problem of fuzzy system modeling or fuzzy model identification is generally the determination of a fuzzy model for a system or process by making use of linguistic information obtained from human experts and/or numerical information obtained from input-output numerical measurements. The former approach is known as knowledge-driven modeling while the later is known as data-driven modeling. It is also possible to integrate the two approaches for developing models of complex real systems. In this paper, the fuzzy model for rapid Ni-Cd batteries charger has been identified through PSO algorithm from the available input-output data. The purpose of development of rapid charger was to reduce the charging time using high charging current, but not at the cost of damage to the batteries. Since the performance characteristics of Ni-Cd batteries were not available at high charging rates, the input-output data was obtained through rigorous experimentation [1, 2]. The parallel nature of PSO algorithm like Genetic Algorithm (GA) enhances the possibility to reach a global minimum.

This paper is organized in seven sections. Brief introduction about fuzzy model identification problem is provided in Section 2 and Section 3 carries details about PSO algorithm. Section 4 provides information about the rapid Ni-Cd battery charger. The framework for fuzzy model identification through PSO algorithm is presented in Section 5. Section 6 presents the simulation results. Conclusions are made in the final section alongwith the scope for extension of the present work.

2 Fuzzy Model Identification Problem

Generally the problem of fuzzy model identification includes the following issues [3, 4]:

- Selecting the type of fuzzy model
- Selecting the input and output variables for the model
- Identifying the structure of the fuzzy model, which includes the determination of number and types of membership functions for the input and output variables and the number of fuzzy rules
- Identifying the parameters of antecedent and consequent membership functions
- Identifying the consequent parameters of the fuzzy rulebase

Three commonly used fuzzy models are:

- Mamdani-type fuzzy models
- Takagi-Sugeno fuzzy models
- Singleton fuzzy models

In this paper, we have considered Mamdani and Singleton-type fuzzy models. Singleton-type fuzzy model is a special case of Takagi-Sugeno fuzzy model.

In Mamdani models, each fuzzy rule is of the form:
R_i: If x_1 is A_{i1} and ... and x_n is A_{in} then y is B
whereas, for Sugeno model, each fuzzy rule is of the form:
R_i: If x_1 is A_{i1} and ... and x_n is A_{in} then y is C

where, $x_1, \ldots, x_n$ are the input variables and y is the output variable, $A_{i1}, \ldots, A_{in}$, B are the linguistic values of the input and output variables in the i-th fuzzy rule and C is a constant. Some commonly used techniques for creating fuzzy models from the available input-output data are Genetic Algorithms

[5, 6, 7, 8], Fuzzy c-means (FCM) clustering algorithm [3, 9, 10], Neural Networks [3] and Adaptive Neuro Fuzzy Inference System model (ANFIS)[11, 12].

3 Particle Swarm Optimization (PSO) Algorithm

The PSO method is a member of the broad category of swarm intelligence techniques for finding optimized solutions. The motivation behind the PSO algorithm is the social behavior of animals viz. flocking of birds and fish schooling. The PSO has its origin in simulation for visualizing the synchronized choreography of bird flock by incorporating concepts such as nearest-neighbor velocity matching and acceleration by distance [13, 14, 15, 16]. Later on it was realized that the simulation could be used as an optimizer and resulted in the first simple version of PSO [15]. Since then, many variants of PSO have been suggested by different researchers [17, 18, 19]. PSO is similar to a genetic algorithm (GA) as it uses a population of potential solutions (called particles) to probe the search space and also it does not require gradient information of the objective function under consideration. One of the most promising advantages of PSO over GA is its algorithmic simplicity, as it uses only primitive mathematical operators, which accounts for low computational requirements. Unlike GA, PSO favors collaboration among the particles instead of rivalry. In PSO, the particles have an adaptable velocity that determines their movement in the search space. Each particle also has a memory and hence it is capable of remembering the best position in the search space ever visited by it. The position corresponding to the best fitness is known as *pbest* and the overall best out of all the particles in the population is called *gbest*.

Consider that the search space is d-dimensional and i-th particle in the swarm can be represented by $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and its velocity can be represented by another d-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$. Let the best previously visited position of this particle be denoted by $P_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. If g-th particle is the best particle and the iteration number is denoted by the superscripts, then the swarm is manipulated according to the following equations (1) and (2) suggested by Shi and Eberhart [14]:

$$v_{id}^{n+1} = wv_{id}^{n} + c_1 r_1^n (p_{id}^n - x_{id}^n) + c_2 r_2^n (p_{gd}^n - x_{id}^n) \tag{1}$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1} \tag{2}$$

where,

$w-$ inertia weight
c_1- cognitive acceleration parameter
c_2- social acceleration parameter
r_1, r_2- random numbers uniformly distributed in the range (0,1)

These parameters viz. inertia weight (w) cognitive parameter (c_1), social parameter (c_2), alongwith V_{max} [14] are the strategy/operating parameters of PSO algorithm and the performance of PSO to a great extent depends on the selection of these parameters. The parameter V_{max} defined by the user is the maximum velocity along any dimension, which implies that, if the velocity along any dimension exceeds V_{max}, it shall be clamped to V_{max}. The significance of these parameters can be found in [13, 14]. Generally the inertia weight is not kept fixed and is varied as the algorithm progresses so as to improve performance [13, 14].

4 Rapid Ni-Cd Battery Charger

Batteries can be classified into two main groups: Primary batteries and Secondary batteries. Unlike primary batteries, secondary batteries once discharged, can be returned to their fully charged state and can be discharged and charged many times, thus making them economical. Nickel-Cadmium (Ni-Cd), Nickel Metal Hydride (Ni-MH) and Lithium Ion (Li-Ion) are some of the commonly used secondary batteries. The most common method to charge Ni-Cd batteries is by means of the constant-current source at the rate of 0.1C (trickle charge) [20], where 0.1C is the charging rate and is commonly expressed as a multiple of the rated capacity of the battery (for a battery with C=500 mAh, 0.1C means charging current of 50 mA). At this rate, the battery is charged for 12-16 hours and can withstand overcharge without any harm. Some chargers have the capability of charging the batteries in about 5 hours using higher charging currents. But with the higher charging rates (C/3 or higher), care must be taken to avoid overcharging, as it may result into excessive rise in the temperature, which does the maximum harm to the batteries [21]. The main objective for the development of rapid battery charger was to charge the Ni-Cd batteries quickly, but without doing any damage to them. Since the behaviour of Ni-Cd batteries at very high charging rates was not available, so there was need to obtain them through experimentation. Based on the initial trials with charging rate of 8C and the fact that a Ni-Cd battery is capable of supplying current of the order of 8C [22], without any damage, the upper limit of the charging current was fixed at 8C i.e. 4A, since batteries with capacity C=500 mAh were the target batteries.

Based on the rigorous experimentation with the Ni-Cd batteries [2], it was observed that the two input variables used to control the charging rate (Ct) are absolute temperature of the batteries (T) and its temperature gradient (dT/dt). From the experiments performed, input-output data was tabulated and that data set consisting of 561 points is available at http://research.4t.com. The input and output variables identified for Ni-Cd batteries along with their universes of discourse are listed in Table 1. Universe of discourse for a variable is defined as its working range.

Table 1. Input and output variable(s) alongwith their universes of discourse

Input Variables	Universe of Discourse
Temperature (T)	$0 - 50^oC$
Temperature Gradient (dT/dt)	$0 - 1(^oC/sec)$
Output Variable	**Universe of Discourse**
Charging Rate(C_t)	$0 - 8C$

5 Framework for Fuzzy Model Identification with PSO Algorithm

The identification of the fuzzy model may be considered as an optimization or a search process. PSO algorithms like GAs have the capability to find optimal or near optimal solutions in a given complex search space and can be used to modify/learn the parameters of fuzzy model. The idea of fuzzy model identification through PSO algorithm is illustrated in Figure 1 and the framework for the identification of fuzzy model through PSO algorithm is represented as flowchart in Figure 2. Mean Square Error (MSE) defined in equation (3) is used as performance index for rating the fuzzy model evolved through PSO algorithm.

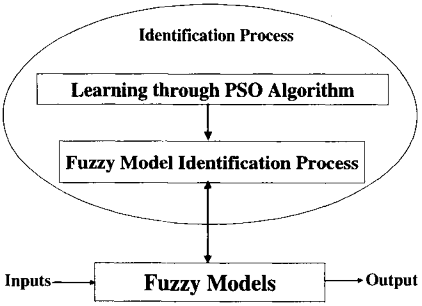

Identification Process
Learning through PSO Algorithm
Fuzzy Model Identification Process
Inputs
Fuzzy Models
Output

Fig. 1. Fuzzy Model Identification process through PSO Algorithm

$$MSE = \frac{1}{N}\sum_{k=1}^{N}[y(k) - \bar{y}(k)]^2 \qquad (3)$$

$y(k)$– desired output (as in the data)
$\bar{y}(k)$– actual output of the model
N– number of data points taken for model validation

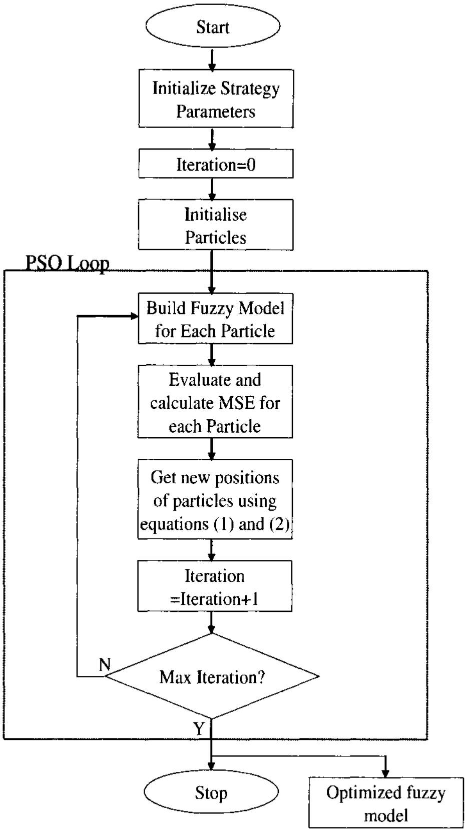

Start
Initialize Strategy Parameters
Iteration=0
Initialise Particles
PSO Loop
Build Fuzzy Model for Each Particle
Evaluate and calculate MSE for each Particle
Get new positions of particles using equations (1) and (2)
Iteration =Iteration+1
Max Iteration?
N
Y
Stop
Optimized fuzzy model

Fig. 2. Flowchart representing the framework for fuzzy model Identification process through PSO Algorithm

Following parameters of the fuzzy model were identified through PSO algorithm:

- antecedent membership functions i.e membership functions for the input variables viz. T and dT/dt
- consequent membership functions for the output variable viz. Ct for Mamdani model and singleton values for Sugeno model
- consequent parameters of the fuzzy rulebase

In this work, only triangular membership functions were used to represent the fuzzy sets. A triangular membership function can be represented by three parameters a, b and c as shown in the Figure 3.

During the entire run of PSO algorithm for fuzzy model identification, it was ensured that for all triangular membership functions $c > b > a$. At the same time the overlapping between the adjacent membership functions was also ensured. The number and types of membership functions for the input and output variables of the fuzzy model were kept fixed. For both Mamdani and

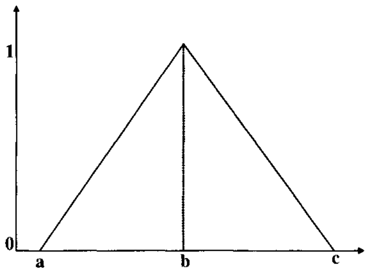


Fig. 3. Triangular membership function

Sugeno fuzzy models, three (3) number of triangular membership functions were taken for each input variable. The numbers of fuzzy sets and singleton values for the output variable for Mamdani and Sugeno models were selected to three (3) and five (5) respectively. In this paper, we considered the complete rulebase. A rulebase is said to be complete when all possible combinations of input membership functions participate in rulebase formation. For the system under consideration the number of rules for both the fuzzy models are nine (9).

Esmin A.A.A. et al. in their work [23] have used PSO algorithm for tuning membership functions only. Since in a fuzzy system, the membership functions and the rulebase are codependent, hence it is advisable to modify them simultaneously. The framework that has been used in this work is capable of doing it.

6 Simulation Results

The simulations were carried out by using PSO Fuzzy Modeler for Matlab [24], a toolbox developed by the authors. The toolbox is an open-source initiative and is hosted on SourceForge.net, which is the world's largest open-source software development website. This Matlab toolbox provides the features to generate the optimized fuzzy model (Mamdani and Sugeno) from the available data automatically using PSO algorithm.

The strategy parameters (for both Mamdani and Sugeno models) of PSO algorithm used for fuzzy model identification of Ni-Cd rapid battery charger are listed in Table 2 and the simulation results obtained for the two fuzzy models viz. Mamdani and Sugeno are presented in Table 3. Center of Gravity and Weighted Average defuzzification techniques were selected for Mamdani and Sugeno fuzzy models respectively.

The results clearly show the effectiveness of the approach as considerable improvement in the performance of fuzzy models was achieved after the complete run of PSO algorithm. More simulation time for Mamdani fuzzy models can be attributed to more complicated defuzzification process.

Table 2. PSO algorithm parameters for fuzzy model identification of Ni-Cd rapid battery charger

Swarm Size	30
Iterations	2500
c_1	2
c_2	2
w_{start}(Inertia weight at the start of algorithm)	1
w_{end}(Inertia weight at the end of algorithm)	0.1
V_{max}	100

The performance or Sugeno model identified through PSO was better than the fuzzy model identified from the same data by using ANFIS [25] model, where a MSE of 0.1321 [12] was obtained.

Table 3. Simulation Results

Model	MSE of Fuzzy System Corresponding to Swarm's gbest		Simulation Time
	After 1st Iteration	After 2500 Iterations	
Mamdani	12.46	0.1455	19.424 hours
Sugeno	46.94	0.1123	16.633 hours

7 Conclusions and Further Scope

The use of PSO algorithm for fuzzy system identification was proposed. For the presentation and validation of the approach, the data from the rapid Ni-Cd battery charger developed by the authors was used. Simulation results presented in Section 6 give a clear indication about the ability of PSO algorithm for fuzzy model identification. The proposed technique is of universal nature and there are no limitations in its usage. Future work can focus on using this method for other fields and applications.

PSO Fuzzy Modeler for Matlab, a toolbox developed by the authors was hosted on the SourceForge.net as an open-source initiative. The toolbox has the capability to generate the optimized fuzzy models from the available data and is going to help the designers build fuzzy systems in a very short time.

The suggested framework can be extended to increase the flexibility of the search by incorporating additional parameters so that the search for the optimal solution could be executed in terms of number of membership functions for each variable, the type of membership function and the number of rules.

Future work could be to investigate the influence of swarm size, number of iterations and possibly trying variants of PSO algorithm [18, 19, 26] for identifying fuzzy systems with an objective to improve their performance further.

It is a well recognized fact that the performance of evolutionary algorithms largely depends on the choice of appropriate operating/strategy parameters [27, 28]. Many users adjust the strategy parameters manually and this decision is usually taken either in terms of most common values given in the literature or by means of trial and error, which is unsystematic and requires unnecessary experimentation. Thus one of the important directions for the future work is to address the issue of parameters selection in PSO algorithm.

References

1. Arun Khosla, Shakti Kumar and K.K. Aggarwal (2002) Design and Development of RFC-10: A Fuzzy Logic Based Rapid Battery Charger for Nickel-Cadmium Batteries,HiPC2002 Workshop on Soft Computing, Bangalore, pp. 9-14.
2. Arun Khosla (1997) Design and Development of RFC-10: A Fuzzy Logic Based Rapid Battery Charger for Nickel-Cadmium Batteries. M.Tech. Thesis, Kurukshetra University, Kurukshetra. India.
3. H.Hellendoorn and D. Driankov (Eds.)(1997), Fuzzy Model Identification - Selected Approaches, Springer-Verlag.
4. John Yen and Reza Langari (2003) Fuzzy Logic - Intelligence, Control and Information, Pearson Education, First Indian Reprint.
5. A. Bastian (1996) A genetic algorithm for tuning membership functions, Fourth European Congress on Fuzzy and Intelligent Technologies EUFIT(96), Aachen, Germany, vol.1, pp. 494-498.
6. B. Carse, T.C. Fogarty and A. Munro (1996) Evolving fuzzy rule-based controllers using GA, Fuzzy Sets and Systems. 80:273-294.
7. O. Nelles (1996)FUREGA–Fuzzy Rule Extraction by GA, Fourth European Congress on Fuzzy and Intelligent Technologies EUFIT(96), Aachen, Germany, vol. 1, 1996, pp. 489-493.
8. K. Nozaki, T. Morisawa, H. Ishibuchi (1995) Adjusting membership functions in fuzzy rule-based classication systems, Third European Congress on Fuzzy and Intelligent Technologies, EUFIT(95), Aachen, Germany, vol. 1, pp. 615-619.
9. M. Setnes, J.A. Roubos (1999) Transparent fuzzy modelling using clustering and GAs, NAFIPS Conference, June 10-12, New York, USA, pp.198-202.
10. Arun Khosla, Shakti Kumar and K.K. Aggarwal (2003) Identification of Fuzzy Controller for Rapid Nickel-Cadmium Batteries Charger through Fuzzy c-means Clustering Algorithm, Proceedings of NAFIPS(2003), Chicago, July 24-26, pp. 536-539.
11. Patricia Melin and Oscar Castillo (2005) Intelligent control of a stepping motor drive using an adaptive neuro-fuzzy inference system, Information Sciences, pp. 133-151.
12. Arun Khosla, Shakti Kumar and K.K. Aggarwal (2003) Fuzzy Controller for Rapid Nickel-Cadmium Batteries Charger through Adaptive Neuro-Fuzzy Inference System (ANFIS) Architecture, Proceedings of NAFIPS(2003), Chicago, July 24-26, pp. 540-544.
13. K.E. Parsopoulos and M.N. Vrahatis (2002) Recent approaches to global optimization problems through Particle Swarm Optimization, Natural Computing, Kluwer Academic Publishers, pp. 235-306.

14. Eberhart, R.C and Shi, Y. (2001) Particle Swarm Optimization: Developments, Applications and Resources, Proceedings of the Congress on Evolutionary Computation, Seoul, Korea, pp. 81-86.
15. J. Kennedy and R. Eberhart (1995), Particle Swarm Optimization, Proceedings of IEEE Conference on Neural Networks, vol. IV, Perth, Australia, pp. 1942-1948.
16. J. Kennedy and R. Eberhart (2001), Swarm Intelligence, Morgan Kaufmann Publishers.
17. Eberhart R.C. and Kennedy J (1995) A New Optimizer Using Particle Swarm Theory, Proceedings Sixth Symposium on Micro Machine and Human Science, IEEE Service Centre, Piscataway, NJ, pp. 39-43.
18. Y. Shi and R. Eberhart (2001) Fuzzy Adaptive Particle Swarm Optimization, IEEE International Conference on Evolutionary Computation, pp. 101-106.
19. Xiao-Feng Xie, Wen-Jun Zhang, Zhi-Lian Yang (2002) Adaptive particle swarm optimization on individual level, ICSP 2002, pp 1215-1218.
20. David Linden (Editor-in-Chief) (1995), Handbook of Batteries, McGraw Hill Inc.
21. Isidor Buchmann (1997) Getting the most out of your cell phone batteries, Express Telecom, May 16-31, pp. 22-23.
22. Sunny Wan (1996) The Chemistry of Rechargeable Batteries, Cadex Electronics Inc., Canada, Revised August 26, 1996.
23. Esmin, A. A. A., Aoki, A. R., and Lambert-Torres G. (2002) Particle swarm optimization for fuzzy membership functions optimization. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 108-113.
24. http://sourceforge.net/projects/fuzzymodeler
25. Jang, J.-S.R (1993) ANFIS: Adaptive-Network-Based Fuzzy Inference System, IEEE Transactions on Systems, Man and Cybernetics, pp. 665-685.
26. K.K. Aggarwal, Shakti Kumar, Arun Khosla and Jagatpreet Singh (2003) Introducing Lifetime Parameter in Selection Based Particle Swarm Optimization for Improved Performance, First Indian International Conference on Artificial Intelligence, Hyderabad, India, pp. 1175-1181.
27. Odetayo M. O. (1997) Empirical Studies of Interdependencies of Genetic Algorithm Parameters, Proceedings of the 23rd EUROMICRO 97 Conference – New Frontiers of Information Technology, pp. 639-643.
28. David E Goldberg (2001) Genetic Algorithms in Search, Optimization and Machine Learning, Pearson Education Asia, New Delhi.

Fuzzy Association Rule Mining for Model Structure Identification

F.P. Pach[1], A. Gyenesei[2], P. Arva[1], and J. Abonyi[1]*

[1]Department of Process Engineering, University of Veszprem, Veszprem, Hungary, 8201, P.O.Box 158., www.fmt.vein.hu/softcomp, *abonyij@fmt.vein.hu

[2]Department of Knowledge and Data Analysis, Unilever Research Vlaardingen, P.O.Box 114, 3130 AC Vlaardingen, The Netherlands

Summary. Effective methods for model structure selection are very important for data–driven modelling, data mining, and system identification. A method for selecting regressors in nonlinear models with mixed discrete (categorical), fuzzy and continuous inputs and outputs is proposed based on fuzzy association rule mining. The selection of the important variables is based on the correlation measure of the fuzzy association rules.

Keywords: [1] Fuzzy association rules, model structure, process data analysis

1 Introduction

Real–world data analysis, data mining and modelling problems typically involve a large number of potential variables. The number of these variables should be minimized, especially when the model is nonlinear and contains many parameters. Therefore, effective methods for feature selection (also called structure– selection) are very important for any modelling exercise. This paper proposes a new data–driven method for the structure selection of linear and nonlinear models that can be represented by the following model: $\mathbf{y} = f(\mathbf{x})$, where $f(.)$ is a nonlinear function and $\mathbf{x}$ represents the vector of the input variables of the model. For dynamic systems, the input–selection problem includes the choice of the model's order (number of lagged inputs and outputs used as regressors) and the number of pure time delays.

A large number of structure–selection methods, like correlation or principal component analysis have been introduced for linear models. Several information–theoretical criteria have been proposed for the structure selection of linear dynamic input–output models. Examples of the classical criteria

[1] The support of the Cooperative Research Center (2004-III-1) and the Hungarian Science Foundation (T037600 and T049534) is gratefully acknowledged.

include the final prediction error and the Akaike information criterion [1]. Subsequently, Schwartz and Rissanen later developed the minimum description length criterion, which was proven to produce consistent estimates of the structure of linear dynamic models [2]. With these tools, determining the structure of linear systems is a rather straightforward task. However, these methods usually fail to discover the significant inputs in real–world data, which are almost always characterized by nonlinear dependencies. Relatively little research has been carried out on the structure selection for nonlinear models [3, 4]. In the paper of Aguirre and Billings [5], it is argued whether a certain type of term in a nonlinear model is spurious. In [6], this approach is used for the structure selection of polynomial models, and an alternative solution is introduced by initially conducting a forward search through the many possible candidate model terms before performing an exhaustive all–subset model selection on the resulting model. A backward search approach based on orthogonal parameter estimation is also applied [7, 8].

One of the most popular research tasks in data mining is the discovery of frequent item sets and association rules. The problem originates in market basket analysis which aims at understanding the behavior of retail customers, or in other words, finding associations among the items purchased together [9]. A famous example of an association rule in such a database is "diapers => beer", i.e. young fathers being sent off to the store to buy diapers, reward themselves for their trouble. Because of the practical usefulness of association rule discovery, this approach can be applied in various research areas.

This paper will present a new application of this data mining tool. A new model–free fuzzy association rule mining based method for the selection of the important variables of a data–driven model will be developed, where a correlation factor of the mined association rules is used to determine the most interesting model structures. The paper is organized as follows. Section 2 shows the base of fuzzy association rule mining, then Section 3 illustrates how the fuzzy association rule mining algorithm can be used to determine the relationships of variables in a function, select the model structure of a linear and nonlinear model, or select the most relevant features that apply to determine product quality in a production process. The general applicability and efficiently of the developed tool are showed by two examples in Section 4.

2 Fuzzy Association Rule Mining

The data available for the identification of the model is arranged into an input $\mathbf{X} = [x_{i,k}]_{n\times N}$ and an output $\mathbf{Y} = [y_{i,k}]_{m\times N}$ matrices, where N represents the number of data points, $k = 1, \ldots, N$. With the help of data–driven clustering or user–defined membership functions this data can be transformed into fuzzy data. The k-th data point is presented as

$$t_k = [a_{1,1}(x_{1,k}), a_{1,2}(x_{1,k}), \ldots, a_{1,n_1}(x_{1,k}), \ldots, a_{n,n_j}(x_{n,k}), \ldots, b_{m,n_m}(y_{m,k})]$$

and fuzzy models can be identified from such data by generating a fuzzy rule base with rules in the form of

$$R_j : If\ x_1\ is\ a_{1,j}\ and \ldots and\ x_n\ is\ a_{n,j}\ then\ y_1\ is\ B_{1,j} \tag{1}$$

In regard to our goal of generating fuzzy rule base, in this section we focus on the problem of mining fuzzy association rules. Such rules can be discovered in two steps: (1) mining frequent item sets, and (2) generating association rules from the discovered set of frequent item sets. For both steps we have to define the concept of fuzzy support, it is used as a criterion in deciding whether a fuzzy item set (association rule) is frequent or not, therefore we first introduce the basic definitions and notations that are needed in frequent item set and association rule mining.

2.1 Counting the Fuzzy Support

Let $D = \{t_1, t_2, \ldots, t_N\}$ be a transformed fuzzy data set of N tuples (data points) with a set of variables $\mathcal{Z} = \{z_1, z_2, \ldots, z_{n+m}\}$ and let $c_{i,j}$ be an arbitrary fuzzy interval (fuzzy set) associated with attribute z_i in $\mathcal{Z}$. From this point, we use the notation $\langle z_i : c_{i,j} \rangle$ for an *attribute-fuzzy interval pair*, or simply *fuzzy item*. An example could be $\langle Age : young \rangle$. For *fuzzy item sets*, we use expressions like $\langle Z : C \rangle$ to denote an ordered set $Z \subseteq \mathcal{Z}$ of attributes and a corresponding set C of some fuzzy intervals, one per attribute, i.e. $\langle Z : C \rangle = [\langle z_{i_1} : c_{i_1,j} \rangle \cup \langle z_{i_2} : c_{i_2,j} \rangle \cup \ldots \cup \langle z_{i_q} : c_{i_q,j} \rangle]$, $q \leq n + m$. In the literature, the fuzzy support value has been defined in different ways. Some researchers suggest the minimum operator as in fuzzy intersection, others prefer the product operator. They can be defined formally as follows: assuming that tuple t_k of the data set D contains value $t_k(z_i)$ for attribute z_i, then the *fuzzy support* of $\langle Z : C \rangle^2$ with respect to D is defined as

$$FS(Z : C) = \frac{\sum_{k=1}^{N} \min_{\langle z_i : c_{i,j} \rangle \in \langle Z : C \rangle} t_k(z_i)}{N} \tag{2}$$

or

$$FS(Z : C) = \frac{\sum_{k=1}^{N} \prod_{\langle z_i : c_{i,j} \rangle \in \langle Z : C \rangle} t_k(z_i)}{N} \tag{3}$$

We treat memberships as probabilities and therefore prefer the product form. A fuzzy support reflects how the record of the identification data set support the item set. An item set $\langle Z : C \rangle$ is called *frequent* if its fuzzy support value is higher than or equal to a user–defined minimum support threshold σ. The following example illustrates the calculation of the fuzzy support value. Let $\langle X : A \rangle = [\langle Balance : medium \rangle \cup \langle Income : high \rangle]$ be a fuzzy item set, the data set shown in Table 1. The fuzzy support of $\langle X : A \rangle$ is given by:

[2] The angle brackets can be omitted within parentheses.

Table 1. Example database containing membership values

$\langle Balance, medium\rangle$	$\langle Credit, high\rangle$	$\langle Income, high\rangle$
0.5	0.6	0.4
0.8	0.9	0.4
0.7	0.8	0.7
0.9	0.8	0.3
0.9	0.7	0.6

$$FS(X : A) = \frac{0.5 \times 0.4 + 0.8 \times 0.4 + 0.7 \times 0.7 + 0.9 \times 0.3 + 0.9 \times 0.6}{5} = 0.364 \tag{4}$$

2.2 Mining Frequent Item Sets

As mentioned above, the first subproblem of discovering fuzzy association rules is to find all frequent item sets. The best–known and one of the most commonly applied frequent pattern mining algorithms, *Apriori*, was developed by Agrawal [10]. The name is based on the fact that the algorithm uses prior knowledge of frequent item sets already determined. It is an iterative, breadth–first search algorithm, based on generating stepwise longer *candidate* item sets, and clever pruning of non–frequent item sets. Pruning takes advantage of the so–called *apriori* (or *upward closure*) *property* of frequent item sets: all subsets of a frequent item set must also be frequent. Each candidate generation step is followed by a counting step where the supports of candidates are checked and non-frequent ones deleted. Generation and counting alternate, until at some step all generated candidates turn out to be non–frequent. A high–level pseudo code of the algorithm is given in the following:

```
Algorithm Mining Frequent Fuzzy item sets (minimum support $\sigma$, data set $D$)
$k = 1$
$(C_k, D_F) = Transform(D)$
$F_k = Count(C_k, D_F, \sigma)$
while $|C_k| \neq 0$ do
    $inc(k)$
    $C_k = Generate(F_{k-1})$
    $C_k = Prune(C_k)$
    $F_k = Count(C_k, D_F, \sigma)$
    $F = F \cup F_k$
end
```

The subroutines are outlined as follows:

- *Transform*(D): Generates a fuzzy database D_F from the original data set D (see next section). At the same time the complete set of candidate items C_1 is found.
- *Count*(C_k, D_F, σ): In this subroutine the fuzzy database is scanned and the fuzzy support of candidates in C_k is counted. If this support is not less than minimum support σ for a given item set, we put it into the set of frequent item sets F_k.
- *Generate*(F_{k-1}): Generates candidate item sets C_k from frequent item sets F_{k-1}, discovered in the previous iteration $k-1$.
  For example, if $F_1 = \{\langle Balance : high\rangle, \langle Income : high\rangle\}$ then C_2 will be $C_2 = \{[\langle Balance : high\rangle \cup \langle Income, high\rangle]\}$.
- *Prune*(C_k): During the prune step, the item set will be pruned if one of its subsets does not exist in the set of frequent item sets F.

2.3 Generation of Fuzzy Association Rules

Since the rules are generated from the frequent item sets, the generation of fuzzy association rules becomes relatively straightforward. More precisely, each frequent item set $\langle Z : C\rangle$ is divided into a consequent $\langle Y : B\rangle$ and antecedent $\langle X : A\rangle$, where $X \subset Z, Y = Z - X, A \subset C$ and $B = C - A$. With the use of this notation a *fuzzy association rule* can be represented in the form of

$$If\ X\ is\ A\ then\ Y\ is\ B \tag{5}$$

or in more compact form of

$$\langle X : A\rangle \Rightarrow \langle Y : B\rangle \tag{6}$$

An association rule is considered *strong* if its support and confidence exceeds a given minimum support σ and minimum confidence threshold γ. Since the rules are generated from frequent item sets, they satisfy the minimum support automatically. The *confidence* of a fuzzy association rule $\langle X : A\rangle \Rightarrow \langle Y : B\rangle$ is defined as

$$FC(X : A \Rightarrow Y : B) = \frac{FS(\langle X : A\rangle \cup \langle Y : B\rangle)}{FS(X : A)} \tag{7}$$

which can be understood as the conditional probability of $\langle Y : B\rangle$, namely $P(\langle Y : B\rangle | \langle X : A\rangle)$. Using our sample database (Table 1), the fuzzy confidence value of the rule "If Balance is medium and Income is high then Credit is high" is calculated as

$$FC(X : A \Rightarrow Y : B) = \frac{0.278}{0.364} = 0.766 \tag{8}$$

Association rules mined using the above support–confidence framework are useful for many applications. However, a rule might be identified as interesting when, in fact, the occurrence $\langle X : A\rangle$ does not imply the occurrence of $\langle Y : B\rangle$.

The occurrence of an item set $\langle X : A \rangle$ is independent of the item set $\langle Y : B \rangle$ if $FS(Z : C) = FS(X : A) \cdot FS(Y : B)$; otherwise item sets $\langle X : A \rangle$ and $\langle Y : B \rangle$ are dependent and correlated as events. The correlation between the occurrence of $\langle X : A \rangle$ and $\langle Y : B \rangle$ can be measured by computing the interestingness of a given rule:

$$Fcorr(\langle X : A \rangle, \langle Y : B \rangle) = \frac{FS(Z : C)}{FS(X : A) \cdot FS(Y : B)} \tag{9}$$

If the resulting value of Equation 9 is less than 1, then the occurrence of $\langle X : A \rangle$ is negatively correlated with the occurrence of $\langle Y : B \rangle$. If the resulting value is grater than 1, then $\langle X : A \rangle$ and $\langle Y : B \rangle$ are positively correlated, meaning the occurrence of one implies the other. If the resulting value is near to 1, then $\langle X : A \rangle$ and $\langle Y : B \rangle$ are independent and there is no correlation between them.

3 MOSSFARM – Model Structure Selection by Fuzzy Association Rule Mining

This section illustrates how the previously presented fuzzy association rule mining algorithm can be used to select the most relevant features of a data–driven model. The proposed method – MOSSFARM (Model Structure Selection by Fuzzy Association Rule Mining) – consists of the following steps:

1. Generate a fuzzy database
2. Mine frequent item sets
3. Generate fuzzy association rules
4. Pruning of the rule base
5. Aggregate the rules for the selection of the input variables

Since in the previous section all of the functions needed to mine general fuzzy association rules were considered, this section will focus on the remaining steps that are needed to solve the studied feature selection problem.

3.1 Generate a Fuzzy Data Set

The original data set, D, may include crisp values (continuous and discrete) for each attribute, hence this data must be transformed into a fuzzy data set to allow fuzzy association rule mining. Therefore, the first step of the algorithm generates a new fuzzy data set from the original data set by user specified fuzzy sets. Where crisp sets are being applied instead of fuzzy ones, this step could be considered a discretization of the numeric (quantitative) attributes of the original data set. The discretization of the data into disjoint subsets (partitioning) for each variable is referred as binning, since the partitions (intervals) defined on the quantitative features can be considered as bins.

Instead of quantizing the input data into hard subsets, fuzzy Gustafson-Kessel (GK) clustering [11] is used to partition all the candidate regressors into fuzzy subsets. As a result, for each input z_j, the cluster centers $\mathbf{v}^j$ and the partition matrix $\mathbf{U}^j \in [0,1]^{n_j \times N}$ are obtained, where elements of the partition matrix represents the membership of the $z_{j,k}$ data in the ith-cluster ($k = 1, \ldots, N$ and $i = 1, \ldots, n_j$, where the number of clusters is n_j). The resulting clusters can be directly used to generate the fuzzy data, e.g. $A_{j,i}(x_{j,k}) = \mathbf{U}^j_{k,i}$. Beside this nonparametric definition of the membership values it is advantageous to design parameterized membership functions to represent the $A_{j,i}(x_{j,k})$ fuzzy sets. For this purpose, trapezoidal membership functions can be used, see in Figure 1. Each trapezoid is represented by four parameters related to the shoulders and the legs of the trapezoid: $a_{j,i}$, $b_{j,i}$, $c_{j,i}$ and $d_{j,i}$. At the current implementation, the position of the shoulders is determined based on a threshold value where $\mathbf{U}^j_{k,i} > 0.9$. The legs of the membership functions were defined to obtain a Ruspini–type partition, as $\sum_{i=1}^{n_i} A_{j,i}(x_{j,k}) = 1, \forall j, k$, i.e. $a_{j,i} = c_{j,i-1}$ and $d_{j,i} = b_{j,i+1}$.

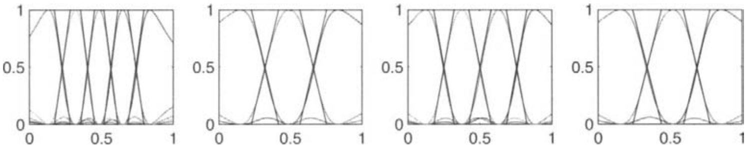


Fig. 1. Trapezoid membership functions

3.2 Pruning of the Rule Base

The advantage of the application of the correlation measure for the analysis of the quality of the rules is that it is *upward closed*. This means that if the items in a fuzzy item set $\langle Z : C \rangle$ are correlated, then every superset of $\langle Z : C \rangle$ is also correlated. In other words, adding items to a set of correlated items does not remove the existing correlation.

When searching from sets of interesting rules, the upward closure property of correlation can be used. Starting with an empty set, we may explore the item set space (or item set lattice), adding one item at a time, looking for *minimal correlated item sets* – item sets that are correlated although no subset of them is correlated. Based on this phenomena a rule based pruning algorithm has been developed that removes the unnecessarily complex rules. Such rules contain input variables that do not significantly improve the correlation of rules.

3.3 Selection of the Relevant Input Variables

In some cases, not only is the generation of interesting fuzzy (association) rules, such as *If X is A then Y is B*, important, but it is necessary to select the most important input variables (feature selection). For this purpose, it is useful to aggregate the support, the confidence, and the correlation of the individual rules. A given X set of the input variables represent a certain class of rules (and frequent item sets). Hence, it is possible to aggregate the measures of these rules $X \in R$, where R represents the set of the interesting rules:

$$FS_X = \sum_{X \in R} FS(\langle X : A \rangle \cup \langle Y : B \rangle) \tag{10}$$

$$FC_X = \sum_{X \in R} FC(X : A \Rightarrow Y : B) \frac{FS(\langle X : A \rangle \cup \langle Y : B \rangle)}{\sum_{X \in R} FS(\langle X : A \rangle \cup \langle Y : B \rangle)} \tag{11}$$

$$Fcorr_X = \sum_{X \in R} Fcorr(\langle X : A \rangle, \langle Y : B \rangle) \frac{FS(\langle X : A \rangle \cup \langle Y : B \rangle)}{\sum_{X \in R} FS(\langle X : A \rangle \cup \langle Y : B \rangle)} \tag{12}$$

With the help of the models the different sets of the input variables can be ordered, and based on this information a decision about the model–structure can be made.

4 Application Studies

4.1 Mixed Continuous and Discrete Data

This example demonstrates how the proposed algorithm can handle data sets containing both continuous and discrete inputs. Consider the following function:

$$y = \begin{cases} x_1^2 + \varepsilon \, \text{for} \; x_2 = 1 \\ x_4^2 + x_3 \varepsilon \, \text{for} \, x_2 = 0 \end{cases} \tag{13}$$

in which the output switches between two nonlinear functions depending on the value of the random discrete regressor $x_2 \in \{0, 1\}$. The remaining variables are randomly generated according to the uniform distribution $U[0, 1]$. Two additional random dummy inputs x_5 and x_6 are included in the set of candidate inputs. The noise term ε is a normally distributed random variable: $\varepsilon \sim N(0, 0.1)$. The regression–tree induction method selects the correct variables x_1 through x_4, but two dummy variables (x_5 and x_6) are also selected. With the new proposed algorithm, the number of clusters in the continuous variables is set to three, while the discrete variable clearly has two distinct values, and at the output variable the number of clusters is also three. The method selects the correct variables x_1 through x_4 with the highest correlation value, where the searching conditions $\sigma = 0.02\%, \gamma = 40\%$. The resulted several length model structures are depicted in Figure 2 and Table 2 shows the first five structures. The first structure x_1, x_2, x_3, x_4 has the highest correlation (*Fcorr* $= 263$), and it does not include any dummy variables, therefore these true model inputs are used in further modelling or analysis.

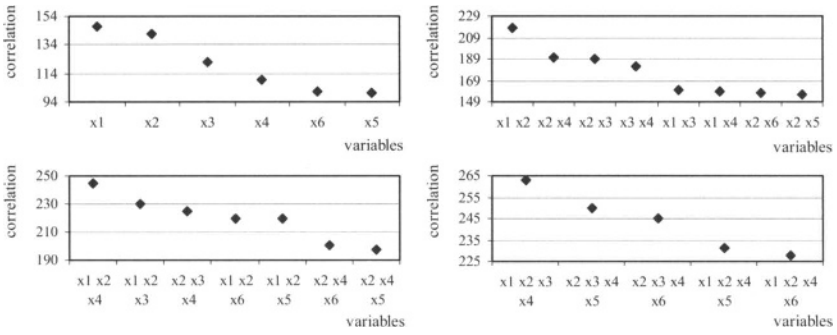


Fig. 2. Selected Model Structures for the Example Function

Table 2. Structures with Highest Correlation

structure #	selected variables	FS	FC	Fcorr
1	x_1, x_2, x_3, x_4	0.58	84	263
2	x_2, x_3, x_4, x_5	0.57	77	250
3	x_2, x_3, x_4, x_6	0.60	77	245
4	x_1, x_2, x_4, x_5	0.69	82	231
5	x_1, x_2, x_4, x_6	0.69	81	228

4.2 Continuous Polymerization Reactor

In the second example, originating from [12], the aim is to generate the model order of a dynamical system based on the data generated by a simulation model of a continuous polymerization reactor. The regression–tree induction method and the proposed algorithm were applied to all of the 941 data points and selected all of the eight variables x_1 through x_8, i.e $y_{k+1} = f(y_k, y_{k-1}, \dots, y_{k-4}, u_k, \dots, u_{k-4})$. The MOSSFARM selected only 2-3 length model structures. The first five structures are depicted in Table 3. As can be seen, the model was able to select the correct model structure $y_{k+1} = f(y_k, u_k, u_{k-1})$ [13] where the number of the clusters are set to three for all input variables, and five for the output one ($\sigma = 1\%$ and $\gamma = 75\%$).

5 Conclusions

A method for selecting regressors in nonlinear models with mixed discrete (categorical), fuzzy and continuous inputs and outputs is proposed based on fuzzy association rule mining. The proposed method performs better than regression trees which in many cases overfit the data by selecting a larger

Table 3. Summary of The Results for Polymerization Reactor

structure #	selected variables	FS	FC	Fcorr
1	x_1, x_5, x_6	0.14	89	612
2	x_5, x_6, x_7,	0.13	87	603
3	x_5, x_6	0.17	82	575
4	x_1, x_5, x_8	0.16	85	574
5	x_1, x_5	0.16	85	573

set of inputs. The proposed approach has been implemented as a MATLAB program called MOSSFARM (Model Structure Selection by Fuzzy Association Rules Mining), which will be available from www.fmt.vein.hu/softcomp.

References

1. Akaike H (1974) IEEE Trans. Autom. Control 19:716–723
2. Liang G, Wilkes D, Cadzow J (1993) IEEE Trans. Signal Process 41(10):3003–3009
3. Hong T P, Kuo C S T, Chi S C (1999) Intelligent Data Analysis 3(5):363–376
4. Kuok C M, Fu A, Wong M H (1998) ACM SIGMOD Record 27(1):41–46
5. Aguirre L A, Billings S A (1995) Int. J. Control 62:569–587
6. Aguirre L A, Mendes E M A M (1996) Int. J. Bifurcation Chaos 6:279–294
7. Korenberg M, Billings S A, Liu Y, McIlroy P (1988) Int. J. Control 48:193–210
8. Mendes E M A M, Billings S A (2001) IEEE Trans. Syst. Man Cybernetics, Part A: Syst. Humans 31(6):597–608
9. Agrawal R, Imielinski T, Swami A (1993) IEEE Transactions on Knowledge and Data Engeneering, Special Issue on Learning and Discovery in Knowledge-Based Databases 5(6):914–925
10. Agrawal R, Srikant R (1994) Fast Algorithms for Mining Association Rules In: Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile 487–499
11. Gustafson D E, Kessel W C (1979) Fuzzy Clustering with Fuzzy Covariance Matrix In: Proceedings of the IEEE CDC, San Diego 761–766
12. Doyle F J, Ogunnaike B A, Pearson R K (1995) Automatica 31:697–714
13. Rhodes C, Morari M (1998) AIChE Journal 44:151-163

Modeling Public Transport Trips with General Regression Neural Networks; A Case Study for Istanbul Metropolitan Area

Hilmi Berk Celikoglu

hbcelikoglu@ins.itu.edu.tr
Technical University of Istanbul, Faculty of Civil Engineering, Division of Transportation, Maslak, 34469 Istanbul, Turkey

1. Introduction

As transportation is one of the major components of the service sector, the investments for transportation have reasonably a huge share when the budget of a government is considered. Because of the infrastructure and the superstructure concept of transportation being generally irreversible, the demand analysis for an investment in a system approach is crucial. By defining the demand for transportation as a potential for traffic flow, the importance of the trip amounts is clarified (Celikoglu and Akad 2005).

Modeling transport generally consists of four steps which are trip generation, trip distribution, modal-split and flow assignment. In the history, growth factor methods are used to estimate the aggregated zonal trip matrices. For modal-splitting stage, stochastic methods are used. The optimization methods based on many kind of algorithms are used to assign the estimated flows to routes.

In the literature, there are many transport forecast studies based on stochastic methods which had been applied during the modal-split phase. As the mode-choice concept generates diversions on trip flows, studying trip flow forecasting becomes essential in demand analysis. Studies considering the daily trip flows in the past, be it time-series models (Ahmed and Cook 1979) or neural network (NN) models (Smith and Demetsky 1994), treat traffic flow as a point process. Road traffic prediction by using a NN approach (Yasdi 1999), recursive prediction models of traffic conditions with ANNs (Zhang 2000), macroscopic modeling of freeway traffic using ANNs (Zhang et al. 2000) are encountered as applications in the past.

The ANN approach, which is a non-linear black box model, seems to be a useful alternative for modelling the complex time series. In the majority of earlier studies feed forward error back propagation method (FFBP) was employed to train the neural networks. The performance of FFBP was found superior to conventional stochastic methods in continuous daily trip flow series forecasting (Celikoglu and Akad 2005). However the FFBP algorithm has some drawbacks. They are very sensitive to the selected initial weight values and may provide performances differing from each other significantly. Another problem faced during the application of FFBP is the local minima issue. During the training stage the networks are sometimes trapped by the local error minima preventing them to reach the global minimum. The methods used in the literature to overcome local minima problem as training a number of networks starting with different initial weights, the on-line training mode to help the network to escape local minima, inclusion of the addition of random noise, employment of second order (Newton algorithm, Levenberg-Marquardt algorithm) or global methods (stochastic gradient algorithms, simulated annealing) were summarized (Maier and Dandy 2000). In the review study of the ASCE Task Committee (ASCE Task Committee 2000a) other ANN methods such as conjugate gradient algorithms, radial basis function (RBF), cascade correlation algorithm and recurrent neural networks were briefly explained. An RBF NN model is applied to overcome the drawbacks of FFBP NN model (Celikoglu and Akad 2005) and choose the adequate forecast making method depending on the PT time series data in comparison to an auto- regressive (AR) model (Celikoglu 2005).

The daily trip time series by different purposes usually varies greatly with time of the day which are often classified into peak and off-peak periods. In this study the public transport (PT) trips are reviewed in respect to a comparison of a stochastic process, an auto- regressive (AR) model, and an ANN model, general regression neural network model, used as forecasting methods.

2. General Regression Neural Network Method

General regression neural network (GRNN), Specht's term (Specht 1991) for Nadaraya-Watson kernel regression (Nadaraya 1964; Watson 1964), also reinvented in the NN literature by Schioler and Hartmann (Schioler and Hartmann 1992), does not require an iterative training procedure as in back propagation method. It approximates any arbitrary function between input and output vectors, drawing the function estimate directly from the training data. Furthermore, it is consistent; that is, as the training set size becomes large, the estimation error approaches to zero, with only mild restrictions on the function. The GRNN is used for estimation of continuous variables, as in standard regression techniques. It is related to the radial basis function network and is based on a standard statistical technique called kernel regression. By definition, the regression of a dependent variable y on an independent x estimates the most probable value for y, given x and a training set. The regression method will produce the estimated value of y which minimizes the mean squared error. GRNN is a method for estimating the joint probability density function (pdf) of x and y, given only a training set. Because the pdf is derived from the data with no preconceptions about its form, the system is perfectly general.

By following Specht (Specht 1991), f(x,y) represents the known joint continuous probability density function of a vector random variable, x, and a scalar random variable, y, the conditional mean of y given X, the regression of y on X, is given by Eq. 1.

$$E[y|X]=\frac{\int_{-\infty}^{\infty} y f(X,y)dy}{\int_{-\infty}^{\infty} f(X,y)dy} \tag{1}$$

When the density f(x,y) is not known, it must usually be estimated from a sample of observations of *x* and y. The probability estimator f'(X, Y) (shown in Eq. 2) is based upon sample values X^i and Y^i of the random variables x and y, where n is the number of sample observations and p is the dimension of the vector variable x.

$$f'(X,Y)=\frac{1}{(2\pi)^{(p+1)/2}\sigma^{(p+1)}}\frac{1}{n}\sum_{i=1}^{n}\exp\left[-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}\right]\exp\left[-\frac{(Y-Y^i)^2}{2\sigma^2}\right] \tag{2}$$

A physical interpretation of the probability estimate f'(X, Y) is that it assigns sample probability of width σ for each sample X^i and Y^i, and the probability estimate is the sum of those sample probabilities (Specht 1991). Defining the scalar function D_i^2 (shown with Eq. 3) and performing the indicated integrations yields the Eq. 4.

$$D_i^2= (X-X^i)^T(X-X^i) \tag{3}$$

$$Y'(X)=\frac{\sum_{i=1}^{n} Y^i\exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n}\exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \tag{4}$$

The resulting regression (Eq. 4) is directly applicable to problems involving numerical data. When the smoothing parameter σ is made large, the estimated density is forced to be smooth and in the limit becomes a multivariate Gaussian with covariance σ^2I. On the other hand, a smaller value of σ allows the estimated

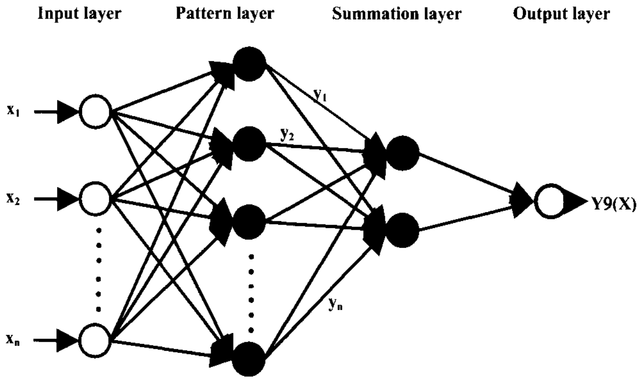


Fig. 1. Schematic diagram of a GRNN architecture

density to assume non-Gaussian shapes, but with the hazard that wild points may have too great an effect on the estimate (Specht 1991). The shematic diagram of a GRNN architecture is shown in Fig. 1.

3. Analysis of Data

In this study the averages of ½ hourly public transport trip flows using AKBIL (a ticketing system integrated to all PT modes) in Istanbul Metropolitan Area are used in order to have realistic comparisons with previous studies (Celikoglu and Akad 2005; Celikoglu 2005). The data relevant to PT vary within public bus, tram, light rail transit, metro, seabus, and city marine lines modes. The data are obtained from the observations of the first 11 month of the year 2002 starting at the January 1st and ending at the November 30th. Table 1 gives a brief description of the data set.

Table 1. Information for the analyzed data set

Type of the data	Region	x_{min}	x_{max}	Mean	Standart Deviation	Skewness	Observation period
Trip flow (Lag: ½ hour; unit: trips)	Istanbul Meropolitan Area	15	2367412	521.13	451.78	0.65	01.01.2002 to 30.11.2002

x_{min} minimum observation value, x_{max} maximum observation value.

4. ANN Preparation and Prediction Results

As the objective of the study is the forecast of the ½ hourly mean trips by using GRNNs, the analysis process consisted of two steps which are training and testing respectively.

GRNN can be treated as a normalized RBF network in which there is a hidden unit centered at every training case. These RBF units are called "kernels" and are usually probability density functions such as the Gaussian. The hidden weights to output weights are the target values, so the output is simply a weighted average of the target values of training cases close to the given input case. The only weights need to be learned are the widths of the RBF units. Optimization of these widths, the smoothing parameters, and the minimization of the mean squared error on GRNN generated values by means of the estimated value of y produced by regression method are the aim of the training process. A code in C language is written for GRNN simulations.

Following the training period the network is applied to the testing data. The regression had after the integrations on Eq. 1 and substitutions of the scalar function and smoothing parameter is directly applied to testing data. The GRNN performance criteria are the mean squared error (MSE) and the determination coefficient (R^2) for the testing stage. The predictions generated at testing period are plotted in the form of time series and scatter plot.

4.1 Preparation of GRNN Configuration

The number of input nodes to the GRNN model is dependent on the number of the time lags that is determined by the analysis of the auto-correlation function (ACF) (Sudheer et al. 2002) belonging to analyzed time series data set. Four time lags were found sufficient in the previous studies with the same data set, hence; the input layer consisted of four nodes (Celikoglu and Akad 2005; Celikoglu 2005).

The determination of the smoothing parameter and the number of the nodes in the input layer providing best training results was the initial process of the training procedure. MSE and the ACF are used as criteria to evaluate the performance of the training simulations.

Because the second step is largely a trial-and-error process, experiments involving GRNNs with input nodes more than four didn't show any sizeable improvement in prediction accuracy. Moreover, like kernel methods in general, GRNN suffers from the curse of dimensionality and can not ignore irrelevant inputs without major modifications to the basic algorithm (Wand and Jones 1995). So, GRNNs failed with more than five or six nonredundant inputs. The smoothing parameter (the width of the RBFs') is selected 0.5 as in previous application studies with the same data set (Celikoglu and Akad 2005; Celikoglu 2005).

4.2 GRNN Prediction Results

The last 348 ½ hourly mean PT trip flow values (the interval starts at the April 1st and ends at the November 30th in the year 2002) are analyzed during the training stage of GRNN. The first 180 values (the interval starts at the January 1st and ends at the March 30th in the year 2002) are then used for the testing phase. As the estimate is bounded by the minimum and the maximum of the observations the last part of the time series is used as training data.

The MSE value for the testing period of GRNN is found 0.000326. The MSEs and the determination coefficients for different orders of time series depending on the optimum hidden unit number for each order is shown in Table 3. It is clear that the best performance criteria are obtained for a GRNN configuration with four input nodes (Table 2). The testing period predictions provided by GRNN are compared with observations in Figure 2. The time series and scatter plots showed a quite satisfactory agreement between GRNN and observation values.

Table 2. The MSE and the determination coefficient values of GRNN predictions (testing period)

Input layer node number	MSE	R^2
1	0.002032	0.916
2	0.000844	0.988
3	0.000403	0.990
4	0.000326	0.993
5	0.000967	0.982

R^2 determination coefficient.

Following the GRNN forecasts a comparison is made by means of a stochastic model. An AR model of order four was used. The AR process $\{X_t\}$ or order p has a stochastic structure, if there exist constants δ, ϕ_1, ϕ_2, ..., ϕ_p and a white noise $\{\varepsilon\}$ such that Eq. 5 is satisfied (Box and Jenkins 1976). The Akaike Information Criterion (AIC), being widely used in the area of model selection, can be a drastically biased estimate of the fitted model so, to select the order of the AR model both the AIC and the input node number of the selected GRNN configuration are evaluated.

$$y_t = \delta + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots + \varphi_p y_{t-p} + \varepsilon_t \tag{5}$$

The data period between the April 1st and the November 30th in the year 2002 (training period of GRNNs) is used to obtain the AR(4) model. Again the data period between the January 1st and the March 30th in the year 2002 (testing period of GRNNs) is used to make forecast with the AR(4) model. The results are presented in Figure 3. The AR(4) is indicated with Eq. 6.

$$y_t = 2.05121\, y_{t-1} - 1.52353\, y_{t-2} + 0.41155\, y_{t-3} + 0.04237\, y_{t-4} \tag{6}$$

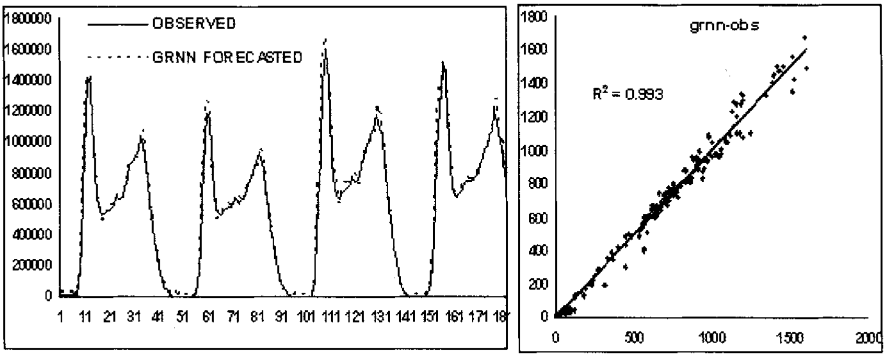


Fig. 2. The comparison of the observed ½ hourly mean PT trip flows with GRNN forecasts in testing period (Please note that values on axis in scatter plot are in terms of thousands)

The MSE obtained from the stochastic model was equal to 0.03025 which is significantly higher compared with the corresponding GRNN value (0.000326). The scatter plot illustrates clearly the deviations from observations (Figure 3).

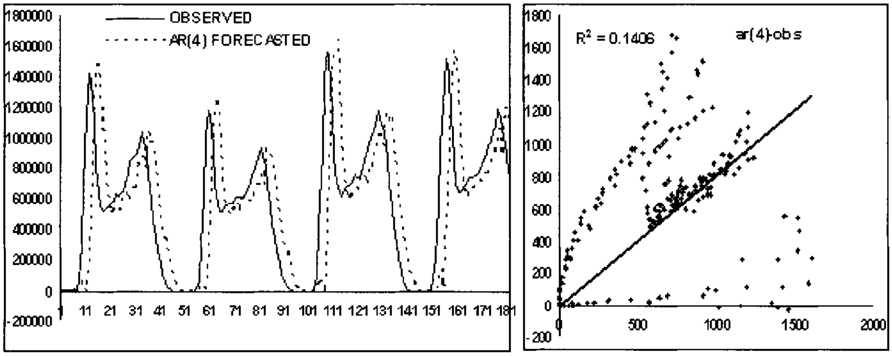


Fig. 3. The comparison of the observed ½ hourly mean PT trip flows with AR(4) model forecasts in testing period (Please note that values on axis in scatter plot are in terms of thousands)

The results point out that the GRNN forecasts for the observed values are closer to the original ones compared with the AR(4) model. To present the difference between two forecasting methods (GRNN and AR(4)) with respect to the result of a former FFBP NN study (Celikoglu and Akad 2005) clearly, the testing period forecasts are plotted with the corresponding observations in Figure 4. Both in the

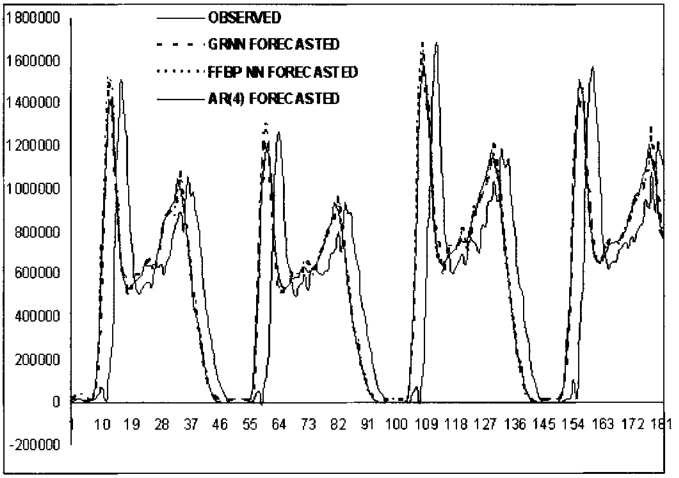


Fig. 4. The comparison of the observed ½ hourly mean PT trip flows with GRNN and AR(4) model forecasts in testing period

peak and off-peak hour periods, the GRNNs provided pretty close estimates whereas the AR(4) model failed by over and under estimating the observations.

5. Conclusions

The superiority of GRNNs over a conventional method in the reviewed prediction study can be attributed to the capability of ANNs to capture the nonlinear dynamics and generalize the structure of the whole data set. ANNs' simulations are easy to code and implement for constructing intricate multipurpose nonlinear solutions. The method has no limitations in the form of fixed assumptions or formal constraints. The NN has a distributed processing structure. Each individual processing unit or the weighted connection between two units is responsible for one small part of the input–output mapping system. Therefore, each component has no more than a marginal influence with respect to the complete solution and that the mechanism will still function and generate reasonable mappings.

Modeling PT trips and having accurate predictions helps to determine the future demand for modal choice which leads the concentration of the forthcoming investments for an urban area. In this study a GRNN trip flow estimating model is developed and compared with a stochastic model with computing the field data for Istanbul Metropolitan Area. Among the studied models, significant improvements in the prediction are made by the NN model due to its flexibility to adapt to time-series data. The provided estimates of the continous variables and convergence to underlying regression surface fit the concept of time-series. GRNNs performed faster learning by not requiring an iterative procedure in contrast to FFBP method (Celikoglu and Akad 2005). Besides, the proposed GRNN did not face the local minima problem providing prediction results with a single iteration. Making forecasts by generating negative values after trials in FFBP algorithm trained NNs (Celikoglu and Akad 2005) is not encountered in GRNN simulations. The GRNNs succeeded by generating nonnegative values and closer estimates.

References

Celikoglu HB, Akad M (2005) Estimation of public transport trips by feed-forward back propagation neural networks; a case study for Istanbul. In: Hoffmann F, Köppen M, Klawonn F, Roy R (eds) Soft Computing: Methodologies and Applications. Springer-Verlag

Ahmed SA, Cook AR (1979) Analysis of freeway traffic time series data by using Box-Jenkins techniques. Transportation Research Record 722 Transportation Research Board, Washington, DC, pp 1-9

Smith BL, Demetsky MJ (1994) Short-term traffic flow rrediction: neural network approach. Transportation Research Record 1453, Transportation Research Board, Washington, DC, pp 98-104

Yasdi R (1999) Prediction of road traffic using a neural network approach. Neural Computing and Applications 8 (2): 135-142

Zhang HM (2000) Recursive prediction of traffic conditions with neural network models. ASCE Journal of Transportation Engineering 126 (6): 472-481

Zhang HM, Ritchie SG, Lo ZP (2000) Macroscopic modeling of freeway traffic using an artificial neural network. Transportation Research Record 1588, TRB, National Research Council, Washington, DC, pp 110-119

Maier HR, Dandy GC (2000) Neural network for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environmental Modeling and Software 15: 101-124

ASCE Task Committee (2000a) Artificial neural networks in Hydrology I. ASCE Journal of Hydrologic Engineering 5 (2): 115-123

Celikoglu HB (2005) Radial basis function neural network approach to estimate public transport trips in Istanbul. In: Abraham A, Dote Y, Furuhashi T, Köppen M, Ohuchi A, Ohsawa Y (eds) Soft Computing as Transdisciplinary Science and Technology, Proceedings of the fourth IEEE International Workshop WSTST'05. Springer-Verlag

Specht DF (1991) A general regression neural network. IEEE Transactions on Neural Networks 2 (6): 568-576

Nadaraya EA (1964) On estimating regression. Theory Probab. Applic. 10: 186-190

Watson GS (1964) Smooth regression analysis. Sankhya: The Indian Journal of Statistics, Series A (26): 359-372

Schioler H, Hartmann U (1992) Mapping neural network derived from the Parzen window estimator. Neural Networks, 5 (6): 903-909

Sudheer KP, Gosain AK, Ramasastri KS (2002) A data-driven algorithm for constructing artificial neural network rainfall-runoff models. Hydrological Processes 16: 1325-1330

Wand MP, Jones MC (1995) Kernel smoothing. Chapman & Hall, London

Box GEP, Jenkins GM (1976) Time series analysis, forecasting and control. Holden Day Inc., San Francisco, California

Part VII

Optimization Problems: Assignment, Partitioning and Ordering

Use of Genetic Algorithm to Optimum Test Frequency Selection

Bartlomiej Puchalski, Lukasz Zielinski, Jerzy Rutkowski

Silesian University of Technology, Institute of Electronics
ul. Akademicka 16, 44-100 Gliwice, Poland
bartlomiej.puchalski@centertel.pl, lukasz.zielinski@aster.pl,
jr@boss.iele.polsl.gliwice.pl

Summary. This paper is dealing with selection of an optimum set of test frequencies in the complex process of analog circuits testing. Genetic algorithm is used as a method for frequency selection. This approach enhances quality and speeds up finding suboptimal solution by finding more then one good result in each genetic algorithm cycle. The approach effectiveness has been verified by practical examples and compared with other approaches.

Keywords: Analog circuits, analog system testing, circuit testing, evolutionary techniques, optimization

1 Introduction

Due to the rapid development of VLSI analog circuits, analog fault diagnosis gains more attention. Following [1] testing techniques for analog circuits may be grouped into two main categories: Simulation Before Test (SBT) and Simulation After Test (SAT). Concerning practical applicability, researchers have recently focused on SBT approach. Methods belonging to this category may be easily applied in case of real manufacturing processes. Especially Fault Dictionary (FD) method [2] is very promising. The main problem, determining practical applicability of this method is minimization of test nodes and maximization of a Circuit Under Test (CUT) diagnostic information. Both factors are contrary to each other. Test node selection is a NP-hard problem and may be solved with the use of heuristic methods. Recently many papers have dealt with this problem [3], [4], [5]. The best results have been obtained with the use of Genetic Algorithm (GA) [8]. This approach requires high computational effort, however simulations are performed before testing process. In order to further reduce the number of test nodes and increase diagnostic information, a new method utilizing many input stimuli has been proposed in [9]. As an input stimuli, AC sine wave with different frequencies has been chosen. Such signals may be easily generated, what saves cost of Automatic Test Equipment (ATE). In [9] test frequencies have been selected based on design-engineer's experience. However, a robust method of optimum frequency selection is required. In this paper, a new method utilizing GA [6], [7] is proposed. This method allows you to find the minimum test point set that maximizes the faults identification in analog electronic circuits. Method is compared with other alternative heuristic techniques [10].

In Section 2, a detailed description of the proposed method is given. In Section 3, a verification of this method on practical circuits is presented and the obtained results are compared with those obtained by other methods.

2 Method Description

The proposed method is a modification of the method presented in [9]. A new term - test point is introduced. Test point is described by two parameters:

- test node, where measurement has to be taken;
- test excitation, at which measurement has to be taken.

Therefore total number of all possible test points P_A is given by equation (1):

$$P_A = E \cdot W \tag{1}$$

E is the number of all possible test excitations, and W is the number of all accessible test nodes. In the proposed method, measurements are taken only at the output of the examined circuit. In this case total number of test points P_A is equal to the number of available excitations (1). It is assumed that F faults have been selected by design-engineer. For each test point, measurements at different circuit conditions (healthy + F faults) are grouped in ambiguity sets [2] and stored in the so called Integer-Code FD [3]. A non-optimum fault dictionary is given in Table 1. Then, such dictionary is optimized with the use of GA [8], [9]. Each test point p^i corresponds to particular input test frequency. The obtained optimum set of test points, P_O corresponds to the optimum set of excitations. This set contains minimum number of test points, providing maximum information about a CUT and is an optimum solution,

Table 1. Fault dictionary for E input stimuli and single test node (output)

Fault			Test Point		
	p^1	p^2	…	p^{E-1}	p^E
f_0 (NOM)			SIGNATURE		
f_1			SIGNATURE		
f_2			SIGNATURE		
⋮			⋮		
f_F			SIGNATURE		

when internal nodes are inaccessible. In case some internal nodes are accessible and number of test nodes is more than one, these excitations may be further used in optimum test point selection. Algorithm described in [9] is then implemented.

Number of genes in the chromosome φ correspond to all available test points (excitations) P_A. Representation of the chromosome is presented in Figure 1. If a gene is equal to 1, the corresponding excitation is included in the optimum set P_O. In order to select the best individuals, the following fitness function is proposed (2):

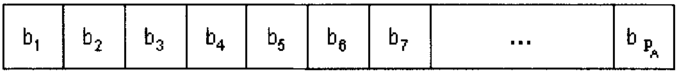


Fig. 1. Chromosome representation

$$F_C = a \cdot \frac{N_{RP}}{N_{AP}} + b \cdot \frac{N_{IF}}{N_{AF}} \tag{2}$$

where:
a, b - weighs:

$$a + b = 1 \tag{3}$$

N_{RP} - number of reduced test points, i.e. total number of check points less the considered points;
N_{AP} - number of all available test points;
N_{IF} - number of identified faults;
N_{AF} - number of all faults.

This is a normalized function consisting of two factors: test factor represented by weigh a and information factor represented by weigh b. By selecting the appropriate weighs it can be decided which part is prevailing in the function. Best individuals, providing the maximum information about a CUT with the minimum number of test points, return values close to 1. The higher value of the fitness function, the better the solution is. The algorithm stops if maximum number of generations g_{mx} is reached.

3 Computational Examples

Effectiveness of the proposed method has been verified on two practical circuits.

A. Example 1

First exemplary circuit is presented in Figure 2. For this active filter a list of $F = 38$ potential faults has been created (all resistors Short or Open). Together with healthy condition it gives 39 possible circuit states.

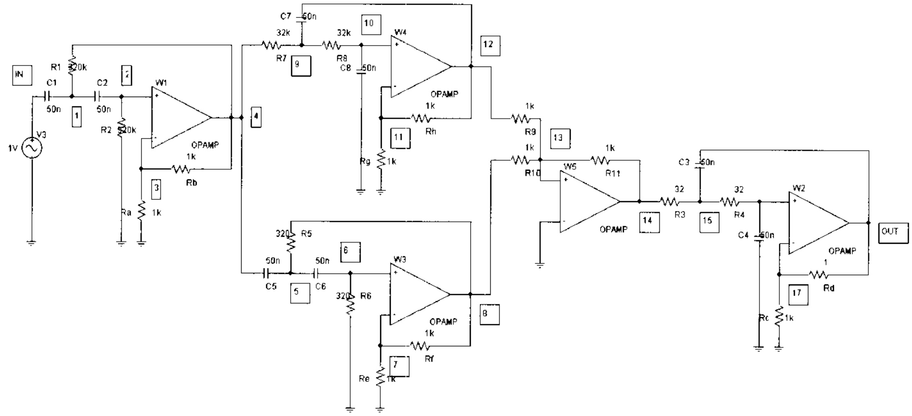


Fig. 2. Exemplary active filter circuit (example 1)

It is assumed, that 500 input test frequencies are available. It gives a set of $P_A = 500$ possible test points. Frequencies are uniformly spread on logarithmic scale, ranging from 1Hz up to 1GHz. The following parameters of the GA are chosen (for both examples):

- weighs values: $a = 0.3$, $b = 0.7$ (information factor is prevailing);
- number of individuals in population $\mu = 500$;
- maximum number of generations $g_{mx} = 300$;
- crossover probability $P_{cr} = 0.7$;
- mutation probability $P_{mu} = 0.05$;
- selection method - stochastic universal sampling.

These parameters (except weights) are typical and their selection has not been investigated. The algorithm has returned 60 alternative solutions (with the same value of fitness function) presented in Table 2. Each optimum set contains four test points, i.e. number of test frequencies has been reduced from 500 to 4! Measurements of circuit response at selected frequencies enable isolation of 16 circuit states:

$$\{\mathit{Healthy},\ R_1\mathrm{O},\ R_2\mathrm{O},\ R_A\mathrm{S},\ R_B\mathrm{O},\ R_5\mathrm{O},\ R_6\mathrm{O},\ R_7\mathrm{S},\ R_8\mathrm{S},\ R_9\mathrm{S},\ R_{10}\mathrm{S},\ R_{11}\mathrm{O},\ R_3\mathrm{S},\ R_4\mathrm{S},\ R_C\mathrm{S},\ R_D\mathrm{O}\}$$

Fitness function for each solution equals 0.5848. The obtained results have been compared with the results of alternative method utilizing information channel concept. The number of selected frequencies and the number of isolated faults are the same. The first solution obtained with the use of GA (No.

Table 2. Optimum frequency sets P_O found with the use of GA.

No.	Optimum frequency set [Hz]				No.	Optimum frequency set [Hz]			
1.	8	30	20k	45k	31.	9	195	24k	65k
2.	8	240	19k	859k	32.	14	64	6k	933k
3.	8	72	19k	859k	33.	14	59	8k	933k
4.	8	72	19k	63k	34.	14	54	6k	933k
5.	8	40	19k	859k	35.	14	42	6k	933k
6.	10	261	24k	55k	36.	14	33	6k	933k
7.	10	261	24k	49k	37.	14	31	6k	933k
8.	10	221	24k	55k	38.	13	21	6k	933k
9.	10	195	24k	55k	39.	13	54	6k	933k
10.	10	109	24k	72k	40.	13	31	6k	933k
11.	10	109	24k	55k	41.	13	31	6k	933k
12.	10	92	24k	60k	42.	11	42	6k	933k
13.	10	92	24k	55k	43.	9	212	6k	933k
14.	10	89	24k	330k	44.	9	31	6k	933k
15.	10	89	24k	55k	45.	8	204	6k	933k
16.	10	78	24k	727k	46.	8	59	6k	933k
17.	10	78	24k	292k	47.	5	54	6k	933k
18.	10	78	24k	55k	48.	3	31	6k	933k
19.	10	78	24k	45k	49.	2	31	6k	933k
20.	10	75	24k	344k	50.	2	54	6k	933k
21.	10	69	24k	55k	51.	2	31	6k	933k
22.	10	66	24k	60k	52.	1	54	8k	933k
23.	10	66	24k	55k	53.	1	33	6k	933k
24.	10	46	24k	727k	54.	1.4	31	6k	933k
25.	10	46	24k	359k	55.	1	31	8k	933k
26.	10	46	24k	317k	56.	1.3	54	6k	933k
27.	10	46	24k	60k	57.	1.3	31	18k	933k
28.	10	46	24k	55k	58.	1.3	31	6k	933k
29.	10	33	24k	344k	59.	1	54	6k	933k
30.	9	261	24k	65k	60.	1	31	6k	933k

1 in Table 2) is identical with the one obtained with the use of information channel concept. It has to be emphasized that the GA [6] method is the only method that returns many alternative solutions, which is a benefit of this approach [9], however, it requires higher computational effort.

B. Example 2

Second exemplary circuit is shown in Figure 3. It has been originally diagnosed in [10]. A list of 500 frequencies, ranging from 1Hz up to 1GHz, uniformly distributed on logarithmic scale is proposed. Under the assumption that measurements may be only taken at the circuit output, it gives P_A = 500 test points. In order to compare the proposed method with the one described in [10] the same global parametric faults are examined:

- $(R_1, \ldots, R_6)$ increased by 20%, equal 12kΩ;

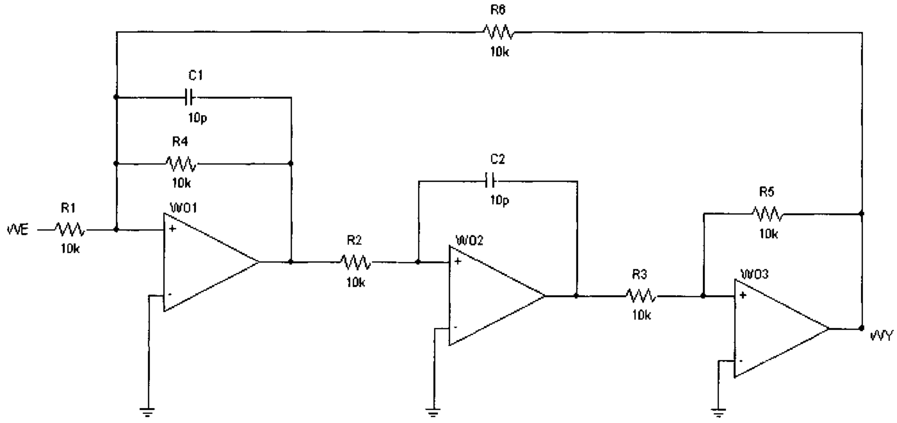


Fig. 3. Exemplary active filter circuit (example 2)

- $(R_1, \ldots, R_6)$ decreased by 20%, equal 8kΩ;
- (C_1, C_2) increased by 20%, equal 12pF;
- (C_1, C_2) decreased by 20%, equal 8pF.

After construction of non-optimum fault dictionary, the GA has been utilized for optimum test point selection. Only two solutions have been found. Each optimum set contains only one frequency!

- $p^{348} \rightarrow$ 1.81MHz;
- $p^{349} \rightarrow$ 1.89MHz;

Fitness function for both solutions equals 0.9994. Each selected frequency enables isolation of healthy circuit from two pairs of global parametric faults:

- $(R_1, \ldots, R_6) = 12\text{k}\Omega$, $(C_1, C_2) = 12\text{pF}$;
- $(R_1, \ldots, R_6) = 8\text{k}\Omega$, $(C_1, C_2) = 8\text{pF}$.

It is impossible to distinguish within the above pairs of faults. Addition of other test points (frequencies) doesn't increase an information about the CUT. Optimum test point set has been selected. Healthy condition can be separated from faulty conditions which is essential for real manufacturing process. It has to be emphasized, that only 1 out of 500 test frequencies is utilized.

The proposed method has been compared with method utilizing information theory. One of the two solutions is identical with a solution found with the use of information channel concept. Subsequently, the GA method has been compared with heuristic method of optimum frequency selection described in [10]. Very similar solution has been obtained. The same separation level of circuit states has been obtained and similar frequencies selected {2.21MHz, 1.78MHz}.

4 Conclusions

The Genetic Algorithm has been already successfully used in optimization of analog circuit testing process. Very good results in selection of an optimum set of test nodes, with single stimulus [8] or many input stimuli selected by design-engineer [9], have been obtained. Optimum stimuli selection is another area where this approach may be applied, and good results can be expected. A new method of optimum stimuli selection has been presented. This method enables selection of an optimum set of test frequencies, i.e. their minimization together with simultaneous maximization of diagnostic information about a CUT. Inaccessibility of internal nodes has been considered, i.e. measurements of output signal only. The method can be easily modified, in case some internal nodes are accessible.

The proposed GA approach has been compared with alternative heuristic approaches. The obtained results are identical or very similar, what confirms effectiveness of the proposed approach. The algorithm returns many alternative solutions, which is a benefit of the GA approach. Evolutionary optimization techniques require higher computational effort, however, calculations are performed only once, off-line, i.e. before testing of real circuits on production line. Finally, it has to be emphasized that the presented method can be modified to optimization of test signal other than sine wave, e.g. step signal. For such stimulus, set of increase times can be optimized.

References

1. Bandler J.W., Salama A., (1985), "Fault diagnosis of analog circuits", Proceedings of the IEEE, vol. 73, pp. 1279-1325.
2. Hochwald W., Bastian J.D., (1979), "A DC approach for analog fault dictionary determination", IEEE Trans. on CAS, vol. 26, pp. 523-529.
3. Lin P.M., Elcherif Y.S., (1988), "Computional Approaches to Fault Dictionary, Analog Methods for c.a. Circuit analysis and diagnosis", Edition T.Ozawa, M. Dekker, pp. 325-363.
4. Prasad V.C., Babu N.S.C, (2000), "Selection of test nodes for analog fault diagnosis in dictionary approach", IEEE Trans. on IAM, vol. 49, pp. 1289-1297.
5. Rutkowski J., (1993), "A DC approach for analog fault dictionary determination", Proceedings of European Conference on Circuit Theory and Design, pp. 877-880.
6. Baeck T., Fogel D. B., Michalewicz Z., (2000), "Evolutionary Computation 1: Basic Algorithms and Operators", Institute of Physics Publishing.
7. Goldberg D., (1989), "Genetic algorithms in search Optimization and Machine Learning", Addison-Wesley.
8. Golonek T., Rutkowski J., (2002), "Optymalizacja doboru wezlow testowych z wykorzystaniem algorytmu genetycznego", Krajowa Konferencja Elektroniki, pp. 289-294.

9. Puchalski B., Zielinski L., Rutkowski J., (2004), "Application of genetic algorithms to optimum test point selection with the use of multiple input stimuli", Proc. IEEE International Conference on Signals and Electronic Systems, pp. 389-392.
10. Rutkowski J., Puchalski B., (2003), "Optimum stimuli selection in testing of analog circuits", European Conference on Circuit Theory And Design, pp. 201-204.

Performance Analysis of Parallel Strategies for Bi-objective Network Partitioning

R. Baños[1], C. Gil[1], J. Gómez[2], and J. Ortega[3]

[1] Dept. Arquitectura de Computadores y Electrónica, Universidad de Almería, La Cañada de San Urbano s/n, 04120 Almería (Spain)
(rbanos,cgil)@ace.ual.es

[2] Dept. Lenguajes y Computación, Universidad de Almería, La Cañada de San Urbano s/n, 04120 Almería (Spain)
jgomez@ual.es

[3] Dept. Arquitectura y Tecnología de Computadores, Universidad de Granada, C/ Daniel Saucedo Aranda, E-18071, Granada (Spain)
julio@atc.ugr.es

Summary. A basic characteristic of multi-objective optimization is the conflict among the different objectives. In most real optimization problems it is very difficult, even not possible, to obtain an unique solution that optimize all the objectives. Metaheuristics methods have become important tools for solving this kind of problems. Most of them use a population of solutions, thus implying runtime increases as the population size grows. The use of parallel processing is an useful tool to overcome this drawback. This paper analyzes the performance of several parallel paradigms in the multi-objective context. More specifically, we evaluate the performance of three parallel paradigms dealing with the Pareto Simulated Annealing algorithm for Network Partitioning.

Keywords: Parallel processing; Multi-objective optimization; Pareto Simulated Annealing; Network-partitioning.

1 Introduction

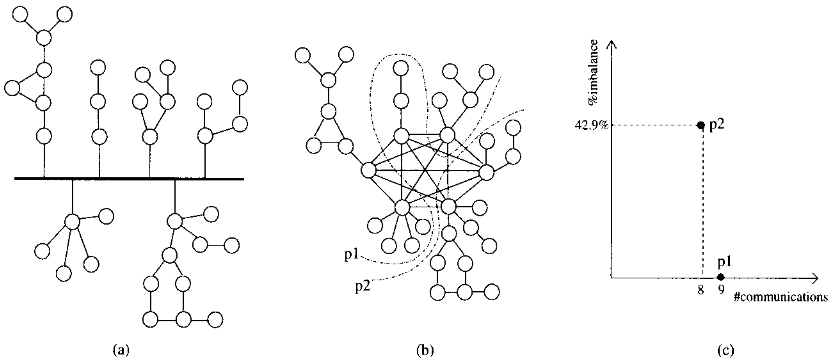
An interesting optimization problem is that of partitioning networks into several parts (sub-networks) such that the number of communication paths among nodes of different sub-networks is minimized. Since the first method for partitioning networks was proposed [6], the research in this field has increased remarkably. Examples are the partitioning of heterogeneous communication networks [25], cellular networks [22], wireless and mobile networks [3], VLSI networks [15], neural networks [16], etc.. As networks can be modeled by graphs, this problem is usually solved by using the graph partitioning methods. Until the moment, there are few studies [18, 19] dealing with multi-objective formulations of this problem.

When the complexity of the problem to solve is high, such as NP-complete problems [7], it is advisable the use of (meta)heuristic methods. Heuristics allow us to tackle large size problems instances by delivering satisfactory solutions in a reasonable runtime. Some of these multi-objective meta-heuristics (MOMHs) [4, 21, 26] use a main population of solutions during the search, and a secondary population, usually named archive that stores the best solutions found during the search process [17]. In these cases, the use of a single-processor computer implies very large runtimes, not acceptable for some real applications. One approach to overcome this weakness is the use of parallel processing. In this paper, our goal is to analyze the behavior of three parallel paradigms in Pareto Simulated Annealing algorithm.

Section 2 describes the way to represent networks by graphs, and how to partition them by using the graph partitioning model. Section 3 defines the multi-objective optimization concept, and describes the Pareto Simulated Annealing algorithm. Section 4 gives an overview of parallel and distributed strategies, and details how to extend them to the analysis of population-based MOMHs. In Section 4 the analysis of the results is provided, while Section 5 provides the conclusions of this empirical analysis.

2 Graph Partitioning as Tool to Partitioning Networks

It is known that graphs model a large variety of real problems [5]. One of these problems is that of Graph Partitioning (GPP). The translation of networks to graphs is direct: vertices in the graph represent nodes in the network, while edges correspond to connections among nodes.



(a) (b) (c)

Fig. 1. (a) LAN Network, (b) Two partitions, (c) Represent. in the solution space.

Definition 1. Graph Partitioning Problem (GPP). Given an undirected graph $G = (V, E)$, where V is the set of vertices, $|V| = n$, and E the set of edges that determines the connectivity of V. The GPP consists of dividing V into SG balanced sub-graphs, $V_1, V_2, ..., V_{SG}$, such that the number of edges that connect vertices belonging to different sub-graphs (cutsize) is minimized, verifying that $V_i \cap V_j = \phi, \forall i \neq j$; and $\cup_{sg=1}^{SG} V_{sg} = V$. The *imbalance* is defined by the maximum sub-graph weight, $M = \max(|V_{sg}|), \forall sg \in [1, SG]$.

In the single-objective formulation of the GPP, the aim is to minimize the *cutsize*, while the *imbalance* defined by M is considered a constraint. Let us suppose we desire to partition the LAN network shown in Figure 1(a) into SG = 2 sub-domains. Figure 1(b) transform the network into a graph, obtaining two partitions, which are represented in the solution spaces as Figure 1(c) shows. If the maximum *imbalance* is x = 30%, the partition *p1* is selected as the best, due to *p2* has a forbidden *imbalance*, despite having smaller *cutsize.* Otherwise, if the maximum allowed *imbalance* is, for example, x = 50%, the partition *p2* would be selected as the best. Multi-objective formulations of this problem [18] will consider equally important both objectives.

3 Sequential Algorithm: Pareto Simulated Annealing

Let P be a Multi-Objective Optimization problem (MOO), with $K \geq 2$ objectives. Instead of a scalar value, one has a vector in the multi-objective case, using Pareto-dominance relations [9]. A solution s_1 *dominates* another s_2 ($s_1 \prec s_2$) when it is better in at least one objective, and not worse in the others. A solution is said to be *non-dominated* if there is no solution that dominates it. Two solutions are considered *incomparable* if both are non-dominated (see Figure 1(c)). The set of non-dominated solutions is called Pareto optimal set. Thus, the goal of multi-objective optimization is to induce this entire set.

In the last decade, the research in the design of MOMHs has increased significantly [11]. Some of these methods [20, 23] are based on Simulated Annealing (SA) [13]. Previous studies [8] have demonstrated the good behavior of SA-based strategies in the single-objective formulation of this problem, so we try to evaluate if this behavior is also true in the multi-objective context. With this purpose, we make use of the *Pareto Simulated Annealing (PSA)* [4]. PSA is a population-based MOMH which tries to ensure an adequate dispersion of the non-dominated solutions, by dynamically varying the weights of the objective function using the Pareto-relations in a SA-based scheme. PSA also uses a secondary population (archive) which tries to guarantee the convergence [17] of the solutions.

4 Parallelization of Pareto Simulated Annealing

Parallel and distributed computing may be considered as a mechanism to speedup the search process when solving large optimization problems [2]. In addition, the simultaneous use of parallelism and cooperation allows to explore different areas of the search space, improving in many cases the quality of the solutions in comparison with the sequential version. Parallelization of MOMHs has been properly studied in the last decade [24]. Our analysis is focused to evaluate the performance of three parallelization paradigms using the procedure PSA: Master-Slave, Islands, and Islands with search space division.

4.1 Master-Slave Parallelization (MS)

The main advantage of the *Master-Slave* paradigm (MS) is that keeps the sequentiality of the algorithms. As we detailed above, one of the characteristics of PSA is that weights in the objective function are dynamically adjusted in runtime according to the dominance relations among individuals. Given this circumstance, the only parallelization strategy that preserves the sequentiality of PSA is the search for neighboring solutions. Let *np* the number of processors, the master processor creates and distributes the initial population among the other processors (each processor evaluates $|P|/np$ solutions). Afterwards, all of them are submitted to the master, and are evaluated as in the standard PSA. The same process is repeated until the stop condition is fulfilled, and the master's archive of non-dominated solutions is returned.

4.2 Island Parallelization (I)

There exist applications where the communication among processors is so expensive that the *Master-Slave* scheme becomes a very slow mechanism. A way to reduce the runtimes is to make use of the *Island* paradigm (I), where each processor runs independently PSA, using a separate sub-population. However, periodically (using a migration rate *mR*) the processors cooperate by exchanging certain individuals found in each island, in order to continue the search process in the promising areas of the search space, represented by these solutions. The adequate selection of a migration rate (mR) become another problem in this model.

4.3 Island with Search Space Division Parallelization (Issd)

A new parallelization scheme for MOEAs [14] consists of dividing the search space in a certain number of regions (e.g. *np* regions). Thus, the entire population is also divided up in *np* sub-populations, and each sub-population works in a specific region. The GPP has the characteristic that *imbalance* can be controlled in every moment, we have divided the search space in *np* regions, such that each processor works in a different range of imbalance.

5 Experimental Results

5.1 Parameter Settings

The executions have been performed in a linux cluster of 16 nodes, interconnected via Gigabit Ethernet. Each node is a dual Intel Xeon 3.06 GHz processor with 2GB of RAM memory. The parallel programs have been implemented in C standard, using the Message-Passing Interface (MPI). Table 1 describes the main characteristics of the test graphs/networks used in the evaluation of the parallel implementations. Information about the number of vertices and edges, maximum connectivity (*max*) (number of neighbors of the vertex with the highest neighborhood), minimum connectivity (*min*), and average connectivity (*avg*) is provided. All them have thousands of vertices and edges. These graphs belong to a public domain set [10] frequently used to compare single-objective graph partitioning algorithms.

Table 1. Test Networks

graph	|V|	|E|	min	max	avg
add20	2395	7462	1	123	6.23
3elt	4720	13722	3	9	5.81
uk	4824	6837	1	3	2.83
add32	4960	9462	1	31	3.82
crack	10240	30380	3	9	5.93
wing_nodal	10937	75488	5	28	13.80

The initial solutions for all the implementations are obtained by using the Graph Growing Algorithm (GGA) [12] using the random interval selection proposed in [1]. The local search process is accomplished by applying a mutation operator consisting of moving neighboring vertices (nodes) between sub-graphs (sub-networks), as it is illustrated in Figure 1(b). With the aim of evaluating the impact of the population size, we use two different configurations: |Ps|={32,128}. As PSA also removes the dominated solutions of the archive, it has not been size-limited. Determining adequate annealing parameters is often a difficult task. After several tests, we have used the following annealing parameters: initial temperature: Ti=100; cooling rate: $Tcr=0.99$; stop condition $Tstop=0.01$. These values imply a total number of iterations equal to 920. However, in order to accommodate the effect of selecting the annealing parameters, we propose to use an independent annealing scheduling in each solution of the population. In concrete, we apply the strategy proposed in [1], which obtains good results in the single-objective context. In the analysis of the speed-up, np={1,2,4,8,16} processors are used. After several experiments, an adequate migration rate is mR={100}. The results here shown, correspond to the partitioning of the test graphs into SG=16 sub-graphs.

5.2 Performance Measures

Performance of parallel implementations are measured using two metrics proposed in [26]. In what follows we describe them:

Definition 2. Coverage of two sets (*C*). Let X, X' be two subsets of solutions. The function *C* maps the pair (X,X') to the interval [0,1]. The value *C*(X,X')=1 means that all solutions in X' are dominated by solutions of X.

$$C(X,X') \leftarrow \left(\frac{|a' \in X'; \exists a \in X : a \prec a'|}{|X'|} \right) \tag{1}$$

Definition 3. Average size of the space covered (*S*). Let X=(x_1, x_2, .., x_n) be a set of solutions. The function *S*(X) returns the average volume enclosed by the union of the polytopes p_1, ..., p_n, where each p_i is formed by the intersection of the following hyperplanes arising out of x_i, along with the axes: for each axis in the objective space, there exists a hyperplane perpendicular to the axis and passing through the point ($f_1(x_i)$, ..., $f_k(x_i)$). In the bi-dimensional case, each p_i represents a rectangle defined by the points (0,0) and (f_1,f_i) (see solution *p2* in Figure 1). The work area in the GPP is defined by the maximum *imbalance* (5%), and the *cutsize* of the worst initial solution.

$$S(X) \leftarrow \sum_{i=1}^{|X|} \left(\prod_{k=1}^{|K|} \left(\frac{fk(xi)}{max(fk(xi))} \right) \right) /|\mathrm{ND}| \tag{2}$$

5.3 Empirical Results

Table 2 presents the comparison, by using metric *C*, for the best alternatives of each parallel model. It is worth noting that, as the *Master-Slave* paradigm maintains the sequentiality of PSA, non-dominated solutions obtained by *MS*

Table 2. Comparing all the methods using Metric *C* (|*Ps*|=32, np=16).

	In each graph						In average		
	MS		I		Issd		MS	I	Issd
MS	add20	3elt	0.67	0.29	0.00	0.20		0.52	0.03
	uk	add32	0.33	0.67	0.00	0.00			
	crack	wing_nodal	0.73	0.44	0.00	0.00			
I	0.00	0.17			0.00	0.00	0.16		0.00
	0.50	0.00			0.00	0.00			
	0.30	0.00			0.00	0.00			
Issd	0.67	0.33	0.67	0.14			0.39	0.38	
	0.50	0.29	0.33	0.33					
	0.20	0.33	0.27	0.56					

and PSA are similar. Quality of the non-dominated solutions in *Issd* often improve to those obtained by the other models. In average, *Issd* dominates to *MS* and *I* in 39% and 38% of cases, respectively. On the contrary, only in 3% of the non-dominated solutions of *Issd* are dominated by solutions of *MS*.

In addition to the Metric *C*, we compare these parallel models with the metric *S*. First and second numerical columns of Table 3 exhibit the absolute and normalized number of non-dominated solutions (|ND|) obtained by each parallel version. *I* obtains the largest sets, while *Issd* sets only have, in average, 2.5 solutions. Third and fourth numerical columns show the absolute and normalized enclosed area, respectively. The best results in this metric are provided by *Issd*. The great difference with respect to the other methods make us think that *Issd* obtains very good solutions in one of the objectives. With the purpose of clarifying this aspect, Figure 2(a) shows the non-dominated solutions obtained by the parallel versions in the partitioning of *add32* into *SG*=16 sub-graphs. As we can observe, even though *Issd* has only a non-dominated solution for this graph, it is not dominated by another solution obtained by other method, because it has a very low *imbalance*, which explains the good performance in metric *S*. On the contrary, this solution dominates the 29% and 33% of the non-dominated solutions obtained by *MS* and *I*, respectively (see Table 2).

Table 3. Comparing all the parallel methods using Metric *S* ($|Ps|$=32, np=16)

graph	pV	$\|ND\|$	$\|ND\|_{n.}$	S	$S_{n.}$
add20	MS	3	**1.000**	0.355	15.435
	I	3	**1.000**	0.336	14.609
	Issd	1	0.333	0.023	**1.000**
3elt	MS	6	0.857	0.077	1.750
	I	7	**1.000**	0.082	1.864
	Issd	5	0.714	0.044	**1.000**
uk	MS	8	**1.000**	0.207	17.250
	I	6	0.750	0.173	14.417
	Issd	1	0.125	0.012	**1.000**
add32	MS	7	**1.000**	0.122	122.000
	I	6	0.857	0.053	53.000
	Issd	4	0.571	0.001	**1.000**
crack	MS	10	0.909	0.161	20.165
	I	11	**1.000**	0.163	20.375
	Issd	1	0.091	0.008	**1.000**
wing_nodal	MS	6	0.666	0.194	17.630
	I	9	**1.000**	0.230	20.909
	Issd	2	0.222	0.011	**1.000**
average	MS	8.333	0.943	-	32.371
	I	8.833	**1.000**	-	20.875
	Issd	2.500	0.283	-	**1.000**

Other important aspect to analyze is the improvement in the quality of the solutions when the population size increases. Figure 2(b) shows the non-

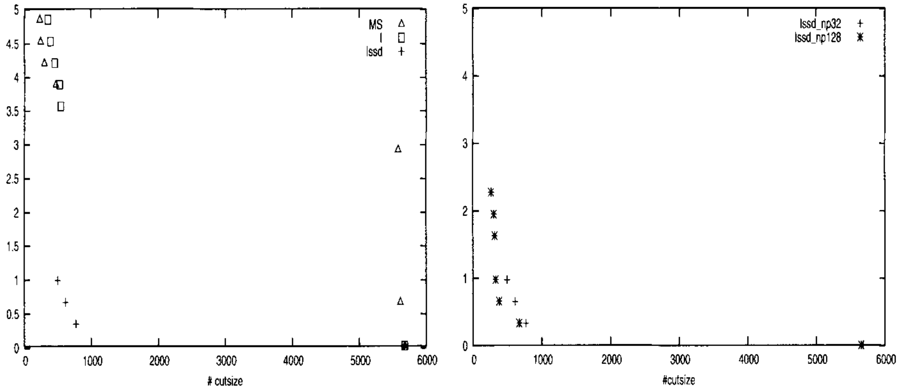


Fig. 2. Results obtained in add32 using different parallel models (a), and effect of modifying the population size (b).

dominated sets obtained by *Issd*, using |*P*|={32,128} in the partitioning of graph *add32* into *SG*=16 sub-graphs. As it is shown, non-dominated solutions obtained with |*Ps*|=128 improve to those obtained with |*Ps*|=32.

We can indicate that, although we have used a multi-objective formulation, the results obtained are close to the best known solutions for the single-objective case. Even, the *cutsize* in *MS* for graph *add20* is *cutsize*=2121, which improves the best solution found until now [10].

Last but not least important is the speed-up analysis. Figure 3 offers information in two ways. First conclusion is that the *I* paradigm obtains the best speed-up results, while in *MS* it is very poor. This difference grows up as the number of processors increases. Results for *Issd* are similar to those of *I*. The second deduction from this figure is that using large-size populations involves a lost of speed-up, due to the corresponding increase in the communication. This fact is even more apparent in the *Master-Slave* scheme because the total number of communications is much higher than in the *Island*-based implementations. Runtimes oscillate between few seconds and several minutes, according to the population size, and the test network.

6 Conclusions

In this paper we present an empirical analysis focused in the evaluation of the performance of several parallelization paradigms in an adapted version of PSA for Network/Graph Partitioning. Experimental results indicate that *Master-Slave* model maintains the sequentiality of PSA and obtains an acceptable speed-up. However, the *Island*-based schemes obtain an improvement in terms of quality of the solutions with respect to the Master-Slave implementation. On the other hand, it is worth noting that the Island model that uses search space division obtains the best results in terms of domination and space covered. These conclusions offer very useful information for parallelization of

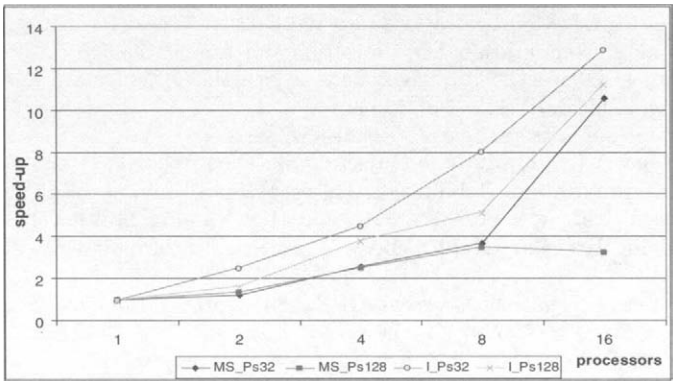


Fig. 3. Speed-up obtained by using different population sizes ($|P|=\{32, 128\}$).

other population-based MOMHs. Last but not least, solutions obtained in this problem are very close to the best ones found up to now. They even improve those obtained by single-objective methods.

Acknowledges

This work was supported by the Spanish MCyT under contracts TIC2002-00228. Authors also appreciate the support of the European Commission's Research Infrastructures activity of the Structuring the European Research Area programme, contract number RII3-CT-2003-506079 (HPC-Europa). R. Baños acknowledges a FPI doctoral fellowship from the regional government of Andalucia.

References

1. Baños R, Gil C, Ortega J, Montoya FG (2004) Parallel Multilevel Metaheuristic for Graph Partitioning. J of Heuristics 10(3): 315–336.
2. Cantu-Paz E (1998) *A Survey of Parallel Genetic Algorithms.* Calculateurs Paralleles, Reseaux et Systems Repartis 10 (2): 141–171.
3. Chen H, Zeng Q, Agrawal D (2004) A Novel Optimal Channel Partitioning Algorithm for Integrated Wireless and Mobile Networks. Wireless Networks, 10(5): 507–517.
4. Czyzak P, Jaszkiewicz A (1998) Pareto Simulated Annealing - A Metaheuristic Technique for Multiple-Objective Combinatorial Optimization, J of Multi-Criteria Decision Analysis, 7: 34–47.
5. Diestel R (1997) Graph Theory. Texts in Mathematics, Springer.
6. Fiduccia C, Mattheyses R (1982) A Linear Time Heuristic for Improving Network Partitions, in Proc. ACM Design Automation Conf., pp. 175–181.
7. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Company, San Francisco.

8. Gil C, Ortega J, Montoya MG, Baños R (2002) A Mixed Heuristic for Circuit Partitioning, Computational Optimization and Applications J., 23(3): 321–340.
9. Goldberg D (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publisher.
10. http://staffweb.cms.gre.ac.uk/˜ c.walshaw/partition/
11. Jaszkiewicz A (2001) Multiple Objective Metaheuristic Algorithms for Combinatorial Optimization. Habilitation Thesis, Poznan Univ. of Technology.
12. Karypis G, Kumar V (1998) A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM J. on Scientific Computing, 20(1): 359–392.
13. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by Simulated Annealing. Science 220(4598): 671–680.
14. Negro FT, Ortega J, Ros E, Mota S, Paechter B, Martin JM (2004) PSFGA: Parallel Processing and Evolutionary Computation for Multiobjective Optimisation. Parallel Computing 30(5-6) 721–739.
15. Park C, Park Y (1993) An Efficient Algorithm for VLSI Network Partitioning Problem using a Cost Function with Balancing Factor. IEEE Transactions on Computer - Aided Design of, Integrated Circuits and Systems 12(11): 1686–1694.
16. Ranganath HS, Kerstetter DE, Sims SRF (1995) Self Partitioning Neural Networks for Target Recognition. Neural Networks 8(9): 1475–1486.
17. Rudolph G, Agapie A (2000) Convergence Properties of some Multi-objective Evolutionary Algorithms. In: Zalzala A, Eberhart R (eds), Congress on Evolutionary Computation, IEEE Press, Piscataway, New Jersey, vol. 2, pp. 1010–1016.
18. Rummler A, Apetrei A (2002) Graph Partitioning Revised - a Multiobjective Perspective, 6th World Conf. Systemics, Cybernetics and Informatics.
19. Selvakkumaran N, Karypis G (2003) Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization, International Conf. on Computer Aided Design, IEEE Computer Society / ACM, pp. 726–733.
20. Serafini P (1993) Simulated Annealing for Multiobjective Optimization Problems. In Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application, Springer.
21. Srinivas N, Deb K (1994) Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation, 2(3): 221–248.
22. Tollis IG (1996) Optimal Partitioning of Cellular Networks, In IEEE Int. Conf. on Communications, vol. 3: pp. 1377–1381.
23. Ulungu EL, Teghem J, Fortemps P, Tuyytens D (1999) MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems, Journal of Multi-Criteria Decision Analysis, 8(4): 221–236.
24. Veldhuizen DA, Zydallys JB, Lamont GB (2003) Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation 7(2): 144–173.
25. Walshaw C, Cross M (2001) Multilevel Mesh Partitioning for Heterogeneous Communication Networks. Future Generation Comp. Syst. 17(5): 601–623.
26. Zitzler E, Thiele L (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Transactions on Evolutionary Computation, 3(4): 257–271.

A Guided Rule Reduction System for Prioritization of Failures in Fuzzy FMEA

Kai Meng Tay and Chee Peng Lim*

School of Electrical and Electronic Engineering, University of Science Malaysia
Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia
Email (*corresponding author): cplim@eng.usm.my

Abstract. Traditional Failure Mode and Effect Analysis (FMEA) utilizes the Risk Priority Number (RPN) ranking system to evaluate the risk level of failures, to rank failures, and to prioritize actions. Although this method is simple, it suffers from several shortcomings. In this paper, use of fuzzy inference techniques for RPN determination in an attempt to overcome the weaknesses associated with the traditional RPN ranking system is investigated. However, the fuzzy RPN model, suffers from the combinatorial rule explosion problem. As a result, a generic rule reduction approach, i.e. the Guided Rule Reduction System (GRRS), is proposed to reduce the number of rules that need to be provided by users during the fuzzy RPN modeling process. The proposed approach is evaluated using real-world case studies pertaining to semiconductor manufacturing. The results are analyzed, and implications of the proposed approach are discussed.

Keywords: FMEA, fuzzy logic, firing strength, Guided Rule Reduction System (GRRS)

1 Introduction

Failure Mode and Effect Analysis (FMEA) is an effective problem prevention methodology that can easily interface with many engineering and reliability methods (Ireson et al. 1995). FMEA can be described as activities intended to recognize and to evaluate the potential failures of a process and its effects. By using FMEA, a series of actions which can eliminate or reduce chance of potential failures from recurring are identified (Chrysler Corporation et. al., 1995).

From the literature review, a number of investigations have been conducted to enhance the FMEA methodology using Artificial Intelligence (AI) techniques. Fuzzy Cognitive Map was applied to FMEA to allow symbolic reasoning instead of numerical reasoning in FMEA (Peláez et al. 1996). Use of the Bayes belief network in industrial FMEA had been examined (Lee, 2001). Price, et al. (1995) and Pugh and Snooke (1996) discussed a qualitative knowledge-based system for FMEA.

Besides, fuzzy inference techniques have been applied to improve the failure risk evaluation, ranking and prioritization abilities of FMEA (Bowles et al. 1995). This approach was applied to FMEA of an auxiliary feed water system in a nuclear power plant for failure risk ranking enhancement by Guimarães et al. (2004).

It was applied to a chemical and volume control system of pressurized water reactor too (Guimarães et al., 2004a). Use of Fuzzy Reasoning and Grey Relation with FMEA for the marine industry had also been proposed to address the traditional FMEA weaknesses in failures risk evaluation (Pillay et al., 2001). Besides, a fuzzy assessment of FMEA for an engine system had been reported in Xu et al. (2002). In addition, properties of the fuzzy RPN model were investigated and enhancement for the fuzzy RPN approach was proposed by refining the weights of the fuzzy production rules (Tay et al., 2004).

FMEA with the fuzzy logic based risk ranking approaches has shown its abilities to evaluate failures according to expert's knowledge (Tay et al., 2004). However, the fuzzy approach often needs a large number of rules, and obtaining a full set of rules is a tedious, time-consuming process. In this paper, a Guided Rule Reduction System (GRRS) is proposed. The GRRS algorithm provides guidance to users on which rules are actually required and which can be eliminated. By doing so, users do not need to give all the rules but only the important and significant ones when constructing the fuzzy RPN model. Performance of the proposed approach is evaluated using case studies relating to semiconductor manufacturing.

2 Failure Risk Prioritization Issues of Traditional FMEA

In FMEA, the RPN is determined by multiplication of three input factors, i.e., Severity, Occurrence, and Detect, as follows:

$$RPN = Severity \times Occurrence \times Detect \qquad (1)$$

Severity is an assessment of the effect of potential failure mode to the next component, subsystem, or customer. Occurrence is defined as the likelihood that a specific cause will occur. Detect is an assessment of the ability of current design control to detect a potential cause (Chrysler Corporation et al. 1995). These three factors are estimated by FMEA users from "1" to "10" based on a commonly agreed evaluation criteria, thus, the scale tables. As the RPN is a measure of failures risk, it is used to rank failures and to prioritize actions. Actions will then be taken with priority given to the failure with the highest RPN.

The traditional RPN ranking has been well accepted for safety analysis in FMEA. However, it suffers from a number of weaknesses. It has been pointed out that the same RPN value can be obtained from a number of different score combinations of Severity, Occurrence, and Detect, but the risk can be different (Ben-Daya et al. 1993, Pillay et al. 2003). Besides, the relative significance of the three factors is neglected in the traditional RPN calculation. These three factors are assumed to be of equal importance, but this may not be the case in practice. Indeed, researches have shown that the relative significance of the three factors varied based on the nature of a process.

3 Modeling of the Fuzzy RPN

3.1 Fuzzy Membership Functions

Severity, Occurrence, and Detect, as in the traditional RPN function, are also used as the input factors for the fuzzy RPN function. The membership functions of these three factors are determined by interpreting the linguistic terms. Tables 1, 2, and 3 summarize some classifications/criteria describing each linguistic term. Figures 1, 2, and 3 depict the membership functions for Severity, Occurrence, Detect, respectively. Output of the fuzzy RPN model, i.e., the RPN value, is divided into five equal partitions, namely *Low, Low Medium, Medium, High Medium*, and *High*, as shown in Figure 4.

Table 1. Scale table for Severity

Severity Scale table		
Rank	Linguistic Term /Classification	Criteria
10	Very High (Liability)	Failure will affect safety or compliance to law.
9~8	High (Reliability / reputation)	Customer impact.
7~6	Moderate (Quality / Convenience)	Impacts customer yield.
5~2	Low (Internal Yield / Special Handling)	Special internal handling, effort or annoyance.
1	None (Unnoticed)	Unnoticed either internally or externally.

Table 2. Scale table for Occurrence

Occurrence scale table		
Rank	Linguistic Term/Classification	Criteria
10~9	Very High	Many/shift
8~7	High	Many/week
6~4	Moderate	Once/week
3	Low	Once/month
2	Very Low	Once/quarter
1	Remote	Once ever

Table 3. Scale table for Detect

Detect scale table		
Rank	Classification	Criteria
10	Extremely Low	No Control available.
9	Very Low	Controls probably will not Detect
8~7	Low	Controls may not Detect excursion until reach next functional area.
6~5	Moderate	Controls are able to Detect within the same functional area
4~3	High	Controls are able to Detect within the same machine/module.
2~1	Very High	Controls will Detect excursions before next lot is produced.

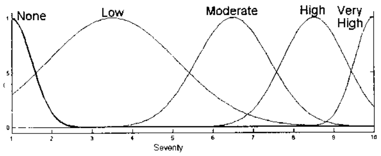


Fig. 1 Membership function for "Severity

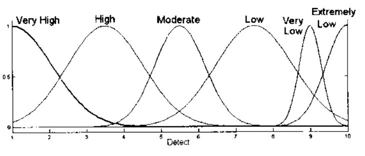


Fig. 3 Membership function for "Detect"

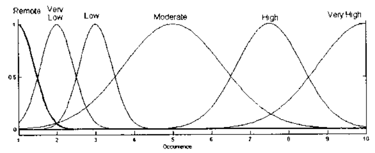


Fig. 2 Membership function for "Occurrence"

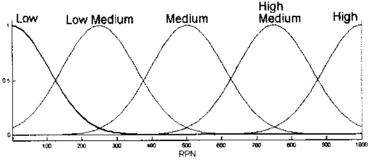


Fig. 4 Membership function for RPN

4. Rule Reduction of the Fuzzy RPN Model

Fuzzy modeling of multi-input systems is a challenging topic (Yen et al. 1999). Similar to other fuzzy models, the fuzzy RPN model suffers from the combinatorial rule explosion problem (Jin 2000). This causes the fuzzy RPN model (even a three-input fuzzy model) often has a large number of rules. It is tedious in getting a full set of rules. With a larger number of rules, the accuracy of the model prediction increases since the probability of having a "hole" over the rule base decreases. However, ease of use of the model decreases, since users need to provide a lot more information for the modeling process. Nevertheless, not all the rules are actually required for reasoning in the fuzzy RPN model, and eliminating some of the rules does not change in the model output significantly. However, some of the rules are important and cannot be ignored.

Over the years, researches on methods to reduce the rule of fuzzy systems have been conducted (Jin 2000, Yen et al. 1999, Lin 1999, and Song 2000). An important issue in rule reduction of fuzzy modeling is the approach to select a set of important fuzzy rules. Generally, these approaches are based on a firing strength matrix. An index is then employed to detect which rules should be retained or eliminated (Yen et al. 1999).

An approach to reduce the rules associated with the fuzzy RPN models, which works on a firing strength matrix, is proposed in this paper. The main advantage of the proposed approach is to formulate an efficient process for acquiring rules from domain users in building the fuzzy RPN model. With the proposed approach, the complexity of the fuzzy RPN is simplified and, at the same time, the computational load of the resulting model is reduced. Figure 5 shows the function of the proposed GRRS algorithm in the fuzzy logic-based FMEA procedure while Figure 6 demonstrates use of the GRRS in the fuzzy RPN model. The GRRS determines the need of each rule and gives guidance to FMEA users which rules are necessary and which can be ignored. Instead of supplying all the rules, users need to provide only the required rules as guided by the GRRS.

5 Algorithm of the Guided Rules Reduction System

The GRRS works on a firing strength matrix, as follows. Let I_N where $N = 1,2,3,\ldots,n$ be the possible inputs to the fuzzy RPN model with Severity (S_N), Occurrence, (O_N) and Detect. (D_N) Matrix I, can be constructed as follows.

$$\mathrm{I} = \begin{bmatrix} \mathrm{I}_1 \\ \mathrm{I}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{I}_n \end{bmatrix} = \begin{bmatrix} \mathrm{S}_1 & \mathrm{O}_1 & \mathrm{D}_1 \\ \mathrm{S}_2 & \mathrm{O}_2 & \mathrm{D}_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \mathrm{S}_n & \mathrm{O}_n & \mathrm{D}_n \end{bmatrix} \qquad (2)$$

The firing strength can be viewed as a measurement of correlation between a single input data set with a particular rule/antecedent. The firing strength is ranged from 0 to 1; the higher the firing strength value, the higher the correlation between the input data set and the particular antecedent of a rule. Let

$\mu_{S_X}(S)$ $\mu_S(S)$; where $X = 1,2,3,\ldots,n$ be the membership function of Severity;

$\mu_{O_y}(O)$; where $X = 1,2,3,\ldots,n$ be the membership function of Occurrence;

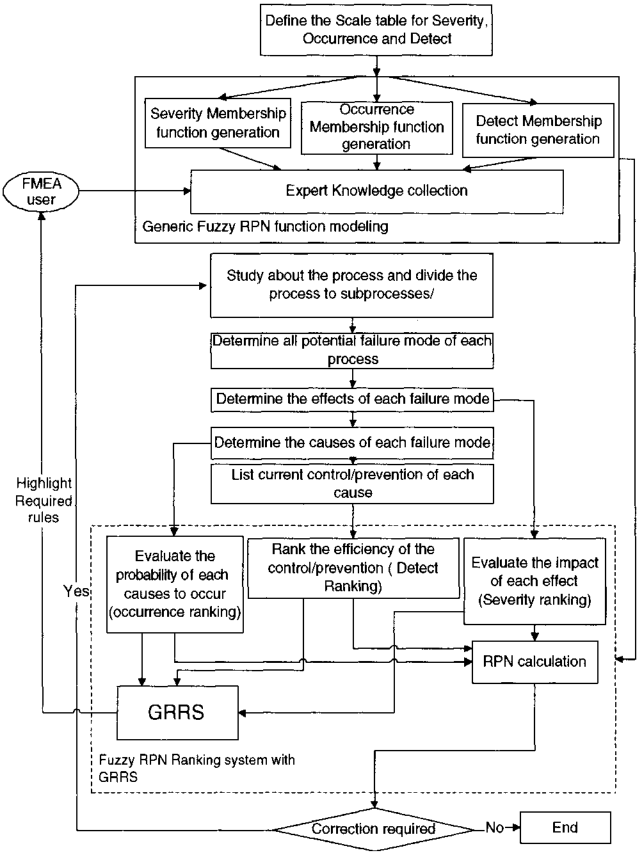

Define the Scale table for Severity, Occurrence and Detect
Severity Membership function generation
Occurrence Membership function generation
Detect Membership function generation
FMEA user
Expert Knowledge collection
Generic Fuzzy RPN function modeling
Study about the process and divide the process to subprocesses/
Determine all potential failure mode of each process
Determine the effects of each failure mode
Determine the causes of each failure mode
List current control/prevention of each cause
Highlight Required rules
Yes
Evaluate the probability of each causes to occur (occurrence ranking)
Rank the efficiency of the control/prevention (Detect Ranking)
Evaluate the impact of each effect (Severity ranking)
RPN calculation
GRRS
Fuzzy RPN Ranking system with GRRS
Correction required
No
End

Fig. 5. Fuzzy FMEA with the GRRS procedure

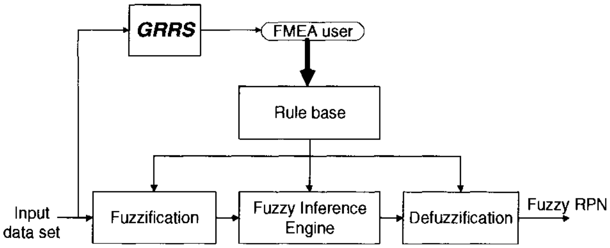


Fig. 6. The fuzzy RPN model with the GRRS

$\mu_{DZ}(D)$; where $Z = 1,2,3,\ldots,n$ be the membership function of Detect

$\mu_{RW}(R)$; where $W = 1,2,3,\ldots,n$ be the membership function of RPN;

Note that R_{XYZ} ($\mu_{SX}(S)$, $\mu_{Oy}(O)$, $\mu_{DZ}(D)$, $\mu_{RW}(R)$) is the fuzzy rule of the combination, and $\mu_{SX}(S)$, $\mu_{Oy}(O)$ and $\mu_{DZ}(D)$ using the grid partition, and the mapped output membership function is μ_{R_W} The fuzzy rule; R_{XYZ} can be transformed into a ternary fuzzy relation based on the Mamdani fuzzy implication function, i.e.,

$$R_{XYZ}(\mu_{S_X}(S),\mu_{O_Y}(O),\mu_{D_Z}(D),\mu_{R_W}(R))=(\mu_{S_X}(S)\times\mu_{O_Y}(O)\times\mu_{D_Z}(D))\times\mu_{R_W}(R) \tag{3}$$

F_{NXYZ} is the firing strength, which acts as a measurement of correlation between a single input data set I_N and the rules R_{XYZ}, where

$$F_{NXYZ}=\mu_{S_X}(S_N)\wedge\mu_{O_Y}(O_N)\wedge\mu_{D_Z}(D_N) \tag{4}$$

Thus, the correlation between the single input I_N and all the antecedents of rules can be represented as a 3-D matrix, F_N The Importance Index, S_{XYZ} is created as a measurement of importance of the antecedent of Rule R_{XYZ} to the fuzzy RPN model. Here, the S_{XYZ} is defined as;

$$S_{XYZ}=Max(F_{1XYZ},F_{2XYZ},F_{3XYZ}.....F_{nXYZ}) \tag{5}$$

Similar to F_n, S can be represented as a 3-D matrix comprising S_{XYZ}. A rule, R_{XYZ} is selected based on the importance index, S_{XYZ}, associated with the rule. Rules with a S_{XYZ} value that is lower than or equal to a prescribed threshold, λ are believed to be inactive and are deleted. This can be represented as follows.

$$\text{Rule } R_{XYZ} \text{ is selected if } S_{XYZ}\geq\lambda \text{ where } 1\geq\lambda\geq 0 \tag{6}$$

Generally, as the prescribed threshold, λ decreases, the number of rules selected increases and the model precision will increase. If λ is set to zero, all the rules will be retained.

During implementation, FMEA users only need to provide the rules as selected or pointed out by this algorithm. By this way, the algorithm gives guidance to users on which rules are required and which rules can be ignored. In this work, the percentage of rule retained is defined as;

$$\%\,of\ Rules\ retained=\frac{No.\,of\ \ rules\ selected}{Number\ of\ total\ rules\ avaiable\ using\ grid\ partition}\times 100\% \tag{7}$$

$$\%\,of\ Rules\ retained=\frac{No.\,of\ rules\ selected}{No.\,of\ MF\ (Severity)\times No.\,of\ MF\ (Occurrence)\times No.\,of\ MF\ (Detect)}\times 100\%$$

6 Case Studies and Experiments

A series of experiments to examine the performance of the proposed approach were conducted on two semiconductor manufacturing processes, i.e., the wafer mounting process, and underfill dispensing process. In each experiment, the potential failures, their causes, effects, and solutions, as well as assessment of the three input scores (Severity, Occurrence and Detect) were first identified/accessed by an expert. Risk level of each failure was then evaluated using the traditional and fuzzy RPN approaches. In addition, use of the fuzzy RPN model with the proposed GRSS (with λ=0.1, 0.2) was evaluated and compared.

6.1 Background

6.1.1 The Wafer Mounting Process

The wafer mounting process facilitates the processing of the wafer from the sawing process through die attach while keeping dies from scattering when the wafer is cut. This process consists of 5 steps: 1) frame loading; 2) wafer loading; 3) application of tape to the wafer and wafer frame; 4) cutting of the excess tape; and 5) unloading of the mounted wafer. Examples of common potential failures during this process are: wafer cracking, bubble trapping on the adhesive side of the tape, scratches on the active side of the wafer, and non-uniform tape tension which can result in tape wrinkles.

6.1.2 The Underfill Dispensing process

The main purpose of this process is to couple the chip and substrate over the entire area of the chip (Tummala, 2000). The process consists of 3 steps: 1) pre-bake-to remove residual moisture from the die and the package; 2) underfill material dispensing-to dispense liquid underfill material between the die and substrate; and 3) underfill material curing-to heat-treat the underfill material that has been dispensed between the die and substrate during underfill.

6.2 Experiment I: The Traditional and Fuzzy RPN Models

This experiment evaluates and compares the capabilities of the traditional and fuzzy RPN models in failure risk evaluation, ranking, and prioritization. Part of FMEA data from the wafer mounting and underfill dispensing processes were chosen for this experiment.

Tables 4 and 5 demonstrate the results of the traditional and fuzzy RPN (with and without the GRRS) models. Columns *"SEV"*, *"OCC"* and *"DET"* (Severity, Occurrence, and Detect, respectively) are the input sets to the RPN models. Failure risk evaluation, ranking and prioritization results based on the traditional RPN model are shown in columns *"RPN"* and "*RPN rank*". Sub columns *"FRPN"* and *"FRPN rank"* show the failure risk evaluation results using the fuzzy RPN model with and without the GRRS.

Referring to sub column *"FRPN rank*" of "*Fuzzy RPN without GRRS (λ=0)"*, the results demonstrate that failures were prioritized according to evaluation of human expert ("*Expert's Knowledge*"), who were engaged in maintenance of the processes, for the RPN determination in both the case studies. As for the results from the wafer mounting process (Table 4), the traditional approach gave Failures 16, 17, 18 and 19 (with input data sets of 3 2 4, 4 3 4, 3 2 5, and 2 2 10 for Severity, Occurrence, and Detect respectively) the RPN values of 24, 48, 30 and 40. The fuzzy RPN model (without the GRRS), however, ranked Failures 16, 17, 18 and 19 with the fuzzy RPN values of 274, 280, 430, and 460, respectively. This was in agreement with the expert's knowledge which indicated *Low Medium*, *Low Medium, Medium*, and *Medium*, for those failures.

Table 4. The failure risk evaluation, ranking and prioritization results using the RPN and fuzzy RPN (with and without the GRRS) of the wafer mounting process.

Input data sets				RPN	RPN Rank	Fuzzy RPN without GRRS (λ=0)		Fuzzy RPN (λ=0.1)		Fuzzy RPN (λ=0.2)		Expert Knowledge
						Number of Rules=180		Number of Rules=16		Number of Rules=12		
						%of rules= 100%		%of rules= 8.8%		%of rules= 6.7%		
No	SEV	OCC	DE T			FRPN	FRPN Rank	FRPN	FRPN Rank	FRPN	FRPN Rank	
1	3	1	1	3	1	99	1	99	1	99	1	Low
2	2	2	1	4	2	107	2	107	2	111	2	Low
3	3	2	1	6	3	112	3	111	3	111	2	Low
4	2	2	2	8	4	161	4	163	4	174	4	Low
5	3	1	2	6	3	168	5	168	5	168	3	Low
6	3	2	2	12	5	174	6	174	6	174	4	Low
7	2	3	2	12	5	178	7	187	7	192	5	Low Medium
8	3	3	2	18	7	192	8	192	8	192	5	Low Medium
9	2	2	3	12	5	222	9	223	9	238	6	Low Medium
10	2	3	3	18	7	224	10	229	10	241	7	Low Medium
11	2	4	2	16	6	231	11	247	14	248	9	Low Medium
12	3	2	3	18	7	238	12	238	11	238	6	Low Medium
13	3	3	3	27	9	241	13	241	12	241	7	Low Medium
14	3	4	1	12	5	246	14	246	13	246	8	Low Medium
15	3	4	2	24	8	248	15	248	15	248	9	Low Medium
16	3	2	4	24	8	274	16	272	16	272	10	Low Medium
17	4	3	4	48	12	280	17	273	17	273	11	Low Medium
18	3	2	5	30	10	430	18	424	18	424	12	Medium
19	2	2	10	40	11	460	19	460	19	503	13	Medium

Table 5. The failure risk evaluation, ranking, and prioritization results using the RPN and fuzzy RPN (with and without the GRRS) of the underfill dispensing process.

Input data sets				RPN	RPN rank	Fuzzy RPN wthout GRRS (λ=0)		Fuzzy RPN (λ=0.1)		Fuzzy RPN (λ=0.2)		Expert Know ledge
						Number of Rules=180		Number of Rules=35		Number of Rules=27		
						% of rules= 100%		% of rules=19.4%		% of rules=15.0%		
No.	SEV	OCC	DET			FRPN	FRPN Rank	FRPN	FRPN Rank	FRPN	FRPN Rank	
1	3	1	2	6	1	91	1	91	1	91	1	Low
2	3	2	1	6	1	160	2	160	3	160	3	Low
3	4	2	1	8	2	166	3	165	4	163	4	Low
4	5	1	2	10	3	201	4	147	2	145	2	Low
5	5	2	2	20	7	247	5	205	5	190	5	Low
6	5	1	5	25	8	320	6	290	6	290	6	Low Medium
7	3	3	2	18	6	357	7	357	7	357	7	Low Medium
8	4	3	1	12	4	362	8	362	8	357	7	Low Medium
9	8	1	1	8	2	461	9	461	9	460	8	Medium
10	8	1	2	16	5	529	10	532	10	531	9	Medium
11	3	3	10	90	12	545	11	558	11	558	10	Medium
12	8	2	1	16	5	627	12	628	12	628	11	High Medium
13	8	2	2	32	9	646	13	658	13	659	12	High Medium
14	4	4	1	16	5	709	14	709	14	709	13	High Medium
15	3	6	1	18	6	748	15	748	15	748	14	High Medium
16	8	3	2	48	10	758	16	760	16	823	15	High
17	8	3	3	72	11	760	17	765	17	779	16	High
18	8	6	1	48	10	849	18	859	18	908	17	High

6.3 Experiment II—The Fuzzy RPN Model with and Without the GRRS.

In this experiment, the effectiveness of the proposed GRRS-based fuzzy RPN model is examined. As can be seen from the results of the case studies, the GRSS-based model is able to reduce the rules/information to be given by domain users effectively while maintaining the ability of the fuzzy RPN model in failure risk prioritization based on expert's knowledge.

For the study on the wafer mounting process, with λ=0.1 and λ=0.2, the GRRS managed to reduce the number of fuzzy rules from 180 to 16 (with the percentage of rule retained=8.8%) and to 12(with the percentage of rule retained=6.7%). As can be observed from Table 4, outputs from the fuzzy RPN model with the GRRS are still in agreement with the expert's knowledge. However, compared with the fuzzy RPN model that had a complete rule set (180 rules), the predicted risk evaluation, the RPN values, and the risk ranking could be different. For example, the fuzzy RPN model (λ=0.1) ranked Failures 16, 17, 18 and 19 with the fuzzy RPN values of 272, 273, 424, and 460, respectively, with the expert's knowledge of *Low Medium*, *Low Medium, Medium*, and *Medium.* When λ=0.2, the failures were ranked 272, 273, 424, and 503, respectively. In comparison with the fuzzy RPN without the GRRS, the predicted results were in line with the expert's knowledge, but with different fuzzy RPN values.

The same situation can be observed from the underfill dispensing process. The GRRS reduced the number of rules to 35 and 27 (λ=0.1 and λ=0.2). Again, all the failures were prioritized in accordance with the expert's knowledge using the GRRS-based fuzzy RPN model. For example, when λ=0.1, Failures 5, 6, 7, and 8 were predicted to have fuzzy RPN values of 205, 290, 357, and 362, respectively, which are in line with the expert's knowledge (*Low*, *Low*, *Low Medium* and *Low Medium*). When λ=0.2, the same scenario was observed, with fuzzy RPN values of 190, 290, 357, and 357, respectively.

We can conclude that the GRRS-based fuzzy RPN model can effectively reduce the number of fuzzy rules needed to be obtained from domain experts. Besides, risk prioritizations predicted by the system are in agreement with the expert's knowledge.

7 Conclusions

In this paper, a fuzzy RPN model with the GRRS has been described and evaluated with real world case studies, i.e. the wafer mounting and the underfill dispensing processes in semiconductor manufacturing. Comparing with the traditional methodology, the advantages of the fuzzy approach are two-fold: (i) it allows failure risk evaluation, ranking, and prioritization to be conducted based on experts' knowledge, experiences, and opinions; (ii) it allows the failure risk evaluation function to be customized based on the nature of a process.

However, the fuzzy RPN model typically constitutes a large number of rules, and it is a tedious task to obtain a full set of rules. A generic rule reduction method for the fuzzy RPN model, i.e. the GRRS, has been proposed. A series of experiments have been conducted to evaluate the effectiveness of the proposed model using real world data. The experimental results positively demonstrate that the GRRS can provide guidance to FMEA users by pointing out the important and necessary rules in the fuzzy RPN model. Hence, the number of fuzzy rules is reduced significantly while the ability of the fuzzy RPN model in risk evaluation, ranking, and prioritization which are in agreement with expert's knowledge is maintained. Using the proposed GRRS, the time-consuming and tedious process in acquiring rules from domain users in building the fuzzy RPN model can be avoided. Besides, the GRRS simplifies the complexity and, at the same time, reduces the computational load of the resulting fuzzy RPN model.

References

Ben-Daya, M., and Raouf, A. (1993). "A revised failure mode and effects analysis model," *International Journal of Quality & Reliability Management, 3(1):*43–7.

Bowles, John B. and Peláez, C. Enrique (1995), "Fuzzy logic prioritization of failures in a system failure mode, effects and criticality analysis," *Reliability Engineering & System Safety,* Vol. 50, Issue 2, Pages 203-213

Chrysler Corporation, Ford Motor Company, and General Motors Corporation (1995), *Potential Failure Mode And Effect analysis (FMEA) Reference Manual.*

Guimarães, Antonio C. F., and Lapa, CelsoMarcelo Franklin (2004), "Effects analysis fuzzy inference system in nuclear problems using approximate reasoning," *Annals of nuclear Energy,* vol 31, pp107-115.

Guimarães, Antonio C. F., and Lapa, CelsoMarcelo Franklin (2004a), "Fuzzy FMEA applied to PWR chemical and volume control system," *Progress in Nuclear Energy, vol 44, pp191-213.*

Ireson, G., Coombs, W., Clyde, F., and Richard Y. Moss (1995). *Handbook of Reliability Engineering and Management.* McGraw-Hill Professional; 2nd edition

Jin. Y.C. (2000), "Fuzzy Modeling of high-dimensional Systems: Complexity Reduction and interpretability Improvement." *IEEE Transactions on Fuzzy Systems. Vol. 8 No 2,* pp.212-221.

Lee, B .H . (2001), "Using Bayes belief networks in industrial FMEA modeling and analysis," *Reliability and Maintainability Symposium, Proceedings.* Annual, 22-25 Jan. 2001 Page(s): 7 -15.

Lin, Sinn-Cheng (1999), "Dynamic-link rule base in fuzzy inference system." *IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings.* 1999, Volume: 5, 12-15 Oct. 1999. Pages:244 - 249 vol.5

Peláez, C. Enrique and Bowles, John B. (1996), "Using fuzzy cognitive maps as a system model for failure modes and effects analysis," *Information Sciences,* Volume 88, Issues 1-4, Pages 177-199.

Pillay, Anand and Wang, Jin (2003), "Modified failure mode and effects analysis using approximate reasoning," *Reliability Engineering & System Safety,* Volume 79, Issue 1, Pages 69-85.

Price, C. J., Pugh, D. R., Wilson, M. S., & Snooke, N. (1995). The Flame system: Automating electrical failure mode & effects analysis (FMEA). *Proceedings Annual Reliability and Maintainability Symposium,* 90-95.

Pugh, D. R., & Snooke, N. (1996). Dynamic analysis of qualitative circuits for failure mode and effects analysis. *1996 Proceedings Annual Reliability and Maintainability Symposium,* 37-42.

Song, Fuijun, and Smith, S.M. (2000), "A simple weight based fuzzy logic controller rule base reduction method." *IEEE International Conference on Systems, Man, and Cybernetics, 2000,* Volume: 5 , 8-11 Oct. 2000. Pages:3794 - 3799 vol.5.

Tummala, R.(2000) *Fundamentals of Microsystems Packaging.* New York McGraw-Hill Professional 2000.

Tay, K.M and Lim, C.P. (2004). Application of Fuzzy inference Techniques to FMEA. 9th On-Line World Conference on Soft Computing, *WSC9*, September 20 - October 08, 2004

Xu, L., Tang, L . C., Xie, M., Ho, L. H., and Zhu, M. L (2002). "Fuzzy assessment of FMEA for engine systems," *Reliability Engineering & System Safety,* Volume 75, Issue 1, 2002, Pages 17-19.

Yen, J., and Wang, L. (1999). "Simplifying fuzzy rule-based models using orthogonal transformation methods." *IEEE Transactions on Systems, Man and Cybernetics, Part B,* Volume: 29, Issue: 1, Feb. 1999 Pages: 13– 24

A Hybrid Method of Differential Evolution and SQP for Solving the Economic Dispatch Problem with Valve-Point Effect

Leandro dos Santos Coelho and Viviana Cocco Mariani

[1] Production and Systems Engineering Graduate Program, LAS/PPGEPS
leandro.coelho@pucpr.br

[2] Mechanical Engineering Graduate Program, PPGEM
viviana.mariani@pucpr.br

Pontifical Catholic University of Paraná, PUCPR/CCET
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil

Abstract. The differential evolution (DE) is an improved version of evolution strategies and Nelder-Mead simplex methods. DE has been successfully applied in various fields, such as optimization nonlinear functions, multi sensor fusion, control system, and system identification. The potentialities of DE are its simple structure, easy use, convergency speed and robustness. This paper proposes a new hybrid methodology for solving the economic load dispatch problem with valve-point effect. The proposed hybrid method combinates the DE and sequential quadratic programming (SQP) technique. The DE is used to produce good potential solutions, and the SQP is used to fine-tune the DE run. The hybrid methodology and its variants are validated for a test system consisting of 40 thermal units whose incremental fuel cost function takes into account the valve-point loading effects. The proposed hybrid method outperforms other state-of-the-art algorithms in solving load dispatch problems with the valve-point effect.

Keywords: differential evolution, economic dispatch, optimization, power generation operation and control.

1. Introduction

To supply high quality electric energy safely and economically to the consumer, electric utilities face many economic and technical problems in the operation, planning and control of electric energy systems. One of the major problems is to determine the most cost-effective and safe way of short-term generation scheduling and dispatch so that all constraints are satisfied simultaneously. In this context the economic dispatch problem (EDP) of electric energy supply is important to meet the requisites of quality and efficiency in the generation of energy.

The objective of the EDP of electric power generation, whose characteristics are complex and highly nonlinear, is to schedule the committed generating unit outputs so as to meet the required load demand at minimum operating cost while satisfying all unit and system equality and inequality constraints [9].

Improvements in scheduling the unit outputs can lead to significant cost savings. In traditional EDPs, the cost function of each generator is approximately represented by a simple quadratic function and is solved using mathematical programming based on several optimization techniques, such as dynamic programming, linear programming, homogenous linear programming, nonlinear programming and quadratic programming. However, none of these methods may be able to provide an optimal solution, for they usually get stuck at a local optimum.

Recently, as an alternative to the conventional mathematical approaches, modern heuristic optimization techniques such as simulated annealing, evolutionary algorithms (genetic algorithm, evolutionary programming, and evolution strategies), particle swarm optimization, neural networks, and taboo search have been given much attention by many researchers due to their ability to find an almost global optimal solution [4], [6], [9], [10].

In this paper, an alternative hybrid method is proposed. The proposed hybrid method combinates the differential evolution (DE) in evolution phase and the sequential quadratic programming (SQP) technique in the learning phase to solve the EDP associated with the valve-point effect. The hybrid method of optimization adopted in this paper is also denominated in the literature of the hybrid evolutionary algorithm, evolutionary algorithm with local search, memetic algorithm or optimization based in Lamarckian evolution [11].

An economic dispatch problem with 40 unit test system using nonsmooth fuel cost function [6] is employed in this paper for demonstrate the performance of the proposed hybrid method. The results obtained with the hybrid approach were analyzed and compared with those obtained in recent literature.

2. Description of Economic Dispatch Problem

The objective of the economic dispatch problem is to minimize the total fuel cost at thermal power plants subjected to the operating constraints of a power system. Therefore, it can be formulated mathematically with an objective function and two constraints. The equality and inequality constraints are represented by equations (1) and (2) given by:

$$\sum_{i=1}^{n} P_i - P_L - P_D = 0 \qquad (1)$$

$$P_i^{min} \leq P_i \leq P_i^{max} \qquad (2)$$

In the power balance criterion, an equality constraint must be satisfied, as shown in equation (1). The generated power should be the same as the total load demand plus total line losses. The generating power of each generator should lie between maximum and minimum limits represented by equation (2), where P_i is the power of generator *i* (in MW); *n* is the number of generators in the system; P_D is the system's total demand (in MW); P_L represents the total line losses (in MW) and P_i^{min} and P_i^{max} are, respectively, the output of the minimum and maximum operation of the generating unit *i* (in MW). The total fuel cost function is formulated as follows:

$$min\, f = \sum_{i=1}^{n} F_i(P_i) \tag{3}$$

where F_i is the total fuel cost for the generator unity *i* (in $/h), which is defined by equation:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{4}$$

where a_i, b_i and c_i are cost coefficients of generator *i*.

A cost function is obtained based on the ripple curve for more accurate modeling. This curve contains higher order nonlinearity and discontinuity due to the valve point effect, and should be refined by a sine function. Therefore, equation (4) can be modified [12], as:

$$\widetilde{F}_i(P_i) = F(P_i) + \left| e_i sin\left(f_i\left(P_i^{\min} - P_i\right)\right)\right| \quad \text{or} \tag{5}$$

$$\widetilde{F}_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i sin\left(f_i\left(P_i^{\min} - P_i\right)\right)\right| \tag{6}$$

where e_i and f_i are constants of the valve point effect of generators. Hence, the total fuel cost that must be minimized, according to equation (3), is modified to:

$$min\, f = \sum_{i=1}^{n} \widetilde{F}_i(P_i) \tag{7}$$

where $\widetilde{F}_i$ is the cost function of generator *i* (in $/h) defined by equation (6). In the case study presented here, we disregarded the transmission losses, P_L; thus, $P_L = 0$.

3. Optimization Methods of Economic Dispatch Problem

DE is a stochastic search algorithm that is originally motivated by the mechanisms of natural selection. DE is very effective for solving the optimization problems with nonsmooth objective functions as it not requires the derivative information. SQP, on the other hand, is a gradient-based optimization method starting from a single point and using gradient information to obtain a solution. The solution found with SQP is a locally optimal solution. In order to obtain a high quality solution, a hybrid method of DE combined with chaotic sequences and SQP is proposed here.

3.1 Differential Evolution

DE is a population-based stochastic function minimizer (or maximizer) relating to evolutionary computation, whose simple yet powerful and straightforward features make it very attractive for numerical optimization. DE uses a rather greedy and less stochastic approach to problem solving than other evolutionary algorithms. The DE algorithm was first introduced by Storn and Price in 1995 [8], and was successfully applied in the optimization of some well-known non-linear, non-differentiable and non-convex functions by Storn [7].

Two variants of DE have been reported: DE/*rand*/1/*bin* and DE/*best*/2/*bin*. The different variants are classified using the following notation: DE/α/β/δ, where α indicates the method for selecting the parent chromosome that will form the base of the mutated vector, β indicates the number of difference vectors used to perturb the base chromosome, and δ indicates the recombination mechanism used to create the offspring population. The *bin* acronym indicates that the recombination is controlled by a series of independent binomial experiments.

The fundamental idea behind DE is a scheme whereby it generates the trial parameter vectors. In each step, the DE mutates vectors by adding weighted, random vector differentials to them. If the cost of the trial vector is better than that of the target, the target vector is replaced by the trial vector in the next generation. The variant implemented here was the DE/*rand*/1/*bin*, which involved the following steps and procedures:

(i) Initialize a population of individuals (solution vectors) with random values generated according to a uniform probability distribution in the n dimensional problem space.

(ii) For each individual, evaluate its fitness value.

(iii) Mutate individuals in according to equation:

$$z_i(t+1) = x_{i,r_1}(t) + f_m\,[x_{i,r_2}(t) - x_{i,r_3}(t)] \tag{8}$$

(iv) Following the mutation operation, crossover is applied in the population. For each mutant vector, $z_i(t+1)$, an index $rnbr(i) \in \{1, 2, \cdots, n\}$ is randomly chosen using uniform distribution, and a *trial vector*, $u_i(t+1) = \left[u_{i_1}(t+1),\ u_{i_2}(t+1), \ldots, u_{i_n}(t+1)\right]^T$, is generated with

$$u_{i_j}(t+1) = \begin{cases} z_{i_j}(t+1), & \text{if } (randb(j) \leq CR) \text{ or } (j = rnbr(i)), \\ x_{i_j}(t), & \text{if } (randb(j) > CR) \text{ or } (j \neq rnbr(i)) \end{cases} \tag{9}$$

To decide whether or not the vector $u_i(t+1)$ should be a member of the population comprising the next generation, it is compared to the corresponding vector $x_i(\ t\ )$. Thus, if F_c denotes the objective function under minimization, then

$$x_i(t+1)=\begin{cases} u_i(t+1), \text{ if } F_c(t+1)<F_c(\mathrm{x_i}(t)), \\ \\ x_i(t), \text{ otherwise} \end{cases} \tag{10}$$

(iv) Loop to step (ii) until a stopping criterion is met, usually a maximum number of iterations (generations), G_{max}.

In the above equations, i =1,2,...,N is the individual's index of population; j=1,2,..., n is the position in n dimensional individual; t is the time (generation); $x_i(t)=\left[x_{i_1}(t), x_{i_2}(t),..., x_{i_n}(t)\right]^{\mathrm{T}}$ stands for the position of the i-th individual of population of N real-valued n-dimensional vectors; $z_i(t)=\left[z_{i_1}(t), z_{i_2}(t),..., z_{i_n}(t)\right]^{\mathrm{T}}$ stands for the position of the i-th individual of a *mutant vector*; r_1, r_2 and r_3 are mutually different integers and also different from the running index, i, randomly selected with uniform distribution from the set $\{1, 2, \cdots, i-1, i+1, \cdots, N\}$; f_m > 0 is a real parameter, called *mutation factor*, which controls the amplification of the difference between two individuals so as to avoid search stagnation and it is usually taken form the range [0.1, 1]; *randb(j)* is the j-th evaluation of a uniform random number generation with [0, 1]; CR is a *crossover rate* in the range [0, 1]; and F_c is the evaluation of cost function. Usually, the performance of a DE algorithm depends on three variables: the population size N, the mutation factor f_m, and the crossover rate CR.

The configurations of DE/*rand*/1/*bin* analyzed in this work are:

- DE(1): classical DE using a population size given by M = 30, a constant mutation factor of F = 0.4 and a crossover rate of CR = 0.8;
- DE(2): DE using a population size given by M = 30, a mutation factor F generated by random number with uniform distribution in the range [0.4; 1], and a crossover rate of CR = 0.8;
- DE(3): DE using a population size given by M = 30, a mutation factor F generated by random number with uniform distribution in the range [0.4; 1] and a crossover rate CR = 0.8. In this case, a cultural algorithm with normative setup [3], [5] is used for each ϕ generations in DE. In this work, the parameter ϕ was 20 generations for the DE(3) approach. The normative knowledge contains the intervals for the decision variables where good have been found, in order to move new solutions towards those intervals. The lower, $N_{lower}(G)$, and upper bounds, $N_{upper}(G)$, of solutions $P_i^{\,j}(G)$ for normative knowledge implementation are given by expressions:

$$\left[N_{lower}(G), N_{upper}(G)\right]=\left[\min P_i(G), \max P_i(G)\right] \tag{11}$$

and the equations (9) of DE are changed for:

$$\text{If } P_{i,r_1}(G) < N_{lower}(G):\ P_i(G+1) = P_{i,r_1}(G) + F\lfloor P_{i,r_2}(G) - P_{i,r_3}(G)\rfloor \qquad (12)$$

$$\text{If } P_{i,r_1}(G) > N_{upper}(G): P_i(G+1) = P_{i,r_1}(G) - F\lfloor P_{i,r_2}(G) - P_{i,r_3}(G)\rfloor \qquad (13)$$

$$\text{Otherwise: } P_i(G+1) = P_{i,r_1}(G) + F\lfloor P_{i,r_2}(G) - P_{i,r_3}(G)\rfloor \qquad (14)$$

3.2 Sequential Quadratic Programming (SQP)

The SQP used in this paper consists of three main stages, as follows: (i) calculation of an approximation of the Hessian matrix of the Lagrangian function using quasi-Newton method; (ii) formulation of the QP problem; and (iii) line search and merit function calculation. Details of the SQP procedure are presented in Fletcher [2]. The SQP method outperforms every other nonlinear programming method in terms of efficiency, accuracy, and percentage of successful solutions over a large number of test problems [1]. The method closely resembles Newton's method for constrained optimization, just as is done for unconstrained optimization. An approximation is made of the Hessian of the Lagrangian function using a BFGS (Broyden-Fletcher-Goldfarb-Shanno) quasi-Newton updating method in each iteration. The result of this approximation is then used to generate a quadratic programming (QP) sub-problem whose solution is used to form a search direction for a line search procedure.

3.3 Hybrid Approach of DE with SQP

The approaches configuration composite by deterministic techniques, this hybridized with stochastic techniques, is a promising alternative in optimization and must be evaluated. DE and SQP methods have supplementary potentialities. In this work, the following ways of hybridizing of DE combined with SQP were tested:

- DE-SQP(1): after of solve the EDP using DE, DE-SQP(1) uses the best solution from DE as a starting point and solve the EDP using SQP method;
- DE-SQP(2): after of solve the EDP using DE with normative knowledge, DE-SQP(2) uses the best solution from DE as a starting point and solve EDP using SQP;
- DE-SQP(3): similar to DE-SQP(2). However, a diversity analysis of members' population of DE is realized in each generation. In this case, the members' population with cost function inferior to 1% of cost function of best member' population are reinitialized.

4. Case Study of 40 Thermal Units

This case study involved 40 thermal units with quadratic cost functions together with the effects of valve-point loading, as shown in table 1. The data of table 1 is also available in Sinha *et al.* [6]. In this case, the load demand expected to be determined was $P_D = 10500\,\text{MW}$.

Table 1. Data for the 40 thermal units

G	P_i^{min}	P_i^{max}	a	b	c	e	f
1	36	114	0.00690	6.73	94.705	100	0.084
2	36	114	0.00690	6.73	94.705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	818	369.03	150	0.063
5	47	97	0.01140	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	278.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.90	722.82	200	0.042
11	94	375	0.00515	12.90	635.20	200	0.042
12	94	375	0.00569	12.80	654.69	200	0.042
13	125	500	0.00421	12.50	913.40	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.00010	8.95	107.87	200	0.042
35	90	200	0.00010	8.62	116.58	200	0.042
36	90	200	0.00010	8.62	116.58	200	0.042
37	25	110	0.01610	5.88	307.45	80	0.098
38	25	110	0.01610	5.88	307.45	80	0.098
39	25	110	0.01610	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

The simulation results obtained for the case study with 50 runs are presented in table 2. The SQP presented a higher cost function than all the other results, presenting a high standard deviation in the 50 runs carried out with different initial solutions. Note that the SQP was influenced by the initial value. As indicated in Table IV, the DE-SQP(1) was the approach that obtained the best fuel cost for the EDP of 40 thermal units. However, the DE-SQP(2) obtained the best mean cost among the tested techniques. The influence of normative knowledge (cultural algorithm) was decisive in attaining this performance in the DE-SQP(2) approach. DE approaches obtain results like with the SQP, nevertheless the computational time was approximately two to three time major. The best result obtained for the case study using DE-SQP(1) is shown in table 3.

Table 4 compares the results obtained in this paper with those of other studies reported in the literature. Note that in this case study, the results of the SQP and DE-SQP(1)-(3) were equal to the ones obtained with evolutionary programming, hybrid evolutionary programming with SQP, and particle swarm optimization with SQP. Nevertheless, the result of 121715.4900 \$/h reported here is comparatively lower than recent studies presented in the literature.

Table 2. Convergence results for case of 40 thermal units with valve point and $P_D = 10500$ MW using software Matlab 6.5 for Windows (PC computer with AMD 1.09 MHz processor and 128 MB of RAM)

method	mean time (sec.)	minimum cost (\$/h)	mean cost (\$/h)	standard deviation (\$/h)	maximum cost (\$/h)
SQP	**10.80**	122904.4243	124883.7692	985.5370	126585.2290
DE(1)	22.67	121813.4385	122503.1532	501.6266	123705.1952
DE(2)	22.85	129787.8713	123829.6008	373.2203	124699.9984
DE(3)	38.91	123137.8314	124201.9811	401.4912	125052.2981
DE-SQP(1)	20.56	**121715.4900**	121918.1695	137.4262	122267.0088
DE-SQP(2)	36.86	121723.4893	**121890.1630**	**120.1294**	**122189.3262**
DE-SQP(3)	38.90	121770.7606	122859.9637	588.7051	124449.2637

Table 3. Best result (50 runs) obtained for the case study using DE-SQP(1)

power	generation	power	generation
P_1	110.8005	P_{21}	523.2797
P_2	110.8007	P_{22}	523.2788
P_3	97.3993	P_{23}	523.2793
P_4	179.7329	P_{24}	523.2797
P_5	87.8001	P_{25}	523.2798
P_6	140.0000	P_{26}	523.2796
P_7	259.6012	P_{27}	10.0004
P_8	300.0000	P_{28}	10.0001
P_9	284.6004	P_{29}	10.0002
P_{10}	130.0002	P_{30}	96.7133
P_{11}	168.7978	P_{31}	190.0000
P_{12}	168.7997	P_{32}	190.0000
P_{13}	125.0005	P_{33}	190.0000
P_{14}	394.2792	P_{34}	200.0000
P_{15}	394.2796	P_{35}	164.8009
P_{16}	304.5190	P_{36}	200.0000
P_{17}	489.2794	P_{37}	110.0000
P_{18}	489.2800	P_{38}	110.0000
P_{19}	511.2789	P_{39}	110.0000
P_{20}	511.2796	P_{40}	511.2794

Table 3. Comparison of best result of this work for the fuel costs (f) presented in literature for the case study with 40 thermal units

technique	fuel costs (f) ($/h)
Evolutionary programming [6]	122624.350
Particle swarm optimization [9]	122930.450
Modified particle swarm [4]	122252.265
Hybrid evolutionary programming with SQP [9]	122379.630
Hybrid particle swarm with SQP [9]	122094.670
Best result of this paper using DE-SQP(1)	**121715.490**

5. Conclusion and Future Research

This paper proposed new hybrid approaches combining DE and SQP to solve economic dispatch problems of electric energy with the valve point effect. The proposed hybrid method uses the property of DE, which can provide a good solution even when the problem begins with many local optimal solutions. The SQP's local search property is used to find a final solution. Basically, the two hybrid methods tested can be divided into two parts. The first part employs DE to obtain a nearly global solution, while the second part employs SQP to determine the optimal solution.

In relation to procedure of solution of the economic dispatch problem of electric energy with effect of valve point, the results with the SQP and DE-SQP for optimization of the equations (1) and (2) were best that the results presented in Sinha *et al.* [6], Victoire and Jeyakumar [9] and Part *et al.* [4].

The performance of the DE and DE-SQP approaches used in the case study of 40 thermal units proved superior to conventional and non-conventional numerical methods, because these approaches provide a global solution, satisfying the constraints with a very high probability in an acceptable computing time.

SQP has therefore proved to be the best nonlinear programming method to solve constrained optimization problems. The method is sensitive to the initial point. It guarantees local optima as it follows a gradient search direction from the starting point towards the optimal point. When integrated with the DE (or DEC), the method produces better quality solutions than the ones found by these techniques when applied separately.

Hybrid methods combining DE and SQP can be very effective in solving economic dispatch problems with the valve-point effect. The search for the best combination of exploitation (convergence speed) and exploration (population diversity) is a constant in the DE, and our future studies will focus mainly on the conception of hybrid approaches with SQP for the solution of nonlinear economic dispatch problems of electric energy.

References

1. Boggs, P T, Tolle J W (2000) Sequential quadratic programming for large-scale nonlinear optimization. Journal of Computational and Applied Mathematics 124(1-2): 123-137
2. Fletcher R (1987). Practical methods of optimization, 2nd edition, John Wiley & Sons, New York, NY
3. Jin X, Reynolds R G (1999) Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach. IEEE Congress on Evolutionary Computation, Washington, DC, 1672-1678
4. Park J -B, Lee K -S, Shin J -R, Lee K Y (2005) A particle swarm optimization for economic dispatch with nonsmooth cost function. IEEE Transactions on Power Systems 20(1): 34-42
5. Reynolds R G (1994) An introduction to cultural algorithms. Proceedings of the 3rd Annual Conference on Evolutionary Programming, Sebald, A. V.; Fogel, L. J. (eds), World Scientific, River Edge, NJ, 131-139
6. Sinha N, Chakrabarti R, Chattopadhyay P K (2003) Evolutionary programming techniques for economic load dispatch. IEEE Transactions on Evolutionary Computation 7(1): 83-94
7. Storn R (1997) Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4): 341-359
8. Storn R, Price K (1995) Differential evolution: a simple and efficient adaptive scheme for global optimization over continuos spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA
9. Victoire T A A, Jeyakumar A E (2004) Hybrid PSO-SQP for economic dispatch with valve-point effect. Electric Power Systems Research 71(1): 51-59
10. Walters D C, Sheble G B (1993) Genetic algorithm solution of economic dispatch with valve point loading. IEEE Transactions on Power Systems 8(3): 1325-1332
11. Whitley D, Gordon V S, Mathias K (1994) Lamarckian evolution, the Baldwin effect and function optimization, Parallel Problem Solving from Nature, Davidor, Y.; Schwefel, H. -P.; Manner, R. (eds.), Springer-Verlag, 6-15
12. Wood A J, Wollenberg B F (1994) Power generation, operation and control, New York, John Wiley & Sons

Multiobjective Prioritization in the Analytic Hierarchy Process by Evolutionary Computing

Ludmil Mikhailov

School of Informatics, University of Manchester
Sackville Street, PO Box 88, Manchester M60 1QD, UK
E-mail: ludi.mikhailov@manchester.ac.uk

Abstract. This paper is concerned with the decision making problem of deriving weights from pairwise comparison judgments, in the framework of the Analytic Hierarchy Process. A new multi-criteria prioritization approach is proposed, minimizing the Euclidean norm and the Number of rank violations. The proposed method is implemented in a system for multiobjective prioritization and decision-making, based on evolutionary computing. The method and the system's performance are illustrated by an example, and the results are compared to those, obtained by a gradient search optimization algorithm.

Keywords: Multiobjective optimization, Analytic Hierarchy Process, Prioritization methods, Evolutionary computing, Genetic Algorithms.

1 Introduction

The assessment of weights of criteria and scores of alternatives is one of the most important tasks in the multicriteria decision-making. In the Analytical Hierarchy Process (AHP), proposed by Saaty [12], the values of weights and scores are assessed indirectly from comparison judgments. The elicitation process for both weights and scores is the same, so they are often called *priorities*.

The pairwise comparison process in the AHP assumes that the decision-maker can compare any two elements at a given hierarchical level and to provide a numerical value of the ratio of their importance. Comparing any two elements E_i and E_j, the decision-maker assigns a ratio a_{ij}, which represents a judgment concerning the relative importance of preference of the decision element E_i over E_j.

A full set of ratio-scale judgments for a level with n elements requires $n(n-1)/2$ comparisons. In order to derive a priority vector $w=(w_1,w_2,...,w_n)^T$ from a given set of judgments, Saaty constructs a positive reciprocal matrix $A=\{a_{ij}\}$. The Eigenvector prioritization method (EV) [11, 12] is based on some properties of the pairwise comparison matrices.

With the exception of the traditional EV method, all other methods for deriving priorities in the AHP are based on some optimization approach. Some optimal prioritization methods are the Goal programming (GP) [1], the Direct Least Squares and the Weighted Least Squares [2], the Logarithmic Least Squares [4], and the Fuzzy Preference Programming [10]. Those methods introduce an objective function, which measures the degree of approximation or the distance between the initial judgments and the solution ratios. Thus the problem of priority derivation is formulated as an optimization task of minimizing the objective function, subject to normalization and some additional constraints.

Despite the multicriteria nature of the requirements, regarding the properties of their solutions, all optimal prioritization methods optimize a single objective function. However, a single objective function cannot encompass and satisfy all requirements.

This paper proposes a new multiobjective optimization approach to prioritization. A two-objective prioritization (TOP) problem is formulated as an optimization task for minimization of the Euclidean norm and the number of rank violations. The TOP problem is transformed into a single-objective problem, which is solved by a system, based on a Genetic Algorithm (GA). A numerical example is given, which illustrates the performance of the system. The Pareto optimal solutions to the TOP problem are also compared to those, obtained by an gradient search algorithm.

2 Multiobjective Prioritization Problem

2.1 Optimization Criteria

Let $S=\{a_{ij} | j>i\}$ be a set of pairwise comparison judgments. The feasible set Q is defined as the set of all priority vectors $w=(w_1,...,w_n)^T$, which satisfy the normalization and non-negativity constraints:

$$Q = \left\{ (w_1, \ldots, w_n) \middle| w_i > 0, \sum_{i=1}^{n} w_i = 1 \right\} \tag{1}$$

The accuracy of the each priority vector $w \in Q$, approximately satisfying the comparison judgments can be measured by the Total deviation criterion

$$T(w) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (a_{ij} - \frac{w_i}{w_j})^2 \tag{2}$$

This criterion is equivalent to the squared Euclidean distance for the upper triangular part of a Saaty's reciprocal comparison matrix.

The rank preservation properties of the solutions can be measured by the Minimum violations criterion [7]:

$$MV = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} v_{ij} \tag{3}$$

where

$$v_{ij} = \begin{cases} 1, & \textit{if } w_i > w_j \textit{ and } a_{ij} < 1, \textit{ or } w_i < w_j \textit{ and } a_{ij} > 1, \\ 1/2, & \textit{if } w_i = w_j \textit{ and } a_{ij} \neq 1, \textit{ or } w_i \neq w_j \textit{ and } a_{ij} = 1, \\ 0, & \textit{otherwise.} \end{cases}$$

The Minimum violation criterion (3) can be represented in the following compact form:

$$V(w) = \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left| signum(a_{ij} - 1) - signum(\frac{w_j}{w_j} - 1) \right| \tag{4}$$

where the signum function is defined as:

$$signum(b) = \begin{cases} 1, & \textit{if } b > 1 \\ 0, & \textit{if } b = 0 \\ -1, & \textit{if } b < 1. \end{cases}$$

2.2 Statement of the Prioritization Problem

The two-objective prioritization (TOP) problem is to find a feasible priority vector that 'simultaneously' minimizes the Total deviation and the Number of violations:

$$\begin{aligned}&\text{minimize } T(w) \text{ and } V(w)\\&\text{subject to } w \in Q,\end{aligned} \tag{5}$$

where $T: R^n \to R^1$ and $V: R^n \to R^1$ are real-valued objective functions, defined by (2) and (4) correspondingly.

Each feasible vector $w \in Q$ determines a unique value of the objective function vector $y = (T(w), V(w))$. Therefore, the feasible set Q in the space of decision variables can be transformed into a *payoff set* Y in the two-dimensional objective space. The payoff set represents a feasible region of the admissible values of $T(w)$ and $V(w)$, and can be considered as the image of the feasible set Q in the objective space. The payoff set Y of the TOP problem consists in parallel line segments, as the function $V(w)$ takes non-negative discrete values in some range, and the function $T(w)$ is bounded.

Definition 1. A priority vector $w^* \in Q$ is said to be an *optimal (superior) solution* to the TOP problem (5), if it attains the minimum values of both criteria simultaneously.

Mathematically, w^* is an optimal solution to the problem if and only if $T(w^*) \leq T(w)$ and $V(w^*) \leq V(w)$ for all $w \in Q$.

Let T* and V* be global minima of the single objective problems:

$$T^* = \min\{T(w) | w \in Q\} \tag{6}$$

$$V^* = \min\{V(w) | w \in Q\} \tag{7}$$

Then $y^* = (T^*,\ V^*)$ is called an *ideal* point (or utopia point) of the problem (5). If the set of pairwise comparison judgments S is weakly transitive on non-transitive, the objectives (2) and (4) are conflicting and cannot be optimized simultaneously, so an optimal solution does not exist. Some compromise solutions, however, could be found.

Definition 2. A priority vector $w^o \in Q$ is said to be a Pareto optimal (or strongly non-dominated) solution, if there is no $w \in Q$ such that $T(w) \leq T(w^0)$ and $V(w) \leq V(w^0)$, with a strict inequality for at least one of these conditions.

The above definition implies that if w^o is a Pareto optimal solution to the TOP problem (5), the value of $T(w^0)$ cannot be decreased without causing a simultaneous increase in the Number of violations, and vice versa.

The objective vectors, corresponding to the Pareto optimal solutions $y = (T(w^0), V(w^0))$ form the Pareto optimal front $C = \{y = (T(w), V(w)) | w \in P\}$.

It is easy to be shown that *C* is on the boundary of the payoff set *Y*. Indeed, for any point $y = (T, V)$ in the interior of *Y*, reduction of *T* could be achieved by moving along the line *V*=const., towards the point *T*=0, until the boundary of *Y* is reached.

Since *Y* is non-connected set, *C* is a discrete curve in $\Re^2$, consisting of isolated points. It is also called a trade-off function, since it shows how much the value of *T* must change to stay in *C* when the value of *V* changes.

3 Solving the TOP Problem by Evolutionary Computing

Some classical Multicriteria Decision-making (MCDM) methods, which can be applied for finding Pareto optimal solutions to the TOP problem (5) are the Weighting Method, the -Constraint Method, the Goal Programming Method [13], or the Proper Equality Constraints method [9]. Generally, the main strength of classical MCDM methods is their efficiency and ability to generate strong Pareto optimal solutions [3]. However, these methods have some weaknesses as well [3, 5]:

- Difficulties in generating the Pareto optimal solutions might occur when specific problem knowledge is not available
- They cannot generate all Pareto optimal solutions with non-convex surfaces
- Several optimization runs are required to obtain an approximation set of Pareto optimal solutions.

Recently, the multiobjective genetic algorithms (GA) have become an alternative to the classical methods for generating Pareto optimal solutions; since they can eliminate some of the drawbacks of the classical methods. A detailed discussion on existing multiobjective GA can be found in [3, 5, 14].

Each of the existing multiobjective GA can be applied for solving the TOP problem (5). However, it is well known that the presence of constraints significantly affects the performance of multiobjective GA. Additionally, as opposed to the single objective case, the ranking of a population in the multiobjective case is not unique [6].

In order to solve efficiently the TOP problem (5), we apply the Weighting Method [15] and associate weights *k* and *(1-k)* to both objective functions, where $k \in (0,1)$. The values of the weight coefficients represent the relative importance of objectives. Thus we transform the TOP problem into a single-objective one, whose objective function is:

$$J(w) = kT(w) + (1-k)V(w) \tag{8}$$

As the second criterion *V(w)* can take discrete values only, by varying the value of *k* from 0 to 1, we can find all Pareto optimal solutions to the initial problem (5).

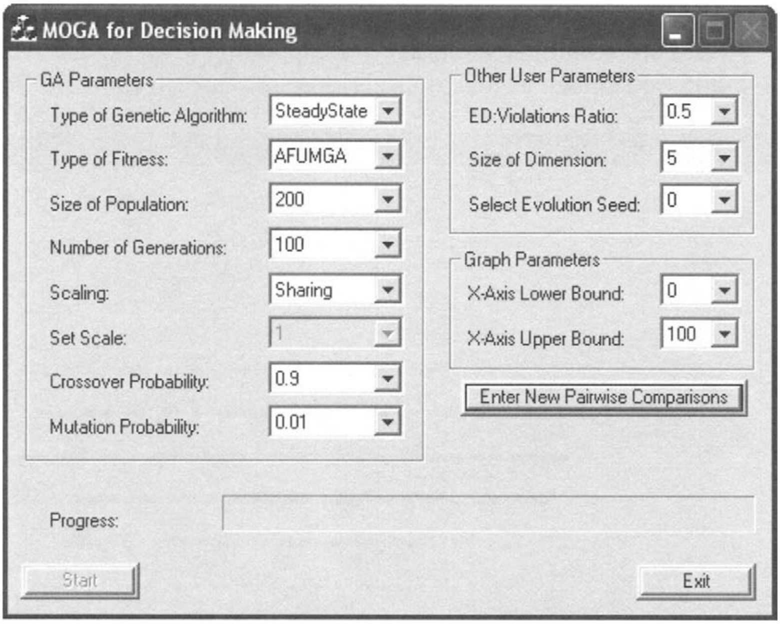
A Windows-based system for solving the modified single-objective TOP problem by using a GA was implemented. The GA performs the basic genetic operators, which are roulette wheel selection, a crossover with random mating and a simple mutation. Elitism has also been applied as an additional selection strategy, to make sure that the best performing chromosome always survives. The elitism has been realized by comparing the fitness of chromosomes from the current population and the fitness of the corresponding offspring. The fittest chromosome from the initial population survives for the next generation.

The user interface of the MOGA (Multi Objective Genetic Algorithm) system is shown on Fig. 1. The user selects the type of the algorithm and enters its parameters. Two types of GA are implemented: a simple, where at each generation the old population is completely replaced by new genomes, and a steady-state, in which only the worst genomes are replaced by new ones.

The size of the population and the number of generations are user-defined values, restricted to a range between 10 and 500. Similarly, the probabilities of the crossover and mutation are user-defined values. The applied selection scheme is the Roulette Wheel, where genomes with a better fitness have a greater chance of being selected. The selection scheme can be combined with a user-defined scaling method. We have implemented linear scaling, where the fitness function is multiplied by a user-defined value, and sharing scaling, where the scaling value is determined by measuring the similarity (the distance) between genomes.

The user enters also data, pertaining to the problem, as the number of the compared elements (dimension of the comparison matrix) and values of the actual comparisons.

In the implemented GA each chromosome is represented by a binary string of *n* components, associated with the *n*-dimensional priority vector *w*. At the beginning of each new generation, all chromosomes are normalized, so that the values of their genes sum up to one. The stopping condition is the number of generations. The experimental results show that the GA converges to the optimal solution for less than 50 generations.



Type of Genetic Algorithm: SteadyState
Type of Fitness: AFUMGA
Size of Population: 200
Number of Generations: 100
Scaling: Sharing
Set Scale: 1
Crossover Probability: 0.9
Mutation Probability: 0.01

ED:Violations Ratio: 0.5
Size of Dimension: 5
Select Evolution Seed: 0

X-Axis Lower Bound: 0
X-Axis Upper Bound: 100

Fig. 1. User interface of the MOGA system

The GA can be run for a single value of the coefficient *k*, corresponding to the user's preferences with respect of the importance of each objective, or for multiple valuses of this coefficient, equally spread in the range between 0 and 1. The increment is defined by the user. Our experiments show that an increment of 0.1, (i. e. running the GA 11 times) is enough to find all Pareto optimal solutions for prioritization problems with different dimensions.

4 An Illustrative Example

Consider a prioritisation problem with 5 unknown priorities. The comparison judgements provided by the decision maker are non-transitive:

$a_{12}=5$, $a_{13}=1/3$, $a_{14}=7$, $a_{15}=2/3$; $a_{23}=2$, $a_{24}=1/2$, $a_{25}=4$; $a_{34}=1/2$, $a_{35}=1/4$; $a_{45}=1/3$.

The MOGA system was used to solve this example. A steady state GA with sharing scaling was chosen, with a population size and a number of generations equal to 100. The crossover and the mutation rate were 0.9 and 0.1, respectively.

The results obtained are plotted in the 2-dimensional objective plane – see Fig. 2. The Pareto optimal solutions, corresponding to an integer Number of violations are shown as blue dots. Their values are given in Table 1.

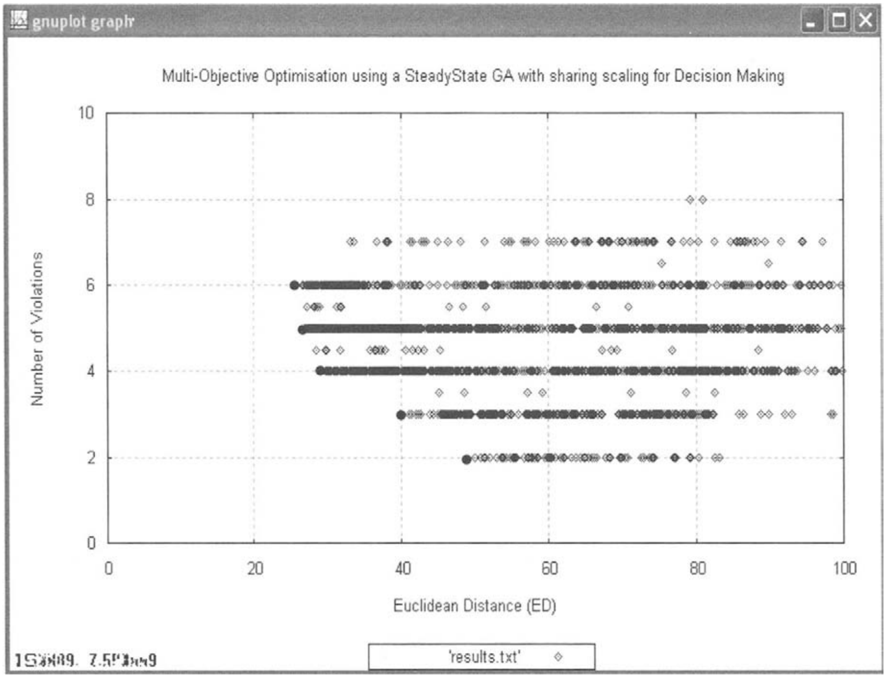


Fig. 2. Results from the GA and Pareto optimal solutions

Table 1. Pareto optimal solutions obtained by the GA

w_1	w_2	w_3	w_4	w_5	T	V
0.309	0.113	0.096	0.135	0.347	50.048	2
0.344	0.075	0.109	0.111	0.361	39.806	3
0.345	0.058	0.140	0.071	0.386	29.717	4
0.363	0.067	0.146	0.058	0.367	26.462	5
0.381	0.076	0.139	0.062	0.342	26.299	6

As it is seen from Fig. 2, the solutions with V=7 and V=8 are not Pareto optimal, as they are dominated by some other solutions. In fact, the Pareto optimal solution for V=6 is also the solution of the single-objective problem (6), giving the absolute minimum of the Euclidean distance. It is obtained for a value of k=1. Similarly, for k=0, the minimization of (8) is equivalent to the problem (7), and the obtained Pareto optimal solution has minimum violations, V=2. Due to the non-transitivity of the problem, there are no feasible solutions with less than two violations.

The same problem was solved using the gradient search algorithm of the Excel solver. The obtained solutions are shown in Table 2. It is seen that the GA solutions are better, with respect to the Euclidean distance criterion.

Table 2. Pareto optimal solutions obtained by a gradient search method

w_1	w_2	w_3	w_4	w_5	T	V
0.283	0.124	0.123	0.186	0.284	55.261	2.0
0.390	0.129	0.128	0.130	0.224	41.716	3.0
0.358	0.103	0.102	0.060	0.376	31.373	4.0
0.361	0.110	0.110	0.059	0.361	30.942	5.0
0.566	0.117	0.224	0.093	0.421	26.479	6.0

From the entire set of non-dominated solutions the decision maker can select one solution, which best satisfies his preferences. For example, an appropriate compromise between accuracy and number of violations can be achieved by the Pareto optimal solutions, corresponding to V=3 and V=4.

5 Conclusions

A new multiobjective prioritization method is proposed, which eliminates some of the limitations of the existing single objective methods. It minimizes the Euclidean norm and the Number of rank violations, thus ensuring a satisfactory accuracy and good rank preservation properties of the final set of Pareto optimal solutions. By exploring the specific characteristics of the multiobjective prioritization problem, the Pareto optimal solutions are obtained by a single-objective GA.

The comparison to a standard gradient search optimization method shows the advantages of this new approach to prioritization in the AHP.

Acknowledgement

The author is very grateful to the research student Mr. Evghenios Constantinou for his efforts in developing the MOGA system.

References

1. Bryson N (1995) A goal programming method for generating priority vectors. Journal of Operational Research Society 46:641-648
2. Chu A, Kalaba R, Springarn K (1979) A comparison of two methods for determining the weights of belonging to fuzzy sets. Journal of Optimisation Theory and Applications 27:531-541
3. Coello C (2000) An update survey of GA-based multiobjective optimization techniques. ACM Computing Survey 32:24-35
4. Crawford G, Williams C (1985) A note on the analysis of subjective judgement matrices. Journal of Mathematical Psychology 29: 387-405.
5. Deb K (1999) Evolutionary algorithms for multi-criterion optimization in engineering design. John Wiley & Sons, Chichester, UK
6. Fonseca C, Fleming P (1998) Multiobjective optimization and multiple constraints handling with evolutionary algorithms, part 1: A unified formulation. IEEE Transactions on SMC 28:26-37
7. Golany B, Kress M (1993) A multicriteria evaluation of methods for obtaining weights from ratio-scale matrices. European Journal of Operational Research 69:210-220
8. Goldberg D (1989) Genetic algorithms in search, optimization and machine learning. Addison Wesley
9. Lin J (1976) Multiple-objective problems: Pareto-optimal solutions by method of proper equality constraints. IEEE Transactions on Automatic Control 21:641-650
10. Mikhailov L (2000) A fuzzy programming method for deriving priorities in the Analytic Hierarchy Process. Journal of Operational Research Society 51:341-349
11. Saaty T, Vargas L (1984) Comparison of eigenvalue, logarithmic least squares and least squares methods in estimation ratios. Mathematical Modeling 5:309-324
12. Saaty T (1977) A scaling method for priorities in hierarchical structures. Journal of Mathematical Psychology 15:234-281
13. Sawaragi Y, Nakyama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, Orlando
14. Veldhuizen D, Lamont G (2000) Multiobjective evolutionary algorithms: analyzing the state-of-the-art. Evolutionary Computing 8:125-147
15. Zitzler E (1999) Evolutionary algorithms for multi objective optimisation: methods and applications. PhD Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland

Evolutionary and Heuristic Algorithms for Multiobjective 0-1 Knapsack Problem

Rajeev Kumar*, P. K. Singh, A. P. Singhal, and Atul Bhartia

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, WB 721 302, India
rkumar@cse.iitkgp.ernet.in

Summary. We consider a formulation of multiobjective 0-1 knapsack problem which involves a *single* knapsack. We solve this problem using multiobjective evolutionary algorithms (MOEAs), and quantify the solution-fronts obtained; we observe that they show good diversity and (local) convergence. Then, we consider two heuristic algorithms and observe that the quality of solutions obtained by MOEAs is much inferior. Interestingly, none of the MOEAs could yield the entire coverage of the Pareto-front. Therefore, we incorporate problem-specific knowledge in the initial population, and get good quality solutions using MOEAs too.

The main point we stress with this work is that, for real world applications of *unknown* nature, it is indeed difficult to realize how good/bad is the quality of the solutions obtained. Conversely, if we know the solution space, it is trivial to get the desired solution space using MOEAs, which is a paradox in itself.

1 Introduction

The 0 - 1 knapsack problem is a well-studied problem and research has been performed on many variants of the problem [1, 6]. There are many facets of the problem. We have to distinguish the single objective from the multiobjective combinatorial optimization problem [21]. Even the single objective case has been proven to be NP-hard. There are several effective approximation heuristics for solving knapsack problems, e.g., [6, 8]. Much research for the single objective case has been performed over the decades and the problem continues to be a challenging area of research.

In general, the multiobjective variant of the problem is *harder* than the single objective case. The multiobjective optimizer is expected to give a set of all *representative* equivalent and diverse solutions [2, 3] . The set of all optimal solutions is called the Pareto-front. Objectives to be *simultaneously* optimized may be mutually conflicting. Additionally, achieving proper diversity in the solutions while approaching convergence is another challenge in multiobjective optimization especially for

* Present Address: Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur, UP 208 016, India. Email: raj@iitk.ac.in.

unknown problems in black-box optimization. Moreover, the size of the obtained Pareto-front may be exponentially large.

Evolutionary algorithms (EAs) are emerging as a powerful black-box tool to solve combinatorial optimization problems. EAs use a randomized search technique with a *population* of individuals. The genetic operators used by EAs do not apply, in general, any problem-specific knowledge. However, special genetic operators may be designed by incorporating domain knowledge to expedite the search in certain applications. In the multiobjective scenario, we aim at effectively finding a set of diverse and mutually competitive solutions.

Zitzler and Thiele [21] pioneered the work for solving multiobjective 0-1 knapsack problem using EAs. They formulated the problem using m knapsacks, and maximized the profits simultaneously for all m knapsacks within weight constraints. Subsequently, many other researchers, (e.g., [10, 11, 16]) attempted to solve the same problem-formulation using many other variants of EAs. In this work, we consider another formulation of 0-1 knapsack problem which uses a *single* knapsack. The optimization problem is expressed with linear equations, though, the problem is NP-hard. We solve this problem-formulation using well-known MOEAs, and get interesting insight in to the EA problem solving strategy for real-world applications whose solution space is not known *a priori.*

For comparison of results obtained using MOEAs, we consider two heuristics and observe that the solutions obtained by the heuristics are much superior for larger problem instances than those obtained by MOEAs. Then, with the aim to get equally good solutions from MOEAs, we take a deeper look into the dynamics of population evolutions, and infer that the genetic operators are not strong enough to extend the coverage of the solution-front as obtained by the heuristics. Therefore, by applying this knowledge of the problem domain to improve upon the solutions obtained by MOEAs, we could get equally good solutions by MOEAs too.

Just getting quality solutions for a real-world application using MOEAs is not the main point we wish to highlight through this research monogram. Important point which we stress is that a problem solver has no idea about the quality of solutions that (s)he expects to get using EAs for those problems whose solution spaces are *unknown.* The available metrics to measure the quality of solutions in terms of convergence, diversity and extents are inadequate for such unknown problems.

The rest of the paper is organized as follows. In Section 2, we include a few definitions pertaining to multiobjective optimization, and formulate the 0-1 knapsack problem which we address in this paper. We include, in Section 3, two fast heuristics which we use to compare the solutions obtained by MOEAs. We present results obtained by MOEAs in Section 4. Finally, we draw conclusions in Section 5.

2 Basic Definitions and Problem Formulation

Definition 1. Multiobjective Optimization Problem (MOP) : *An m-objective optimization problem includes a set of n decision variables $\mathbf{X} = (\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_n)$, a set of m objective functions $F = \{f_1, f_2, ..., f_m\}$ and a set of k constraints*

$C = \{c_1, c_2, ..., c_k\}$. *The objectives and the constraints are functions of the decision variables.* The goal is to:

$$\begin{aligned}&\text{Maximize/Minimize} : F(\mathbf{X}) = \{\mathbf{f_1}(\mathbf{X}), \mathbf{f_2}(\mathbf{X}), ..., \mathbf{f_m}(\mathbf{X})\}\\ &\text{subject to satisfaction of the constraints:}\\ &C(\mathbf{X}) = \{\mathbf{c_1}(\mathbf{X}), \mathbf{c_2}(\mathbf{X}), ..., \mathbf{c_k}(\mathbf{X})\} \leq (\mathbf{0}, ..., \mathbf{0})\end{aligned}$$

The collection of decision variables constitutes the *decision space.* The set of objective values form the *objective space.* In some problem definitions the constraints are treated as objective functions. The objectives may also be treated as constraints to reduce the dimensionality of the objective-space.

Definition 2. Pareto-optimal Set : *Without loss of generality we assume an m-objective minimization problem. We say that a vector of decision variables $x \in X'$ is Pareto-optimal if there does not exist another $x^* \in X'$ such that $f_i(x^*) \leq f_i(x)$ for all $i = 1, 2, ..., m$ and $f_j(x^*) < f_j(x)$ for at least one j. Here, X' denotes the feasible region of the problem (i.e. where the constraints are satisfied.*

Definition 3. 0 - 1 **Knapsack Problem :** The *knapsack problem* with n items is described by the knapsack of size b and three sets of variables related to the items: decision variables $x_1, x_2, ..., x_n$; positive weights $W_1, W_2, ..., W_n$; and profits $P_1, P_2, ..., P_n$; where, for each $1 \leq i \leq n$, x_i is either 0 or 1. The W_i and P_i represent the weight and profit, as integers, of the i^{th} item respectively.

The single-objective knapsack problem can be formally stated as :

$$\begin{aligned}&\text{Maximize } P = \textstyle\sum_{i=1}^{n} P_i x_i\\ &\text{subject to } \textstyle\sum_{i=1}^{n} W_i x_i \leq b,\\ &\text{where } x_i = 0 \text{ or } 1\end{aligned}$$

We recast the above single-objective problem to a bi-objective knapsack problem of n items with two conflicting objectives (maximizing profits and minimizing weights). The bi-objective problem is formulated as :

$$\text{Maximize } P = \textstyle\sum_{j=1}^{n} P_j x_j \qquad \text{and} \qquad \text{Minimize } W = \textstyle\sum_{j=1}^{n} W_j x_j$$

Therefore, if the set of items is denoted by I, the aim is to find all sets $I_i \subseteq I$, such that for each I_j there is no other set which has a larger profit than profit (I_j) for the given weight bound $W(I_j)$. Hence, it is equivalent to finding the collections of items with the maximum profit in a knapsack with capacity $W(I_j)$.

The number of Pareto-optimal points can range from $n + 1$ to 2^n. Thus, for arbitrary values of the weights P_j and W_j the Pareto-optimal set can be exponential in n. This problem is shown to be NP-hard. For further details and analysis, see [13].

3 Heuristic Algorithms

In this section, we present two heuristics which we use to approximate solutions for bi-objective 0-1 knapsack problem.

3.1 Profit-Weight Ratio Heuristic

In this algorithm, we arrange the items in descending order of their profit to weight ratio (Algorithm 1). This has an $O(n \log n)$ computational complexity, and can yield a subset of n solution points.

Algorithm 1 Profit Weight Ratio

```
*Input*: $n$ - number of items, $W_i$ and $P_i$ - weight and profit of item $i$, respectively.
*Output*: $F$ - Approximation front (a set of tuples $(P_i, W_i)$)
*Algorithm*:
for $i = 1$ to $n$ do
    $R_i = P_i/W_i$
end for
Sort $R_i$ in descending order, let $P_{R_i}$ and $W_{R_i}$ be the profit and weight, respectively, of
    item $i$ in the sorted list
for $i = 1$ to $n$ do
    $P_i = P_{i-1} + P_{R_i}$
    $W_i = W_{i-1} + W_{R_i}$
end for
Output $F$
```

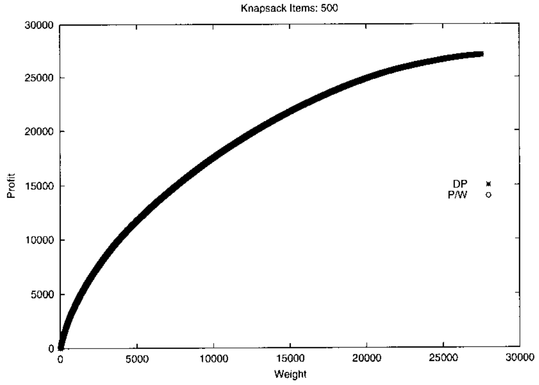


Fig. 1. 500-item knapsack. Pareto-front obtained by Algorithm 1 (profit-weight ratio) and Algorithm 2 (dynamic programming) completely overlap each other.

3.2 Solutions by Dynamic Programming

This algorithm 2 [19] uses optimal substructure, that is, if we remove item j from the optimal load, the remaining problem must be optimal for remaining $W - W_j$ pounds for overall solution to be optimal. Let $P[i, W]$ = value of solution for items $1..i$ and maximum weight W. This has an $O(nW)$ computational complexity.

Algorithm 2 Dynamic Programming

```
Input: $n$ - number of items, $W_i$ and $P_i$ - weight and profit of item $i$, respectively.
Output: $F$ - Approximation front (a set of tuples $(W, P[i, W])$)
Algorithm:
for $W = 0$ to $W_{total}$ do
    $P[0, W] = 0$
end for
for $i = 1$ to $n$ do
    $P[i, 0] = 0$
    for $W = 1$ to $W_{total}$ do
        if $W_i \leq W$ then
            if $P_i + P[i-1, W - W_i] > P[i-1, W]$ then
                $P[i, W] = P_i + P[i-1, W - W_i]$
            else
                $P[i, W] = P[i-1, W]$
            end if
        else
            $P[i, W] = P[i-1, W]$
        end if
    end for
end for
Output $F$
```

We applied the above two algorithms on knapsacks of 100, 250, 500 and 750 items each (data set is the same as used by Zitzler and Thiele [21]). The Pareto-front obtained from 500 item knapsack is shown in Fig. 1; both the fronts completely overlap each other, however, the solution-points obtained by Algorithm 2 are many times more than the number of points obtained by Algorithm 1.

4 Solution by Evolutionary Algorithms

We attempted to find solutions from EAs assuming we do not know the location of the desired Pareto-front from other algorithms. This is important because we run EAs without using the information about the location of the Pareto-front as the convergence criterion. For this, we considered three MOEAs, namely, There exist many algorithms/implementations which have been demonstrated to achieve diverse and equivalent solutions.

Most multiobjective evolutionary algorithms and implementations use some predetermined metrics (e.g., number of generational runs) as the stopping criterion. Other common metric used is the distance metric which finds distance of the obtained solution front from the true Pareto front; this is trivially done for *known* problems. Such a metric is based on a reference and, a true reference is not known for unknown problems. A commonly practiced approach to determine the reference for unknown problems is to extract the reference from the best-solutions obtained so far, and the reference is incrementally updated with every generation in iterative refinement based algorithms.

In this work, we consider three algorithms, namely, NSGA-II [5], SPEA2 [20] and PCGA [15]; first two are well known algorithms, and the third, PCGA makes use of rank-histogram(s) [14] to assess the convergence. Additionally, PCGA has the option of running multiple randomly generated populations sets to a convergence criterion which is assessed by the rank-histogram(s), however, all the results presented here are taken form a single set of randomly generated population.

Many metrics have been proposed for quantitative evaluation of the quality of solutions [2, 3]. Essentially, these metric are divided into three classes:

- Diversity : Sampling of the obtained solutions across the front,
- Convergence : Distance of the obtained front from the *reference* front, and
- Extent : Coverage of the obtained solutions across the front.

Some of the commonly used metrics are C-measure [21], R-measure [7], S-measure (hyper-volume) [21], Generational distance (GD) [18], Spread measure [3, 17], and Convergence measure [4]. Some of these metrics (e.g., generational distance, volume of space covered, error ratio measures of closeness of the Pareto-front to the true Pareto front) are only applicable where the solution is known. In case of unknown nature, the metrics are sensitive to the choice of the reference. Other metrics (e.g., ratio of non-dominated individuals, uniform distribution) quantify the Pareto-front and can only be used to assess diversity. Knowles & Corne gave a detailed critical review of these measures in his paper [12] and recommended use of some of the metrics as stable measures. They have also shown the sensitivity of some of the metrics with respect to the arbitrary choice of the reference point/front.

We quantified the solution front using the following four metrics: hypervolume, spread, convergence and C-measures. We assumed that the reference solution front is not known, therefore, we combined all the three fronts obtained by three MOEAs to serve as the reference for computation of the metrics.

We generated initial population from a uniform random generator. We varied the cross-over and mutation probabilities and the population size in an attempt to get the best solutions from each of the three MOEAs. The best solution fronts obtained from each of the three MOEAS along with the initial population are shown in Fig. 2. The quantification of the solution front is tabulated in Table 1. Apart from these three measures in the table, C-measure indicated almost full coverage for each of the three algorithms with respect to the solutions obtained by any other two algorithms.

If we look at the convergence metrics, they are very close to zero, and thus show a case of excellent convergence. However, since we have already computed these

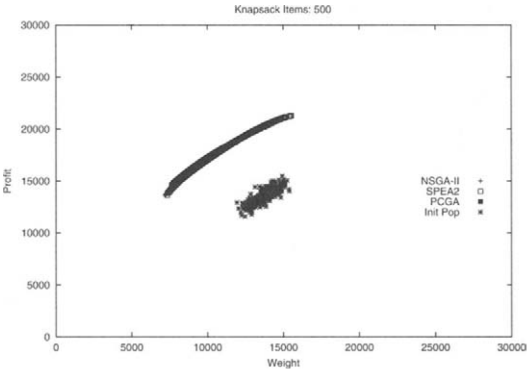


Fig. 2. 500 item knapsack. Solutions obtained from NSGA-II, SPEA2 and PCGA along with initial population. There does not exist any significant difference among all the three solution sets.

Table 1. Metrics for 500 item knapsack.

Algorithm	Hypervolume	Spread	Convergence
NSGA-II	387 M	0.619	0.014
SPEA2	383 M	0.219	0.010
PCGA	368 M	0.675	0.000

Pareto-front from other techniques, the quality of the solutions obtained by MOEAs is too inferior in respect of the coverage toward either ends of the front. Within the middle range of the front, the solution quality obtained by heuristics and EAs is comparable. Since, we know the solutions, next, we attempted how to improve upon the quality of the EA solutions towards either ends of the weights and profits.

Injecting Extreme Points in the Initial Population

We observed from Fig. 2 that the genetic operators used become weak in distributing the population along the end-points of the Pareto-front shown in Fig. 1. Second, the EA solutions lie within the middle portion of the solution front. If we look at the initial population, this is also centered at the middle portion of the solution space. This is natural because we use a uniformly random generator, therefore, the number of zeros and ones in each of the chromosomes are uniformly distributed. We can infer that the solutions which lie at the left end of the solution space are heavily skewed towards zeros, and solutions towards right end are skewed towards ones. EA operators could not explore the search space during the extreme ranges.

Understanding this dynamics, we injected few individuals having heavily skewed distribution of zeros and ones in their chromosomes in the initial population. With such specifics chromosomes, we used EAs again, and got solutions covering the entire space as shown in Fig. 1

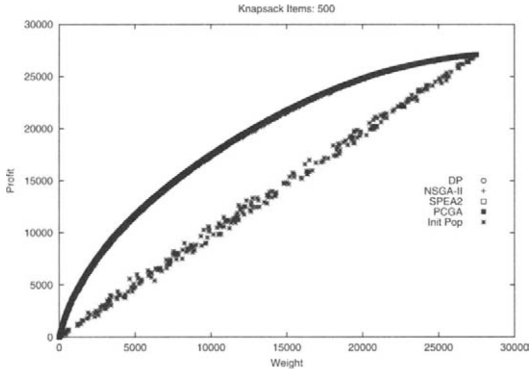


Fig. 3. 500 item knapsack. Solutions obtained from NSGA-II, SPEA2 and PCGA along with *uniformly distributed* initial population. Reference front from heuristics is also shown. As such there does not exist much differences.

Uniformly Distributed Initial Population

With the success of the above experiment, we generated initial population in such a way that the number of zeros and ones in the chromosomes gets distributed in the entire range of solution-front - see the initial population in Fig. 3. Then, we repeated the experiments to get the final solutions from each of the three MOEAs. Interestingly, with such an initial distribution of chromosomes, each of the three MOEAs could get the entire front at much *reduced computational cost*; the obtained solution set is comparable to the front obtained in Fig. 1. The computed metrics for assessing the quality are given in Table 2. Most of the numbers in this are comparable barring the spread metric; this may be due to the sensitivity of the metric, however, we do not aim to draw any inference from such differences across the MOEAs in this work.

Table 2. Metrics computed from Fig. 3 for 500 item knapsack

Algorithm	Hypervolume	Spread	Convergence
NSGA-II	517 M	0.529	0.005
SPEA2	521 M	0.169	0.001
PCGA	522 M	0.578	0.000

5 Discussion & Conclusions

In this work, we have taken a seemingly simple instance of a problem which is represented by just two linear equations in a single variable. We have shown, first, that each of the three MOEAs obtained good quality of solutions, both visually as well as quantitatively. The solution front was good till we were not aware of the solution front obtained from other algorithmic paradigms. For such a simple problem, there existed simple heuristics which obtained apparently the complete solution-front. On comparison, the best solutions discovered by EAs (black-box optimization) were too inferior. If we were not having the solution set obtained by other algorithm paradigms, we were mislead that EAs had obtained quality solutions. (In this work, we considered three different MOEAs so that we can explore the solution space in best possible ways, and get whatever the best can be obtained; we did not aim to compare the solutions obtained by each of these three algorithms.)

However, on knowing the solution space, we embedded this simple knowledge of the solutions in to EAs' solution-evolving process, and we could easily get the complete solution front which is comparable with the best Pareto-front. This is a paradox in itself that for solving a problem *per se* we need to know the solution space before hand. In a recent research, Ishibuchi and Narukawa [9] observed that good solutions are not always obtained by MOEAs, and proposed use of hybrid approaches.

The main agenda that we stress with this work is: can we effectively solve *unknown* problems using black-box optimization technique of EA, especially in a multi-objective setting? All the metrics for convergence look for some reference set which is mostly not available for *harder* unknown problems. Most diversity measuring metrics aim at achieving uniformly distributed solution set; if the actual solution set itself is not uniformly distributed the measurements would be poorer.

Apparently, the obvious question is: how can one trust on the solutions obtained for real-world applications (RWAs) by such black-box optimization techniques? Can we effectively approximate the quality of solutions for such unknown problems?

Acknowledgement

A part of the research is supported from the Ministry of Human Resource Development (MHRD), Government of India project grant.

References

1. I. Asho. Interactive Knapsacks : Theory and Applications. Ph.D. Thesis, Tech Report No.: A-2002-13, Department of Computer and Information Sciences, University of Tampere, 2002.
2. C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multiojective Problems*. Boston, MA: Kluwer, 2002.
3. K. Deb. *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.
4. K. Deb and S. Jain. Running Performance Metrics for Evolutionary Multiobjective Optimization. In *Proc. 4th Asia-Pacific Conf. Simulated Evoluation and Learning (SEAL 02)*, pages 13–20, Singapore, Nov 2002.

5. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182 – 197, April 2002.
6. T. Erlebach, H. Kellerer, and U. Pferschy. Approximating Multiobjective Knapsack Problems. *Management Science*, 48(12):1603 – 1612, 2002.
7. M. P. Hansen and A. Jaszkiewicz. Evaluating the Quality of Approximations to the Nondominated Set. Tech. Rep. IMM-REP-1998-7, Tech. Univ. Denmark, March 1998.
8. O. H. Ibarra and C. E. Kim. Fast Approximation Algorithms for the Knapsack and Sum of Subset Problem. *J. ACM*, 22:463 – 468, 1984.
9. H. Ishibuchi and K. Narukawa. Comparison of for Evolutionary Multiobjective Optimization with Reference Solution-Based Single-Objective Approach. In *Proc. Genetic and Evolutionary Computations Conference (GECCO 05)*, pages 787 – 794, ACM, June 2005.
10. A. Jaszkiewicz. On the Performance of Multiobjective Genetic Local Search on the 0-1 Knapsack Problem - A Comparative Experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402 – 412, August 2002.
11. J. D. Knowles and D. W. Corne. M-PAES: A Memetic Algorithm for Multiobjective Optimization. In *Proc. IEEE Conference on Evolutionary Computation*, pages 325 – 322, Las Vegas, Nevada, USA, July 2000.
12. J. D. Knowles and D. W. Corne. On Metrics for Comparing Nondominated Sets. In *Proc. Congress Evolutionary Computation (CEC '02)*, volume I, pages 711–716, Piscataway, NJ, May 2002. IEEE.
13. R. Kumar and Nlanjan Banerjee. Running Time Analysis of a Multiobjective Evolutionary Algorithm on Simple and Hard Problems. In *Proc. Foundations of Genetic Algorithms (FOGA-05), LNCS 3469*, pages 112 – 131, 2005.
14. R. Kumar and P. I. Rockett. Assessing the Convergence of Rank-based Multiobjective Genetic Algorithms. In *Proc. 2nd IEE/IEEE Int. Conf. Genetic Algorithms in Engineering Systems: Innovations and Applications (Galesia 97)*, volume I446, pages 19 – 23. IEE, London, UK, Sep. 1997.
15. R. Kumar and P. I. Rockett. Improved Sampling of the Pareto-front in Multiobjective Genetic Optimization by Steady-State Evolution: A Pareto Converging Genetic Algorithm. *Evolutionary Computation*, 10(3):283 – 314, 2002.
16. G. Raidl. An Improved Genetic Algorithm for the Multiconstrained 0-1 Knapsack Problem. In *Proc. IEEE Conference on Evolutionary Computation*, pages 207 – 211, May 1998.
17. J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithms. Masters Thesis, Department of Aeronautics and Astronautics, MIT, Massachusetts, 1995.
18. D. A. van Veldhuizen. Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations. Ph.D. Thesis, Technical Report No. AFIT/DS/ENG/99-01, Air Force Institute of Technology, Dayton, OH, 1999.
19. Web. Dynamic Programming for 0-1 Knapsack Problem. [http://www.cse.uta.edu/holder/ couses/cse2320/lectures/l15/node12.html].
20. E. Zitzler, M. Laumanns, , and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *Proc. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN)*, Sep. 2001.
21. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. Evolutionary Computation*, 3:257 – 271, 1999.

The Assignment of Referees to WSC10 Submissions: An Evolutionary Approach

Joshua Knowles

University of Manchester, Manchester M60 1QD, UK
j.knowles@manchester.ac.uk

Summary. The method used to assign referees to the manuscripts submitted to the 10th online World Conference on Soft Computing (WSC10) is described. Each of the 94 manuscripts received was associated with three unique reviewers from a program committee of 78 persons. A fairly advanced, hand-tuned fitness function based on inverse keyword-frequencies, keyword coverage, and penalties for referee-overuse was used to evolve a satisfactory assignment, via a standard evolutionary algorithm. The resulting referee-to-manuscript matches were hand-adjusted in a small number of cases. In total, only seven assignments (out of more than 280) were declined by referees, and the return-rate for completed reviews was 81%.

Keywords: Evolutionary algorithm, constrained optimization, assignment problem, transportation problem, penalty methods

1 Introduction

The assignment of reviewers to submitted manuscripts is a problem that arises at every technical conference that referees its papers. Much of the time human judgement is still used to complete this job but as technical conferences grow in size and diversity of content, and become more frequent, the use of automated methods is becoming more appealing, and may soon be a necessity.

The WSC10 conference is a typical case in point. Although it is still relatively small, it has been growing steadily and this year the program committee comprises an additional 20 referees. Moreover, the fairly broad umbrella that is Soft Computing means that many referees would feel unqualified to review more than a small minority of the submitted papers. Both of these facts, combined with the need to obtain a workable assignment in just a short space of time led us to develop a simple, (semi-)automated assignment method.

Although many approaches are possible, including treating the problem as an IP and solving it by enumeration methods such as branch-and-bound, an important factor here was a short development time, so a simpler evolutionary algorithm approach was favoured. Nonetheless, the fitness function used to evaluate the complete-assignment is more advanced than has been considered in other recent evolutionary algorithms for this problem [9], and represents the main contribution of the work.

1.1 Related Work

Roughly speaking, there are two distinct parts to the referee-manuscript assignment problem: one is the assessment of the suitability of each referee to review each manuscript; the other is actually solving the assignment problem itself, such that some function of overall suitability is maximized and all constraints are satisfied. These can be treated together (as we shall do) or as two phases to be completed independently.

Traditionally, the Program Chair (or whoever is in charge of refereeing) would know each of the referees and could scan the papers to assess suitability. But, as conferences have grown, the use of keywords has become necessary, as a kind of crude indicator of a match between a referee and a manuscript. More recently, some advanced methods of assigning a match-score to a referee-manuscript pair have been investigated; these eschew keywords in favour of document-retrieval methods based on comparing online abstracts from the referees and the full-text of the submitted manuscripts [6, 1, 2, 10]. These state-of-the-art methods may be preferable to simple keyword-matching strategies, and probably represent the future, but it is a time-consuming business to obtain full texts and abstracts from the Web, in reality, unless robust, fully-automated Web-trawling agents have been developed. Therefore, we opted for a middle-ground approach that relies on keywords but takes into account (to some extent) their information content, rather than just simply rewarding all keyword matches equally. The method that we use for this has some resemblance to the term frequency - inverse document frequency (tf-idf) [3] method often used in document retrieval (e.g. search engines).

In the case where match-weights have already been associated between each referee and each manuscript, the problem of actually carrying out the full assignment can be seen as a variant of the generalized assignment problem (GAP). GAP is *NP*-hard but can be solved exactly using advanced enumeration methods for problems of a reasonable size [4], or approximately, using heuristic methods for larger, difficult instances with correlated flows [7]. Hartvigsen et al [8] formulate referee assignment as a particular type of transportation problem, which is then decomposed into a large number of integer programming problems.

Although exact methods such as those in [4, 8] are possible when weights between a manuscript and a paper have been assigned in a separate phase, we think that this approach does not account for the possibility of synergy between the referees assigned to a manuscript. It may be that while each referee assigned to the same paper might not match some of its aspects, together the referees cover all aspects of the paper. The approach we propose herein can reward such assignments because our fitness function assesses referee assignments in the context of other assignments.

2 Problem and Method

2.1 The Referees

Most of the referees for this year's conference have served on earlier WSC program committees. A further twenty referees known to the author personally were added for this year, giving a total of 78 persons. At no stage was any attempt made to obtain a specific balance to the program committee (PC) in terms of the different aspects of soft computing. Referees were not asked to provide their own keywords, so to do this initially, the webpages of all referees were visited to 'farm' keywords from publications lists or 'research interests'. Later, this initial referee-keyword assignment was tuned, once manuscripts were received (see below).

2.2 The Manuscripts

Submitted papers were received (more or less) in two separate batches: 75 papers in a first batch that were submitted before the general conference deadline, and a second batch of 19 papers mainly submitted to special sessions with later deadlines, received a few days later. At the time of receiving the first batch, the size and details of the second batch was not known, and it was desirable to assign referees to the first batch quickly (within at most five days). Hence, the algorithm for doing the assignment was developed for the first batch, run, and papers for these were sent out to the referees. Only afterwards was the second batch dealt with, taking into account the numbers of papers already assigned to each referee. Most papers included a set of keywords that had been chosen by the authors completely freely.

2.3 Creating a Better Keyword List

Many of the keywords provided with the initial 75 manuscripts were far too esoteric, e.g. 'H_2-norms', while some others were not specific enough, e.g. 'soft computing'. A different problem, degeneracy, occurs with the referee keywords farmed from webpages: many different keywords can be used for essentially the same thing, e.g. 'fuzzy-neuro' and 'neuro-fuzzy'. To create a keyword list common to both manuscripts and referees that would circumvent these problems, the frequency of every keyword was first calculated, for both groups, and displayed. Manual changes were then made to remove certain keywords (the too general and the too specific), while similes were merged under one heading, in most cases. In the end, a list of 62 keywords was derived and each manuscript and referee was updated to use only these terms. This was the most labour-intensive part of the whole reviewing process but it is difficult to automate this process effectively. The same list of keywords was used for the second round of reviewing. The list with both the referee and the manuscript frequencies (overall from the 94 papers) is shown in Table 1.

Table 1. The list of 62 keywords that was manually created, and the frequency of each with respect to the list of 94 papers and 78 referees. The mean number of keywords per paper is 2.79, and the mean number of keywords per referee is 2.91. The most difficult keywords — having the highest ratio $f_{\text{papers}} : f_{\text{referees}}$ are 'retrieval'; and 'time series', with a ratio of 6:1

Keyword	f_{papers}	f_{referees}	*Keyword*	f_{papers}	f_{referees}
adaptation	3	3	image processing	13	9
ant algorithms	3	8	kernel methods	1	1
autonomous robotics	9	11	machine learning	5	4
bioinformatics	1	6	metaheuristics	6	4
chaos theory	2	2	multiobjective optimization	7	9
classification	6	3	multi-agent	3	9
clustering	6	6	networks	6	3
coevolution	2	1	neural networks	15	19
combinatorial optimization	7	3	optical character recognition	1	1
control	7	7	operations research	1	1
credit scoring	1	2	parallel metaheuristics	1	1
data mining	6	9	particle swarm	2	1
design	1	1	pattern recognition	2	6
differential evolution	3	2	reinforcement learning	1	2
evolutionary neural networks	3	1	representations	5	2
engineering	5	2	retrieval	6	1
evolutionary computing	20	13	signal processing	1	1
evolutionary computing in medicine	1	1	self-organizing maps	1	3
evolution strategies	1	1	splines	2	1
evolvable hardware	2	2	statistics	1	1
fault diagnosis	3	1	supply chain management	3	2
FPGAs	2	3	surrogate models	1	1
fuzzy	12	19	texture analysis	1	1
fuzzy classifiers	2	2	time series	6	1
fuzzy control	8	4	timetabling and scheduling	5	3
fuzzy-evolutionary	2	5	transport	3	1
fuzzy-neuro	7	4	uncertainty	4	7
game theory	1	1	vision	8	3
genetic algorithms	11	7	visualization	7	2
graph theory	3	3	wavelets	1	1
image classification	1	1	wireless networks	2	1

2.4 Early Algorithm Design Choices: Encoding, Operators and General Approach

The main objective behind most of the algorithm design choices made in the early stages was to reduce development time and the risk of a complete failure to obtain a working assignment. Thus, a simple evolutionary algorithm (EA) approach was preferred because this does not require a precise formulation of the problem beforehand, and facilitates making several attempts by simply changing the fitness function. To further keep development time down, no heuristic or greedy method of initialization was considered: the EA would start from a population of random assignments.

With initially 75 papers to assign to 78 referees, an allocation of three referees to each manuscript seemed most logical because this would give slightly less than three papers to review, for each member of the program committee (PC), on the average. This would give enough leeway to accommodate the additional manuscripts expected, without over-burdening any referee, while giving a sufficient basis for paper acceptance/rejection decisions.

A simple fixed-length coding (Figure 1) was thus selected, in which the assignment to each manuscript is represented by precisely three unique referee ID numbers. The total unconstrained search space for this representation is then given by:

$$\binom{78}{3}^{75} = 76076^{75}, \tag{1}$$

which is approximately 10^{366}.

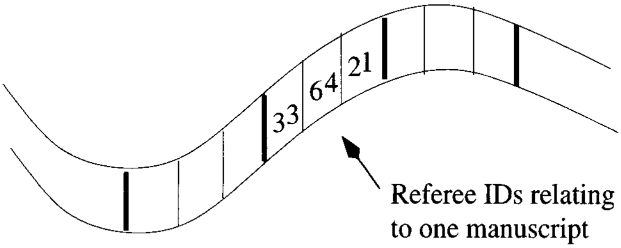


Fig. 1. The assignment of papers was represented using a chromosome of length $L = 3N$, where N is the number of manuscripts. Each locus codes for a referee ID; the three IDs relating to the same manuscript must be different from one another.

Initialization is performed by assigning referee IDs randomly to the papers, with the constraint that the three referees belonging to the same manuscript are not the same.

The mutation operator changes a single locus of the chromosome, i.e. referee ID, to another random value. If the referee ID is not unique to that paper, another ID is chosen at random until a unique ID is obtained. No crossover operator was used.

A default evolutionary algorithm configuration was chosen to begin exploring possible solutions. This configuration uses a population size of 100 and a 'steady-state' update of the population. In each 'generation' one individual is selected uniformly at random (without taking into account fitness), cloned and mutated, and replaces its 'parent', if it is fitter. If the mutant is not fitter than its parent, then it is compared with its two nearest neighbours in the population list (the list is circular to remove edge effects). If it is fitter than either of them, it replaces one at random, else it is deleted. This local replacement strategy is inspired by [5].

2.5 Dummy Runs and Visualization of Results

A very crude fitness function was used to obtain and explore initial solutions. This fitness function had just two terms:

$$f = \text{total \#keyword-matches} - \sum_{j \in \mathrm{PC}} (\text{\#assignments}(j) - 3)^2, \qquad (2)$$

where PC is the set of referees in the program committee.

Results were visualized by printing a text file with, for each manuscript, the title, authors and keywords displayed, followed by the three selected referees and their associated keywords. Several runs were made with this fitness function, adjusting the population size and stopping the evolutionary algorithm when convergence of the fitness level seemed to occur. It was immediately obvious from observing the output file that although many matching keywords were occurring, the less frequent keywords were hardly ever matched, and for many papers, the set of keywords were not being covered by the referees. Additionally, many referees were being assigned to papers having no keywords in common at all; often one of the referees out of the three for a paper was a 'redundant' referee in this sense. Finally, the term penalizing referees for any departure from 3 assignments each was not satisfactory: this was unnecessarily stringent. One pleasing observation along with these was that no checks to prevent a referee from being assigned their own paper was necessary, as this occurred very infrequently and could be corrected manually.

At this stage, a multiobjective optimization approach was briefly considered but rejected on the basis that it would be too difficult to visualize very many tradeoff solutions given the crude text-file method described above. Thus, I opted for a fitness function summing up three terms, with weights on each term. The first term would reward good matches, the second would penalize redundant referees and the third would penalize over-working any referee.

2.6 Rewarding Good Matches

An advanced method for rewarding good matches was designed, taking account of the following observations:

- The less frequently a keyword appears in the set of manuscripts, the more reward should be given when that keyword is matched, since it probably relates to a more specific topic and is thus more important in characterizing the manuscript
- The less frequently a keyword appears in the set of referees, the more reward should be given for matching that keyword because few referees can make that match.
- The total reward available per paper should not differ greatly with the number of keywords it uses, or their frequencies
- The coverage of all keywords is important: penalties should be given if any of a manuscript's keywords is not matched by at least one referee.

These led to the following reward term R of the fitness function:

$$R = \sum_{m \in M} \sum_{k \in \text{keywd}(m)} r(h(k)), \qquad \text{with} \tag{3}$$

$$r(h(k)) = \begin{cases} h(k) \cdot \left(\frac{1}{f_{\text{man}}(k)}\right) \cdot \left(\frac{1}{f_{\text{ref}}(k)}\right) \cdot \left(\frac{\text{rank}(k,m)}{\text{total}(k,m)}\right), & \text{if } h(k) > 0 \\ -\left(\frac{\text{rank}(k,m)}{\text{total}(k,m)}\right) & \text{otherwise,} \end{cases} \tag{4}$$

where M is the set of manuscripts; keywd(m) is the set of keywords associated with manuscript m; $h(k)$ is the number of referees $\in \{0,1,2,3\}$ that match keyword k in the assignment; $f_{\text{man}}(k)$ is the frequency of the keyword k within the set of manuscripts, M; $f_{\text{ref}}(k)$ is the the frequency of the keyword k within the set of referees; rank(k,m) is the position in a list of the keywords of manuscript m, sorted in ascending order by their $f_{\text{man}}(k)$ values (so tied values yield different ranks); and total(k,m) is the total number of keywords of the manuscript. The rank/total term means that a match of the fifth lowest-frequency keyword from five associated with a manuscript, only contributes less than a fifth as much as a match of the least frequent word. Moreover, assuming all frequencies are five, a manuscript with two keywords, each matched by all three assigned referees will score

$$\left(\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{1}{1}\right) \cdot \left(\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{1}{2}\right) \cdot 3 = \frac{9}{50},$$

while a manuscript with five keywords matched by all referees will score

$$\left(\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{1}{1}\right) \cdot \left(\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{1}{2}\right) \cdots \left(\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{1}{5}\right) \cdot 3 = \frac{20}{50},$$

just over twice as much. In practice, when the keyword frequencies get progressively smaller, this difference becomes even less, so the total reward available per paper does not vary by more than about 50%. Finally, if any keyword is not matched at all then there is a penalty, the magnitude of which relates to its rank.

2.7 Penalizing for 'Redundant' Referees

The penalty term S penalizes for every referee assignment where no manuscript keywords match the referee's keywords:

$$S = -\sum_{m \in M} \sum_{j \in \text{refs}(m)} 1 - \min(1, \text{match}(m,j)) \tag{5}$$

where M is the set of manuscripts, refs(m) is the set of referees assigned to manuscript m, and match(m,j) returns the number of keywords that referee j has in common with manuscript m.

2.8 Penalizing for Over-use of a Referee

A soft limit to the maximum number of papers assigned to any one referee was set to five. In addition, some referees specified that they could not referee more than x papers (some notified me of this after the first batch of assignments had been mailed out, and some had already specified this at the time of joining the PC). The penalty T for over-use of a referee was thus:

$$T = -\sum_{j \in \mathrm{PC}} v(j), \qquad \text{with} \tag{6}$$

$$v(j) = \begin{cases} \max(0, \#\text{assignments}(j) - 5) & \text{if } x(j) \text{ not specified} \\ 10.\max(0, \#\text{assignments}(j) - x(j)) & \text{otherwise,} \end{cases} \tag{7}$$

where PC is the set of referees in the program committee, as before.

2.9 Putting it all Together

The three separate terms of the fitness function defined above were put together in a single equation, as follows:

$$F = \rho \cdot R + \sigma \cdot S + \tau \cdot T. \tag{8}$$

Several runs were conducted with different choices of the weighting coefficients ρ, σ and τ. The final values used were:

$$\rho = 1, \qquad \sigma = 100, \qquad \tau = 1000. \tag{9}$$

3 Results

For the first batch of 78 papers, several runs were performed with different population sizes to assess the effect of this variable on performance. Figure 2 shows plots of the evolution of fitness with different population sizes for up to 1 million fitness evaluations. With the low selection pressure of the EA used, a population size of 10 is adequate and, in fact, gives the best results. The correlation with the keyword coverage is shown by the plot on the right of the figure, which shows the same runs. The assignment for this first batch had 86% of all keywords in the manuscripts matched by at least one referee.

The final assignment also achieved a good balance in terms of workload for each reviewer. The distribution of this workload is given in Figure 3, this time for the entire 94 manuscripts.

Manual Adjustments

A small number of adjustments were made because on two occasions a referee was assigned to their own paper, and because one or two obviously good referee-swaps could be made.

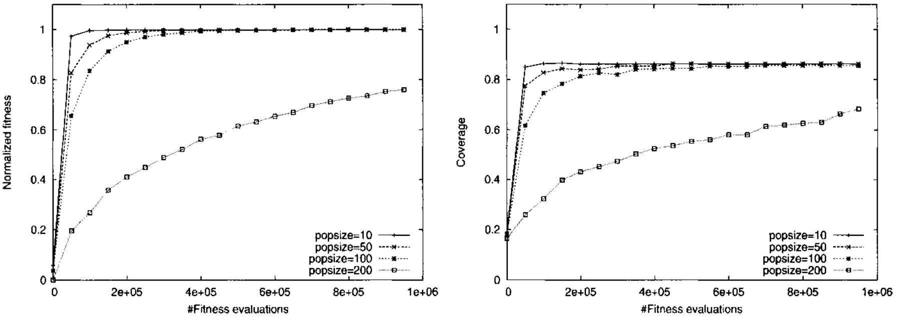


Fig. 2. The evolution of fitness (left) and keyword coverage (right) with different population sizes for the first batch of 75 papers. Results are for one run only but there is very little variation between runs. There is good correlation between the two measurements, while the fitness function additionally encourages more important (less frequent) keywords to be prioritised and ensures no referee is overworked.

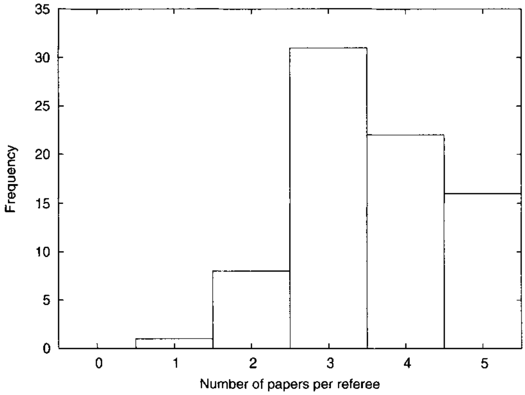


Fig. 3. Histogram showing the distribution of referee-workload in numbers of papers, for the entire final assignment of 94 manuscripts.

4 Summary and Discussion

Judging from the reports returned by referees, most of the assignments have been successful in practice, and there was a very low rate of declining reviews. However, we did not include a question to ask reviewers about their confidence/expertise on the review-form and this should be done in future. We should also consider the use of a web-based conference tool where referees and authors are asked to choose keywords from a specified list (in addition to giving a free choice) in order to build the keyword-lists more easily. Finally, it may be worth adding an extra term in the fitness function to assess the effect of some reviewers being unable to complete their assignments; this happened to us and it turned out that with only a small number of reviewer-absences, several papers had only one or zero reviews, initially.

Acknowledgments

Joshua Knowles is supported by a David Phillips Fellowship from the Biotechnology and Biological Sciences Research Council (BBSRC), UK. Thanks is also due to Ashutosh Tiwari and Erel Avineri for making the paper assignments possible, and to the whole program committee of WSC10 for their help in reviewing submissions.

References

[1] K. Antonina, B. Barbro, V. Hannu, T. Jarmo, and V. Ari. Prototype-matching system for allocating conference papers. In *Proceedings of the 36th International Conference on System Sciences (HICSS'03)*. IEEE Press, 2003.

[2] C. Basu, H. Hirsh, W. W. Cohen, and C. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research*, 1:231–252, 2001.

[3] C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report TR87-881, Cornell University, Ithaca, NY, 1987.

[4] D.G. Cattrysse and L.N. van Wassenhove. A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, 60:260–272, 1992.

[5] R. J. Collins and D. R. Jefferson. Selection in massively parallel genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 249–256. Morgan Kaufmann, 1991.

[6] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the 15th Annual International SIGIR*, pages 233–244. ACM, 1992.

[7] GR Raidl H Feltl. An improved hybrid genetic algorithm for the generalized assignment problem. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 990–995, 2004.

[8] David Hartvigsen, Jerry C. Wei, and Richard Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30:865–876, 1999.

[9] J.J. Merelo-Guervós and P. Castillo-Valdivieso. Conference paper assignment using a combined greedy/evolutionary algorithm. In *Parallel Problem Solving from Nature (PPSN VIII)*, number 3242 in LNCS, pages 602–611. Springer, 2004.

[10] D. Yarowsky and R. Florian. Taking the load off the conference chairs: towards a digital paper-routing assistant. In *Proceedings of the 1999 Joint SIGDAT Emprical Methods in NLP and Very-Large Corpora*, 1999.

Part VIII

Optimization Methods: Development and Analysis

Robustness using Multi-Objective Evolutionary Algorithms

A. Gaspar-Cunha, J.A. Covas

IPC – Institute of Polymer and Composites, University of Minho, Portugal
agc@dep.uminho. pt, jcovas@dep.uminho.pt

Abstract. In this work a method to take into account the robustness of the solutions during multi-objective optimization using a Multi-Objective Evolutionary Algorithm (MOEA) was presented. The proposed methodology was applied to several benchmark single and multi-objective optimization problems. A combination of robustness measures and the use of the Reduced Pareto Set Genetic Algorithm with Elitism (RPSGAe), that is an algorithm that distributes the solutions uniformly along the Pareto frontier, provided good results and are expected to be adequate for "real" optimization problems.

Keywords: Multi-objective evolutionary algorithms, robustness,

1 Introduction

Most "real-world" engineering optimization problems are multi-objective, i.e., their solution must consider simultaneously various performance criteria (which are often conflicting). The set of solutions corresponding to all possible criteria combinations (the Pareto frontier) can be obtained using Evolutionary Multi-Objective Algorithms (EMOAs), which are particularly adequate for this purpose, as they work with a population (vectors or solutions) rather than with a single point [1-7]. Pareto frontiers represent the trade-off between the criteria and the decision variables [8, 9].

Simultaneously, the practical solutions to those engineering optimization problems should be robust, *i.e.*, the performance of the optimal solution should be little affected by small changes of the design variables or of environmental parameters. Usually, the traditional optimization algorithms assume these variables as constant. However, robustness and performance can be conflicting and so it is important to know their relation for each optimization problem, *i.e.*, optimization rithms should not only find the solutions that maximize performance, but also that guarantee their robustness. Thus, the robustness can be taken into account as the search proceeds, by introducing a measure of robustness during the optimization. This can be done in two ways, by replacing the original objective function by an expression which measures both the performance and the expectation of each criterion in the vicinity of a specific solution, or by inserting an additional optimization criterion assessing robustness [10]. Since, as noted above, robustness and performance are often conflicting, they can also be approached as a multi-objective optimization problem, where the trade-off between performance and robustness is to be obtained [10, 11].

Robustness can consider the variations of design variables, which are done in the majority of the situations [10-13], or can consider environmental parameters [14]. Robustness analyses have been applied to single objective optimization problems [10, 12, 13], but a comparative study of their performance is lacking. Moreover, their extrapolation to multi-objective problems is much rarer and rather introductory [11].

2 Measures of Robustness

A multi-objective maximization problem can be defined as follows:

$$\begin{array}{llll} \max\limits_{x_l} & f_m(x_l) & l=1,\cdots,L & m=1,\cdots,M \\ \text{subject to} & g_j(x_l)=0 & j=1,\cdots,J & \\ & h_k(x_l)\geq 0 & k=1,\cdots,K & \end{array} \qquad (1)$$

where f_m are the M objective functions of the L parameters x_l, and g_j and h_k are the J equality ($J \geq 0$) and K inequality ($K \geq 0$) constraints, respectively. As discussed above, robustness can be taken into account [10, 11]:

- Replacing f by a measure of both its performance and expectation in the vicinity of the solution considered (**expectation measure**).
- Considering an additional criterion for each of the M objective functions, (f_{M+M}), measuring the variation of the original objective function around the vicinity of the design point considered (for example, variance) – **variance measure**.

The adequacy of expectation measures should be evaluated considering features such as straightforward application to problems where the shape of the objective function is not known *a priori*, capacity of defining robustness regardless of that shape, independence of algorithm parameters and efficiency. In addition, when assessing variance measures, it is also important to verify whether there is a clear definition of the function maxima in the Fitness *versus* Robustness Pareto representation.

2.1 Expectation Measures

Four different expectation measures to calculate $F(x_i)$ were explored. In equation 2 $F(x_i)$ is calculated by dividing the original fitness by the absolute value of an estimated derivative, as this reduces the magnitude of the fitness function where the derivative has high values.

$$F(x_i) = \frac{f(x_i)}{\left|\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}\right|} \tag{2}$$

Instead, $F(x_i)$ can be calculated as the average of the fitness of the points located at a distance lower than a prescribed maximum value (d_{max}) – equation 3.

$$F(x_i) = \frac{\sum_{j=0}^{N'} f(x_j)}{N'} \tag{3}$$

were N' is the number of individuals of the population whose $d_{i,j} < d_{max}$, where $d_{i,j}$ is the Euclidian distance between points i and j. Tsutsui and Ghosh [12] and Wiesman *et al.* [14] computed $F(x_i)$ using equation 4:

$$F(x_i) = f(x_i)\left[\Phi\left(\frac{x+p}{g}\right) - \Phi\left(\frac{x-p}{g}\right)\right] \tag{4}$$

where p is the width of the peak under consideration, g is the size of the Gaussian noise added to the system and $\Phi(x_i)$ is the distribution of the standard normal distribution.

Finally, we propose (in equation 5) that fitness is estimated from the inverse of the average value of the difference between the normalized fitness on point i, $\tilde{f}(x_i)$, and that on its neighbors (j), (which varies in the interval [0;1]), multiplied by the original fitness for normalization purposes. This equation measures the slop around the point where the robustness was to be determined, without the need to use additional points besides the population size of the EA.

$$F(x_i) = \left(1 - \frac{\sum_{j=0}^{N'} \left|\tilde{f}(x_j) - \tilde{f}(x_i)\right|}{N'}\right) f(x_i) \tag{5}$$

2.2 Variance Measures

In this case the aim will be to optimize simultaneously the original fitness function and a robustness measure. Therefore, the performance of this method can be assessed using the Pareto frontier that can be established between the original fitness function and the robustness measure.

A variance robustness measure is proposed here (equation 11), where the robustness of individual *i* is defined as the average value of the ratio of the difference between the normalized fitness of individual *i* and that of the individuals located inside a vicinity lower than d_{max} (*j*), to the distance separating them:

$$f_i^R = \frac{1}{N'}\sum_{j=0}^{N}\left|\frac{\tilde{f}(x_j)-\tilde{f}(x_i)}{x_j - x_i}\right|, \qquad d_{i,j} < d_{\max} \tag{11}$$

This simple method can be applied to problems where the shape of the objective function is unknown and is independent of the algorithm parameters.

3 Extending Robustness Measures to Multiple Objectives

The main objective of this work is to apply a robustness analysis to multiobjective problems. In MOEA, it is possible to combine the individual expectation measures, or variance measures, with the initial objective functions. In the case of expectation measures, the number of criteria remains the same as in the original problem, but the decision maker ignores the trade-off between the criteria and the corresponding robustness. Conversely, variance measures must be combined with the original objective functions (*i.e.*, f_i+f_i^R), which duplicates the number of criteria and, consequently, may require significant computational resources for a large number of criteria, *M*. Instead, one may incorporate only the robustness of those design variables that are considered as particularly sensitive, or replace the robustness of every criterion by a single global measure. For instance, the robustness of individual *i* can be simply calculated as the robustness average obtained for each criterion *j*:

$$f_i^{R1} = \frac{1}{M}\sum_{m=1}^{M} f_{i,m}^R \tag{12}$$

The global robustness could also be defined as the maximum of the robustness measures calculated for individual *i* in each criterion *m*, *i.e.*:

$$f_i^{R2} = \max_{m=1,\dots,M} f_{i,m}^R \tag{13}$$

Therefore, it makes sense to study the applicability of equations 11-13 (original fitness with individual variance measures (f_i+f^R_i), original fitness with global variance (f_i+f^{R1}) and original fitness with a different global variance measure (f_i+f^{R2})) and compare the results with those obtained using the original fitness functions (f_i) and the expectation measures calculated with equation 5 (F_i). These robustness measures are used together with a MOEA named Reduced Pareto Set Genetic Algorithm (RPSGA), which is an algorithm that provided good results on some difficult test problems as described in detail elsewhere [15-17].

4 Results and Discussion

4.1 Test Problems

The five test problems represented by equations 6 to10 (test problems 1 to 5, respectively) will be used to assess the expectation measures (all problems are to be maximized, except Problem 4 that is to be minimized). Problem 1, equation 6 (x ranges in the interval [0;1]), defined by Tsutsui [12], has 4 sharp (progressively smaller) and 1 broad peak, whose location depends on the values of a and b (for 0.4 and 0.6, respectively, the latter is located at x = 0.486, with a fitness of 0.715). Thus, in this problem, the greatest performance occurs at x = 0.1, but the most robust peak exists at x = 0.486. In the case of problem 2, equation 7 (x ranges in the interval]0;1]), one must cope with 5 peaks with the same peak value, but with different slopes. Test problem 3, equation 8 (x ranges in the interval]0;1]) combines the characteristics of the two previous, as it contains 5 uneven decreasing maxima. The function to be minimized in problem 4, equation 9 (x ranges in the interval [1;10]) oscillates between positive and negative values and has also five peaks. Finally, problem 5, equation 10 (x ranges in the interval [-6;6]) has one maximum in each quadrant, the parameters c and d determining both their location and slope in the vicinity (c = d = 20 and c = 11, d = 7 were used).

$$f(x) = \begin{cases} e^{-2\left(\frac{x-0.1}{0.8}\right)^2 \ln 2} \left|\sin(5\,\pi\,x)\right|^{0.5}, & a < x \le b \\ e^{-2\left(\frac{x-0.1}{0.8}\right)^2 \ln 2} \sin^6(5\,\pi\,x), & \text{otherwise} \end{cases} \tag{6}$$

$$f(x) = \sin^6\left(5\pi\left(x^{0.75} - 0.05\right)\right) \tag{7}$$

$$f(x) = e^{-2\left(\frac{x-0.1}{0.8}\right)^2 \ln 2} \sin^6\left(5\pi\left(x^{0.75} - 0.05\right)\right) \tag{8}$$

$$f(x) = 2\sin\left(10\,x\,e^{(-0.08\,x)}\right)e^{-0.25\,x} \tag{9}$$

$$f(x) = \frac{2186 - \left(x_1^2 + x_2 - c\right)^2 - \left(x_1 + x_2^2 - d\right)^2}{2186} \qquad (10)$$

4.2 Expectation Results

Since the location of the fittest solution will depend on the required degree of performance and robustness, equations 2 to 5 should produce F(x) curves able to differentiate the various individual peaks, so that the optimization algorithm can converge to them. Table 1 compares the main performance characteristics of the expectation measures proposed by equations 2 to 5, when applied to test problems 1 to 5. For that purpose the F(x) was calculated for the entire search space. The performance characteristics considered are: i) a clear definition of the robustness along the entire search space; ii) independence of the robustness measure from algorithm parameters; iii) efficiency in founding the existent robust solutions in practice. All approaches can be applied to problems where it is not possible to calculate derivatives. Equation 3 does not provide a smooth definition of robustness (expectation measure) along the search space. Equations 2 and 5 are independent of the algorithm parameters. Finally, equation 5 seems to be the most efficient, since it is able to differentiate the most robust solutions without affecting significantly the expectation measure of the other possible solutions.

Table 1. Comparative performance of the expectation measure methods used.

Equation	Clear definition	Independence of parameters	Efficiency
2	Yes	Yes	Small
3	No	No	Medium
4	Yes	No	Small
5	Yes	Yes	High

4.3 Variance Results

Four different test problems (problems 6 to 9) will be used to study the performance of the expectation measures. Test problems 6 and 8 build on problem 1 (equation 6) and involve two and three objectives, respectively. Test problem 6 considers $f_1(x)$ with $a = 0.4$ and $b = 0.6$ and $f_2(x)$ with $a = 0.2$ and $b = 0.4$ and test problem 8 considers $f_1(x)$ with $a = 0.4$ and $b = 0.6$, $f_2(x)$ with $a = 0.2$ and $b = 0.4$ and $f_3(x)$ with $a = 0.6$ and $b = 0.8$. In the case of problem 6, the most robust solution (*i.e.* the broad peak) of the second function is located at the same position of the second sharp peak of the first function. The most robust solution of $f_3(x)$ in problem 8 is located at the same position as the 4th sharp peak of function $f_1(x)$. In problems 7 and 9, which build on problem 2 (equation 7), the uneven peaks have a maximum fitness of one at different locations. Test problem 7 considers $f_1(x)$ with $e = 0.75$ and $f_2(x)$ with $e = 0.60$. Test problem 9 considers $f_1(x)$ with $e = 0.75$, $f_2(x)$ with $e = 0.60$ and $f_3(x)$ with $e = 0.45$.

Five hundred possible solutions were obtained randomly in the search space (for each problem), in order to determine the criteria values, the individual and the competing global robustness measures, i.e., F_1 and F_2 were calculated without optimization. As an example, figure 1 presents the trade-off between the objective functions (f_1 and f_2) and between the resulting expectation measures (F_1 and F_2) for problem 6. In the first case (Figure 1A), only point (1.0, 1.0) is non-dominated and corresponds to the first peak, at $x = 0.1$. The other two points signaled by the arrows identify the robust peaks of each function, at $x = 0.3$ and $x = 0.499$, respectively. Thus, if the original objective functions are used for optimization, the solution will be the first peak ($x = 0.1$). In Figure 1B the arrows point at the first three peaks, which are all non-dominated and, consequently, will be the result of the application of a multi-objective optimization algorithm using the expectation measures as criteria.

Table 2 contains a global comparison of the performance of the five criteria combinations used to solve problems 6 to 9. Three performance measures are used, namely the total number of peaks obtained, the efficiency and the precision of the results. The first measure computes the percentage of peaks obtained in each problem (the total being 7, 10, 8 and 15, for problems 6 to 9, respectively). Efficiency is defined as the capacity of the optimization algorithm to find the fittest and more robust solutions, i.e., the ration between the peaks found and the total number of these type of peaks (there are 3, 2, 4 and 3 of these peaks in problems 6 to 9, respectively). Finally, the precision of the results is estimated from the difference between the fitness of the solutions obtained and the effective best fitness. On average, combination 3 performs better than the remaining in all measures. Therefore, the quantitative information provided by this method and made available to the decision maker should make easier the task of selecting a single point as the practical solution to the optimization problem under study.

5 Conclusions

A measure of the robustness of the solutions was introduced in a multi-objective evolutionary algorithm, in order to make possible the simultaneous optimization of fitness and robustness. Two types of robustness measures are defined, expectation and variance. Expectation measures quantify both fitness and robustness of the solution considered, whereas variance simply determines the variation of the original function in the vicinity of the solution.

New criteria quantifying robustness were proposed, either by defining a robustness measure for each original criterion, or by defining a single robustness measure aggregating the robustness of each original criterion. The most performing combination uses the original criteria combined with an equal number of new criteria quantifying robustness through the variance measure proposed in this work.

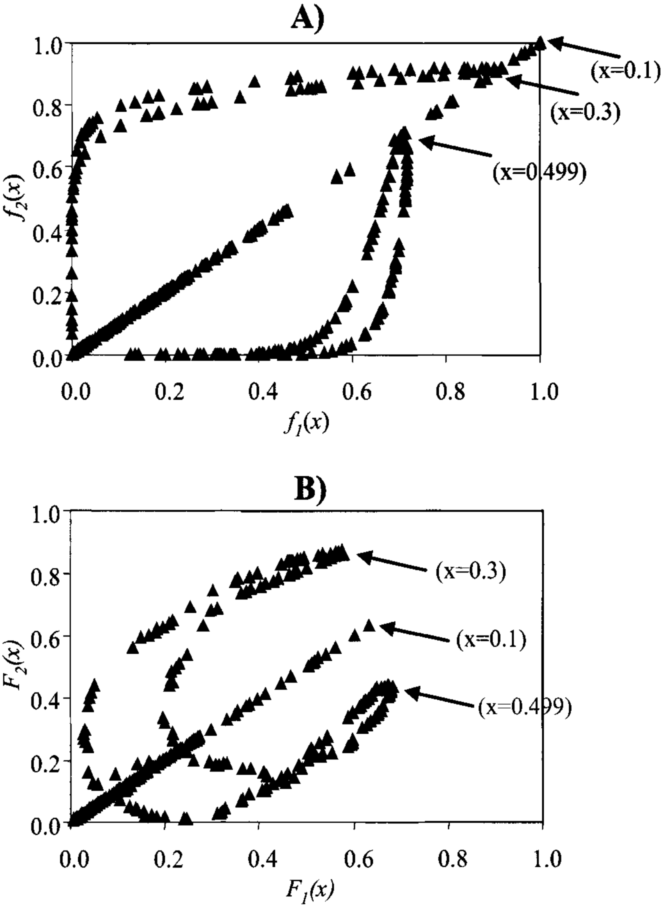


Fig. 1. Test problem 6: A) original fitness functions; B) expectation measures.

Table 2. Comparative performance of combinations 1 to 5.

Comb.	Problem	No. of peaks (%)		Efficiency (%)		Precision (%)		Global (%)
1	6	14	21	33	52	20	26	33
	7	10		50		19		
	8	13		25		20		
	9	47		100		47		
2	6	14	18	33	40	19	21	26
	7	10		0		12		
	8	13		25		20		
	9	33		100		33		
3	6	57	76	67	92	79	81	83
	7	100		100		98		
	8	100		100		99		
	9	47		100		46		
4	6	43	59	100	88	59	64	70
	7	50		50		50		
	8	75		100		80		
	9	67		100		66		
5	6	29	54	67	85	38	58	66
	7	60		100		60		
	8	75		75		79		
	9	53		100		53		

Acknowledgements

This work was supported by the Portuguese Fundação para a Ciência e Tecnologia under grant POCTI/EME/48448/2002.

References

1. Schafer, J.D.: Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms, Ph. D. Thesis, Nashville, TN, Vanderbilt University (1984)
2. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, Proc. Fifth Int. Conf. on Genetic Algorithms, Morgan Kauffman (1993) 416–423
3. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation, **2** (1995) 221–248
4. Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization, Proc. First IEEE Conf. on Evolutionary Computation (1994) 82–87

5. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGAII, IEEE Transactions on Evolutionary Computation, **6** (2002) 182–197
6. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK report no. 103, Swiss Federal Institute of Technology, Zurich, Switzerland (2001)
7. Knowles, J.D., Corne, D.W.: Approximating the Non-dominated Front using the Pareto Archived Evolutionary Strategy, Evolutionary Computation Journal, **8** (2000) 149–172
8. Deb, K.: Multi-Objective Optimisation using Evolutionary Algorithms, Wiley (2001)
9. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer (2002)
10. Ray, T.: Constrained Robust Optimal Design using a Multiobjective Evolutionary Algorithm, Proceedings of the 2002 Congress on Evolutionary Computation – CEC'02 (2002) 419–424
11. Jin, Y., Sendhoff, B.: Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach, Lecture Notes in Computer Science – EMO'2003, Faro, Portugal, **2632** (2003), pp. 237–251.
12. Tsutsui, S., Ghosh, A.: Genetic Algorithms with a Robust Solution Scheme, IEEE Transations on Evolutionary Computation, **1** (1997) 201–208
13. Chen, W., Sahai, A., Messac, A, Sundararaj, G.J.: Physical Programming for Robust Design, 40th Structures, Structural Dynamics and Materials Conference, St. Louis, USA (1999)
14. Wiesmann, D., Hammel, U., Bäck, T.: Robust Design of Multilayer Optical Coatings by Means of Evolutionary Algorithms, IEEE Transations on Evolutionary Computation, **2** (1998) 162–167
15. Gaspar-Cunha, A., Covas, J.A. - RPSGAe - A Multiobjective Genetic Algorithm with Elitism: Application to Polymer Extrusion, in Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems, Gandibleux, X.; Sevaux, M.; Sörensen, K.; T'kindt, V. (Eds.), Springer (2004)
16. Gaspar-Cunha, A., Oliveira, P., Covas, J.A.: Use of Genetic Algorithms in Multicriteria Optimization to Solve Industrial Problems, Seventh Int. Conf. on Genetic Algorithms, Michigan, USA (1997)
17. Gaspar-Cunha, A.: Modelling and Optimisation of Single Screw Extrusion, Ph. D. Thesis, University of Minho, Guimarães, Portugal (2000)

Genetic Programming, Probabilistic Incremental Program Evolution, and Scalability

Radovan Ondas[1], Martin Pelikan[2], and Kumara Sastry[3]

[1] Dept. of Math. and Computer Science, University of Missouri at St. Louis,
CCB 331, 8001 Natural Bridge Rd., St. Louis, MO 63121, USA
ondasr@umsl.edu

[2] Dept. of Math. and Computer Science, University of Missouri at St. Louis,
CCB 320, 8001 Natural Bridge Rd., St. Louis, M O 63121, USA
pelikan@cs.umsl.edu

[3] Dept. of General Engineering, University of Illinois at Urbana-Champaign, 117
TB, 104 S. Mathews Ave., Urbana, IL 61801, USA
ksastry@uiuc.edu

Summary. This paper discusses scalability of standard genetic programming (GP) and the probabilistic incremental program evolution (PIPE). To investigate the need for both effective mixing and linkage learning, two test problems are considered: ORDER problem, which is rather easy for any recombination-based GP, and TRAP or the deceptive trap problem, which requires the algorithm to learn interactions among subsets of terminals. The scalability results show that both GP and PIPE scale up polynomially with problem size on the simple ORDER problem, but they both scale up exponentially on the deceptive problem. This indicates that while standard recombination is sufficient when no interactions need to be considered, for some problems linkage learning is necessary. These results are in agreement with the lessons learned in the domain of binary-string genetic algorithms (GAs). Furthermore, the paper investigates the effects of introducing unnecessary and irrelevant primitives on the performance of GP and PIPE.

Keywords: Genetic Programming, PIPE, scalability, order problem, trap problem

1 Introduction

To solve large and complex problems, scalability is among the primary concerns of an optimization practitioner. However, only few studies [14,15] exist that study scalability in genetic programming (GP) [7]. The same holds for simple approaches to using probabilistic recombination in GP within the estimation of distribution algorithm (EDA) framework [8, 9, 11], such as the probabilistic incremental program evolution (PIPE) [13].

The purpose of this paper is to study the scalability of standard GP and PIPE on two decomposable GP problems: ORDER and TRAP [14]. The two algorithms perform as expected and they solve ORDER scalably while failing to scale up on TRAP. Additionally, the paper studies the effects of introducing unnecessary and irrelevant primitives. Both GP and PIPE are shown to deal with these two sources of difficulty well. The results presented in this paper confirm that binary-string genetic algorithms (GAs) have a lot in common with GP and PIPE, and thus the lessons learned in the design, study, and application of standard GAs and their extensions should carry over to GP as argued for example in [5, 14, 15].

The paper starts by describing the algorithms investigated in this paper: GP and PIPE. Section 3 explains test problems. Section 4 provides and discusses experimental results. Section 5 summarizes the paper. Finally, Section 6 concludes the paper and presents important topics for future work.

2 Methods

Both GP and PIPE work with programs encoded as labeled-tree structures and both can be applied to the same class of problems. This section describes GP and PIPE. Both GP and PIPE were implemented using the `lilgp` library developed by GARAGe at the Michigan State University.

2.1 Genetic Programming

Genetic programming (GP) [7] is a genetic algorithm (GA) [3] that evolves programs instead of fixed-length strings. Programs are represented by trees where nodes represent functions and leaves represent variables and constants.

GP starts with a population of random candidate programs. Each program is evaluated on a given task and its fitness is then computed based on the evaluation. A population of promising programs is then selected using one of the standard GA selection operators, such as tournament or truncation selection. Some of the selected programs can be directly copied into the new population, the remaining ones are copied after applying variation operators, such as crossover and mutation. Crossover usually proceeds by exchanging randomly selected subtrees between two programs, whereas mutation usually replaces a randomly selected subtree of a program by a randomly generated one. This process is repeated until termination criteria are met.

Since standard GP variation operators proceed without considering interactions between different components of selected programs, they are likely to experience difficulties with solving problems where different program components interact strongly. However, problems that can be decomposed into subproblems of order one should be easy for any standard GP based on recombination. This intuition is verified with experiments in Section 4.

2.2 PIPE

In the probabilistic incremental program evolution (PIPE) algorithm [13] computer programs or mathematical expressions are evolved like in GP [7]. The initial population is also generated at random. All programs in the population are then evaluated and selection is applied to select the population of promising programs. Instead of applying crossover and mutation to a part of the selected population to generate new programs, PIPE builds a probabilistic model of the selected programs in the form of a tree. This probabilistic model is then sampled to generate new candidate programs that form the new population. The process is repeated until the termination criteria are met.

The probabilistic model in PIPE is a tree with the structure corresponding to the structure of candidate programs. Since different programs may be of different structure and size, the population is first parsed to find the smallest tree that covers every structure in the selected population. If there are functions of different arities, the number of children of each node in the probabilistic model is equal to the maximum arity of a function in this node and for a function of smaller arity, the first children are interpreted as arguments of this function. Each node of a program in the selected population then directly corresponds to one node in the model, whereas the children of each internal node represent arguments of the function in this node.

PIPE then parses the selected population and computes the probabilities of different functions and terminals in each node of the probabilistic model. The nodes of the probabilistic model thus consist of tables of probabilities, and there is one probability for each function or terminal in each node.

Sampling of the probabilistic model starts in the root of the probabilistic model. The same recursive procedure is used to generate each node. First, a function or terminal is generated in the current node based on the distribution encoded by the table of probabilities in this node. If the function requires several arguments, a necessary number of children are generated recursively. The recursive generation terminates in a node whenever a terminal is generated in this node and thus no children have to be generated. Since the probabilistic model is built from an actual population of programs, the sampling will never cross the boundaries of the model.

3 Test Problems

In order to test scalability, we need a class of problems where size can be modified while the inherent problem difficulty does not grow prohibitively fast. In fixed-length string GAs, decomposable problems of bounded difficulty [4] can be used as a challenging but solvable class of problems. Two types of decomposable problems for fixed-length string GAs are common: onemax and concatenated traps.

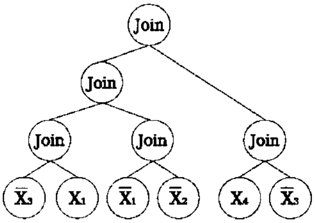


Fig. 1. This figure shows an example of a candidate solution for a 4-primitive ORDER problem. The sequence of leaves visited during the inorder parse is $\{\overline{X}_3, X_1, \overline{X}_1, \overline{X}_2, X_4, \overline{X}_3\}$, the expression of this sequence is $\{X_1, \overline{X}_2, \overline{X}_3, X_4\}$, and the fitness of this solution is thus 2.

Similar problems to onemax and concatenated traps were also created for GP where candidate solutions are represented by program trees [5, 14]. Two classes of problems from [14] are considered: (1) ORDER (onemax-like, GP-easy problem), and (2) TRAP (deceptive-trap-like, GP-difficult problem).

ORDER should be easy for any recombination-based GP. However, since standard variation operators do not consider interactions between different program components, TRAP can be expected to lead to exponential scalability of both standard GP and PIPE. The problems are described next.

3.1 Problem 1: Order

The primitive set of an l-primitive ORDER problem [14] consist of a binary function JOIN and l complementary pairs of terminals X_i and $\overline{X}_i$ for $i \in \{1, 2, \ldots, l\}$. A candidate solution of the ORDER problem is a binary tree with JOIN in all internal nodes and either X_i's or $\overline{X}_i$'s at its leaves. The candidate solution's output is determined by parsing the program tree inorder (from left to right). The program expresses X_i if, during the inorder parse, X_i is encountered before its complement $\overline{X}_i$ and neither X_i nor its complement are encountered earlier. For all $i \in \{1, 2, \ldots, l\}$, if X_i is unexpressed, $\overline{X}_i$ is expressed instead. One terminal is thus expressed from each pair X_i and $\overline{X}_i$.

The fitness is defined as the number of positive terminals X_i that were expressed during the inorder parse. The fitness can thus be seen as onemax applied to a binary string where the ith bit is 1 if X_i was expressed, whereas the ith bit is 0 if $\overline{X}_i$ was expressed (see Figure 1).

3.2 Problem 2: Deceptive Trap

In standard GAs, deceptive functions [2, 4] are designed to thwart the very mechanism of selectorecombinative search by punishing any localized hill-climbing and requiring the mixing of whole building blocks at or above the order of deception. Using such adversarially designed functions is a stiff test of algorithm performance. The idea is that if an algorithm can beat adversarially designed test functions, it can solve other problems that are equally hard or easier than the adversary.

TRAP is designed to test the same mechanisms in GP. Fitness is computed so that if interactions between different components of the program are not considered, optimization may be misled away from the global optimum. Similarly as with standard GAs on deceptive functions, standard GP is expected to fail in solving TRAP scalably, indicating the need for linkage learning in GP.

Programs in TRAP also consist of one binary function JOIN and l pairs of complementary primitives X_i and $\overline{X}_i$. The expression mechanism of the program for TRAP is identical to that of ORDER. The difference is in the fitness evaluation procedure.

In TRAP, the expressed set of primitives is first mapped to an l-bit binary string. The ith bit of the string is 1 if and only if X_i was expressed; otherwise, the ith bit of the string is 0. The resulting binary string is then partitioned into groups of k bits each (the partitioning is fixed during the entire run) and a trap function is applied to each group:

$$trap_k(u) = \begin{cases} 1 & \text{if } u = k \\ (1-\delta)\left(1 - \frac{u}{k-1}\right) & \text{otherwise} \end{cases} \tag{1}$$

where u is the number of ones in the input string of k bits. The difficulty of the trap function can be adjusted by modifying the values k and δ; in this paper we use traps with $k = 3$ and $\delta = 1$. TRAP fitness function is then computed by adding the contributions of all groups of k bits together.

An important feature of additively separable trap functions is that if looking at the performance of any strict subset of k bits corresponding to one trap, it seems to be better to propagate 0s than 1s (here $\overline{X}_i$ would be propagated at the expense of X_i); however, the optimum is in the string of all 1s (X_i is expressed for any i). As shown in [14], if interactions between different components of the program are not considered, it can be expected that GP will scale up poorly on this problem.

3.3 Other Primitives

The purpose of additional tests was to determine how GP and PIPE respond to more complex interactions and unnecessary program primitives. Two additional types of primitives were added into ORDER problem: (1) NEG_JOIN function and (2) JUNK terminals.

NEG_JOIN affects all its descendant terminals by expressing each primitive X_i as its negation $\overline{X}_i$; analogically, all descendants $\overline{X}_i$ are expressed as X_i. If a terminal has more NEG_JOIN ancestors, only one of them is considered and the terminal is negated only once. NEG_JOIN is unnecessary for solving ORDER. Furthermore, NEG_JOIN introduces interactions into ORDER because the best value in each leaf depends on its ancestors. Nonetheless, these interactions are relatively simple as many leaves are expected to contain NEG_JOIN on the path to the root.

JUNK terminals represent unnecessary primitives that are irrelevant for the particular problem. In biological terms, JUNK terminals correspond to junk code in DNA. During the expression phase, JUNK terminals are simply ignored and thus they do not influence the overall fitness at all. The influence of JUNK terminals can be tuned by changing the number of unique JUNK terminals.

4 Experiments

This section compares the performance of GP and PIPE on three variants of ORDER and one variant of TRAP.

4.1 Description of Experiments

The scalability of GP and PIPE was tested on four classes of problems: (1) ORDER (no JUNK or NEG_JOIN), (2) TRAP (no JUNK or NEG_JOIN), (3) ORDER with NEG_JOIN, and (4) ORDER with JUNK terminals where the number of unique JUNK terminals is set to $l/5$.

The scalability experiments were performed by testing both algorithms on problem instances with an increasing number l of primitives. Additionally, the effects of increasing the number of unnecessary primitives on the performance of GP and PIPE were studied by testing GP and PIPE on a 20-primitive ORDER with an increasing number of JUNK terminals (from 5 to 40).

Binary tournament selection was used in both GP and PIPE. The probability of crossover in GP was set to 1.0. To focus on the effects of recombination, no mutation was used. The initial population in both methods was generated using the standard half-and-half method. Maximum tree depth was set to be one more than the depth of the minimum tree to store the global optimum. The population size that is within 10% of the minimum population size required to solve 30 independent runs was used. The population size was determined using a bisection method. The runs were terminated when the algorithms found the global optimum or when the number of generations was too large for the particular problem (based on experience).

4.2 Results

Figure 2 shows the scalability of GP and PIPE on ORDER without NEG_JOIN or JUNK terminals. Problem instances of different size were examined; more specifically, l = 5, 10, 20, 40, 60, 80, and 100. The figure shows the average number of function evaluations of 30 successful runs with respect to the problem size l. The results indicate that PIPE is slightly more efficient than GP but both GP and PIPE scale up with a low-order polynomial. These results are in agreement with the behavior observed in binary-string GAs on the

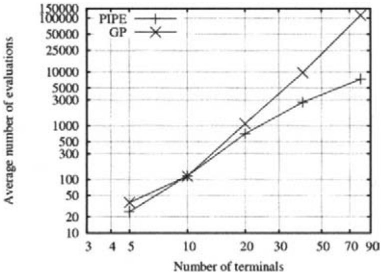


Fig. 2. Scalability of GP and PIPE on ORDER.

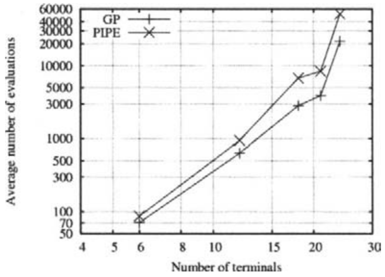


Fig. 3. Scalability of GP and PIPE on TRAP.

simple onemax problem. On onemax, both simple GA and UMDA find the optimum in low-order polynomial time [4,6,10,12]; however, UMDA performs slightly better [12] because it uses a more effective recombination for this type of problems.

Figure 3 compares the scalability of GP and PIPE on TRAP without NEG_JOIN or JUNK terminals. Problem instances of different size were examined; more specifically, l = 6, 12, 18, 21, 24, and 33. On TRAP, GP performs slightly better than PIPE. This can be explained by its weaker recombination operator because here recombination causes disruption of important partial solutions [16] as can be hypothesized based on the performance of standard GAs on similar problems. Nonetheless, both GP and PIPE scale up poorly and they indicate an exponential growth of the number of function evaluations with problem size.

Figure 4 compares the scalability of GP and PIPE on ORDER with NEG_JOIN. Problem instances of different size were examined; more specifically, l = 5, 10, 20, 40, 60, 80, and 100. Both GP and PIPE perform similarly as on basic ORDER without NEG_JOIN, but there is a slight decrease in their performance because of the interactions introduced by NEG_JOIN.

Figure 5 compares the scalability of GP and PIPE on ORDER with $l/5$ unique JUNK terminals. For example, a problem instance with $l = 20$ positive terminals contains 4 unique JUNK terminals. Both GP and PIPE seem to be capable of dealing with these irrelevant terminals and achieve performance comparable to that on basic ORDER.

The last two sets of experiments are similar in that they show how the performance of GP and PIPE changes when adding irrelevant terminals into the representation. ORDER with $l = 20$ terminals is used with the number of JUNK terminals ranging from 5 to 40 (5, 10, 15, 20, and 40). The experiments differ in the bound on the maximum tree depth. Figure 6 shows the results with the depth limited to at most 6 (so there are at most 7 levels including the root). Figure 7 shows the results with the depth limited to at most 7 (so there are at most 8 levels including the root). The problem with the smaller maximum depth is more difficult for both GP and PIPE because JUNK terminals obstruct the creation of an optimal solution that is only slightly

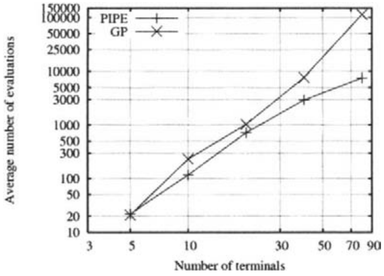


Fig. 4. Scalability of GP and PIPE on ORDER with NEG_JOIN.

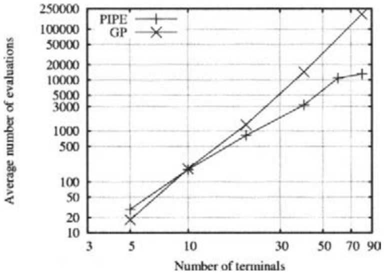


Fig. 5. Scalability of GP and PIPE on ORDER with $l/5$ copies of JUNK terminals.

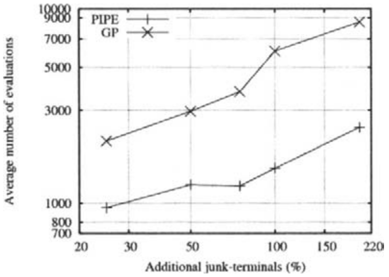


Fig. 6. Scalability of GP and PIPE on ORDER with an increasing number of JUNK terminals (from 5 to 40).

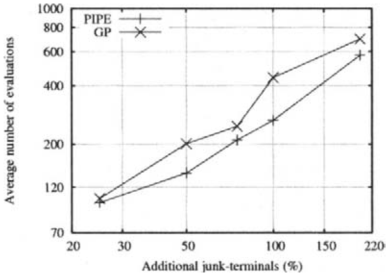


Fig. 7. Scalability of GP and PIPE on ORDER with an increasing number of JUNK terminals (from 5 to 40).

larger than the maximum allowed tree. PIPE deals better with this "lack of space" than GP does. However, in both cases, the number of evaluations still appears to grow with a low-order polynomial as irrelevant terminals are added.

5 Summary

This paper focused on the scalability of two GP algorithms: standard GP and PIPE. Two basic test functions were used: ORDER and TRAP. Both functions were defined using one binary function JOIN and l complementary terminal pairs. ORDER can be solved without considering interactions between different program components, whereas TRAP introduces strong interactions, which make this function difficult for both standard crossover and mutation of GP, as well as the probabilistic recombination of PIPE.

The scalability of GP and PIPE was tested on basic ORDER and TRAP. Additionally, ORDER was extended by adding either of the following two primitives: (1) a binary function NEG_JOIN and (2) JUNK (or irrelevant) terminals. Thus, there were 4 problem types examined.

On all four problem types, the scalability of GP and PIPE was first tested by applying these algorithms to problem instances of different size (number l

of positive terminals). Then, the sensitivity of GP and PIPE to the proportion of irrelevant terminals to the relevant ones was examined.

6 Conclusions and Future Work

The results presented in this paper indicate that the behavior of different variants of GP can be expected to be similar to that of standard binary-string GAs. There are two important consequences of this fact. First, as it was indicated in [14], to solve some classes of GP problems scalably, linkage learning may have to be incorporated into GP in order to identify and exploit interactions between different program components. Second, the lessons learned in the design and application of binary-string GAs should carry over to GP as argued for example in [5,15]; the first steps along this direction are represented by the decision-making model of the population sizing in GP [15], which was based on the decision-making population-sizing model for standard GAs [4,6].

The results also indicate that if the recombination operator captures interactions in the problem properly, increasing the mixing effects of recombination leads to better performance. That is why PIPE outperformed standard GP on problems where program components could be treated independently. This fact together with the need for linkage learning should encourage the application of probabilistic recombination operators of estimation of distribution algorithms (EDAs) [8,9,11] to the domain of GP.

Finally, the results show that both GP and PIPE can deal with irrelevant terminals and unnecessary functions relatively well and their performance gets only slightly worse when adding these primitives.

Future work should study the scalability of GP, PIPE, and other similar approaches on the problems presented in this paper and other problems where problem size can be modified without affecting the inherent problem difficulty. The efforts to introducing linkage learning into GP (for example [1, 14]) should continue to succeed in the design of robust and scalable GP algorithms applicable to broad classes of difficult decomposable problems.

Acknowledgments

This work was partially supported by the Research Award and the Research Board at the University of Missouri. This work was also sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-03-1-0129, the National Science Foundation under ITR grant DMR-99-76550 (at Materials Computation Center), and ITR grant DMR-0121695 (at CPSD), and the Dept. of Energy under grant DEFG02-91ER45439 (at Fredrick Seitz MRL). The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

References

1. P. A. N. Bosman and E. D. de Jong. Learning probabilistic tree grammars for genetic programming. pages 190–199, 2004.
2. K. Deb and D. E. Goldberg. Analyzing deception in trap functions. *Foundations of Genetic Algorithms*, 2:9–108, 1993.
3. D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, MA, 1989.
4. D. E. Goldberg. *The design of innovation: Lessons from and for competent genetic algorithms*, volume 7 of *Genetic Algorithms and Evolutionary Computation.* Kluwer Academic Publishers, 2002.
5. D. E. Goldberg and U.-M. O'Reilly. Where does the good stuff go, and why? how contextual semantics influence program structure in simple genetic programming. *Proceedings of the First European Workshop on Genetic Programming*, 1391:16–36, 14–15 Apr. 1998.
6. G. R. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Proceedings of the International Conference on Evolutionary Computation (ICEC-97)*, pages 7–12, 1997. Also IlliGAL Report No. 96004.
7. J. R. Koza. *Genetic programming: On the programming of computers by means of natural selection.* MA: The MIT Press, Cambridge, 1992.
8. P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Kluwer, Boston, MA, 2002.
9. H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature*, pages 178–187, 1996.
10. H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.
11. M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002. Also IlliGAL Report No. 99018.
12. M. Pelikan, K. Sastry, and D. E. Goldberg. Scalability of the Bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258, 2002. Also IlliGAL Report No. 2001029.
13. R. Salustowicz and J. Schmidhuber. Probabilistic incremental program evolution. *Evolutionary Computation*, 5(2):123–141, 1997.
14. K. Sastry and D. E. Goldberg. Probabilistic model building and competent genetic programming. April 2003.
15. K. Sastry, U.-M. O'Reilly, and D. E. Goldberg. Convergence-time models for the simple genetic algorithm with finite population. IlliGAL Report No. 2001028, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2001.
16. D. Thierens. *Analysis and design of genetic algorithms.* PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1995.

Adaptive Parameter Control of Evolutionary Algorithms Under Time Constraints

Sandip Aine, Rajeev Kumar*, and P. P. Chakrabarti

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, WB 721302, India
{sandip, rkumar, ppchak}@cse.iitkgp.ernet.in

Summary. Parameter control of evolutionary algorithm (EA) poses special challenges as EA uses a population. A widely practiced approach to identify a good set of parameters for a particular class of problem is through experimentations. Ideally, the parameter selection should depend on the resource availability, and thus, a rigid choice may not be suitable. In this paper, we attempt to address the problem of parameter control of EA under given time constraints. We propose an automated framework for parameter selection, which can adapt according to the constraints specified. We propose both static and dynamic selection strategies based on a probabilistic profiling method. Experiments performed with traveling salesman problem (TSP) show that an adaptive parameter control mechanism can yield better results than a static selection.

1 Introduction

The rate of convergence of evolutionary algorithms is strongly influenced by the choice of certain parameters, such as population size and mutation and cross-over probabilities, collectively termed as control parameters of the algorithm. In the past, a considerable amount of effort has been put to device strategies for choosing a good set of parameters to improve the performance of evolutionary algorithms [4]. However, it has been shown that it is not possible to find a set of control parameter values which is optimal for all problem domains [10], and thus, this prompted researchers to concentrate on particular classes of problems. Generally, researchers experiment with a set of problems from a particular domain tuning the control parameters on the basis of such experimentation. Another approach is to set the parameters dynamically using some adaptive equations [8, 11] or prior knowledge [3]. Eiben et al. [4]

* Present Address: Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur, UP 208 016, India.

provides a good survey of the different adaptive strategies proposed in literature. He argued that the evolutionary algorithms are implicitly dynamic in nature. Therefore, the use of constant parameters is in contrast to the general evolutionary spirit and an adaptive strategy is preferable.

Whenever a parameter selection or adaptation strategy is proposed, the standard methodology is to measure the performance of an algorithm under several parameter choices, and to report the best possible configuration. Generally, the objective is to maximize the solution quality within certain time/resource limits or to minimize the time/resource when a given quality target is specified. In real-life situations, the limits on resources available may vary for different environments. Thus, a selection of parameters reported, is only useful when the experimental assumptions (reported by the researchers) and the real constraints (for the users) match. For example, if a parameter set choice C is reported to perform best for a given time limit T_1 (for a given class of problems), there is no guarantee that C will still be the best choice if the time given is T_2. Similarly, an adaptation strategy which gives good performance under a particular constraint may not be suitable for others. In this work, we attempt to formulate a meta-level reasoning framework for parameter selection that can possibly adapt according to the given time constraints. We use a profile based methodology to select the right set of parameters (cross-over and mutation probabilities) when a given constraint is specified. We discuss both static (parameter tuning) and adaptive (parameter control) frameworks and compare their performances.

Anytime algorithm [2] based methods have been proposed to address the problem of parameter selection under given resources. Fukunaga [5] suggested an anytime portfolio technique and showed that a portfolio of several independent EA runs with different control parameters can outperform a set of restarts using a single best control choice. However, the methodology suggested is intrinsically a static one, no revision of choices is performed during a particular run.

In our formulation, we attempt to model steady-state genetic algorithms [9] as interruptible *anytime* algorithms and device a strategy for dynamic parameter selection based on the pre-computed profiles. The basic methodology is to interrupt a particular run at pre-defined intervals and select a new set of parameters monitoring the current state of the algorithm. We evaluate the efficacy of the methodology using a standard steady-state genetic algorithm to the traveling salesman problem (TSP) . We use integer encoding for individual chromosomes and permutation based cross-over strategies [7]. The experimental results show that a dynamic adaptive strategy can significantly outperform the static selection strategies.

The rest of the paper is organized as follows. In Section 2, we formulate the problem of parameter control under time constraints. Sections 3 and 4 describe the static and dynamic parameter control frameworks, respectively. We present the experimental results in Section 5 and conclude in Section 6.

2 Problem Formulation

The basic problem for the meta-level controller, we address in this work, is to decide on a control parameter setting which maximizes the expected quality under the given time constraints. To determine the intermediate solution quality of an individual element of a population, we use the concept of approximation ratio . For cost minimization problems, approximation ratio is given by,

$$Appx.\ Ratio = Fitness(App.\ Soln.)/Fitness(Opt.\ or\ Ref.\ Soln.) \qquad (1)$$

The approximation ratio can be used in a generic way as it does not depend on the exact cost of a solution, which can vary for different problem instances. The reference point can be obtained in various ways, e.g., using some approximate bound on optimal solution or relative values in terms of cost differences. The population quality can be expressed as some aggregate of the individual qualities. We discuss different heuristics for approximating the population quality in Section 4.

Given a run-time constraint , the meta-level controller tries to find a strategy that will maximize the expected quality. The selection strategy can either be static or adaptive. For static framework, the parameter values are decided at the start of the algorithm and the decision is not revised during runtime. The static model works well when there is little or no uncertainty about the progress of the algorithm. For algorithms where the progress is not that predictable and different parameter settings are suitable at different stages, a dynamic monitoring based strategy is preferred. In the dynamic case, the control decision is updated during runtime by monitoring the progress of the algorithm for a particular run. For multi-parameter cases, mixed strategies can also be used, which fall in between these two extremes of static and dynamic decision, i.e., some of the parameters are fixed at the start and some of them are controlled at runtime.

If there are n-controllable variables, 2^n combinations of static and dynamic decision making options are possible. We describe our framework considering two parameters – mutation and cross-over probabilities. The framework can easily be extended to include any number of variables. The possible options for strategies with two variables are:

$\mathbf{F_M F_C}$: Fixed mutation and cross-over rate (Parameter Tuning),
$\mathbf{F_M A_C}$: Fixed mutation rate and adaptive cross-over rate,
$\mathbf{A_M F_C}$: Adaptive mutation rate and fixed cross-over rate, and
$\mathbf{A_M A_C}$: Adaptive mutation and cross-over rate (Parameter Control).

We formulate the static ($F_M F_C$) and the adaptive ($A_M A_C$) strategies in this paper. We also propose the mixed strategies ($F_M A_C$, $A_M F_C$) which can be obtained as intermediate variants of the two policies described.

The control vector (V_c) chosen for the evolutionary algorithm comprises of two elements, namely, cross-over (c_p) and mutation (m_p) probabilities. where

m_p is the mutation probability and c_p is the The meta-level controller tries to assign appropriate values to the control vector (either statically or dynamically) selecting from an initial set of choices. In the next sections, we discuss the static and adaptive frameworks for parameter selection.

3 Static Selection of Control Parameters

The meta-level control framework requires a model to measure the improvement of the solution quality with time for different control parameter settings. Dean and Boddy [2] introduced the term *performance profile* of an anytime algorithm, which represents the expected solution quality as a function of the allocated time. We extend the basic concept of performance profile to get a *parameterized performance profile* of an anytime algorithm as the model for the algorithm's quality-time relation depending on the control parameter settings.

Definition 1 (PPP). *Parameterized Performance Profile of an anytime algorithm given by $P(q_i|t, v_c)$ is the probability of getting a solution with quality q_i, where t is the time allocated and v_c is the control vector choice.*

The Parameterized Performance Profile can be represented by a 3-tuple $< P, t, v_c >$ using a look-up table. All the components of PPP namely, the solution quality, the computational time and the control parameter values are discretized to have a finite number of choices. Sampling intervals for each of the parameters is guided by the sensitivity of that parameter with respect to the application.

So with a given PPP and a run-time constraint, the static meta-level controller has to select the control vector values before the start of the algorithm, to maximize the expected quality.

Strategy 1 (Static Selection ($\mathbf{F_T F_C}$)) *Given an anytime algorithm with PPP $P(q|t, v_c)$ and a run-time limit T_{max}, an optimal static selection of control vector (v_c*) is given by,*

$$(v_c*) = arg\ max\ (v_c) \sum_i P(q_i|T_{max}, v_c) * q_i \qquad (2)$$

i.e., the static decision chooses the argument (control vector) for which the expected quality is maximum.

4 Adaptive Parameter Controlling

If there is uncertainty about the progress of an algorithm then monitoring the algorithm's performance and revising the control strategy (depending on the features of the current solution state) can significantly improve the performance. For single point algorithms (e.g., A* or simulated annealing), the predictive profiles can be obtained using only the quality of the intermediately available solution. This methodology is not suitable for evolutionary algorithms mainly due to the following two reasons,

- Evolutionary algorithms use a population based approach (contrary to A* or simulated annealing), thus, an *aggregate* quality of a population is to be measured, and,
- Convergence of EA is not singularly dependent on the objective qualities (fitness values) of a population, characteristics of the encoding space (such as diversity [11]) can significantly influence the progress.

We design our adaptation framework using a feature based profiling. Two metrics are used to evaluate an intermediate population, *quality* (for the objective space) and *diversity* (for the encoding space). For this, we describe an intermediate solution state (*population state* S) as a vector of two elements (q, d), where q the quality level of the population and d is the diversity index.

Quality Approximation

The population quality can be approximated in various ways. For example, we can use the following heuristics,

1. The best child fitness,
2. The mean of fitness values of individual elements, and
3. A weighted average of the individual fitness values, with a bias toward better children.

Each of these techniques has some advantages and dis-advantages over the others. Using the best-child fitness, the population contribution is not accounted for. On the other hand, taking mean values as quality approximation gives undue weights to inferior children. Weighted average seems the best choice for approximation. However the time complexity of feature extraction increases, if an weighted average method is used (weighted average method has complexity $O(nlogn)$, where as other two have $O(n)$ complexity) .

Diversity Approximation

Similarly for encoding space, approximation can be done in several ways. For example, we can use the following two metrics,

1. Hamming distance based method using number of uncommon edges between individual chromosomes of the population (Genotype [11]), and
2. Standard deviation of the fitness values.

In each case, we obtain a normalized value of diversity between zero and one [11]. Using the standard deviation method ($O(n)$, n is the population size) the time complexity is less than using genotype ($O(k^2n^2)$, n is the population size and k the length of an individual string). However, high standard deviation does not necessarily mean high diversity, or vice versa, as diversity is dependent on the actual encoding and standard deviation is measured from the decoded fitness values.

Dynamic Profiling

We extend the idea of Parameterized Performance Profile (PPP) (Definition 1) to state based Dynamic Parameterized Performance Profile (DPPP) for monitoring based framework. DPPP predicts the quality improvement with time based on the state of the currently available solution and the control vector used.

Definition 2 (DPPP). *The Dynamic Parameterized Performance Profile of an anytime algorithm $P(s_j|s_i, \Delta t, v_c)$ is defined as the probability of reaching a population state s_j from the current population state s_i, if an additional time Δt is given and the control vector is set to v_c.*

The model assumes that the improvement of quality is dependent on the current population state only and not on the path to reach that solution (Markov property), which we feel, is a suitable assumption for steady state EAs.

Adaptive Strategy Generation

Next, we formulate the meta-level control strategy based on run-time monitoring. Monitoring can be continuous, assuming that the time required for monitoring to be negligible, or we can assume that each monitoring step incurs a cost, thus monitoring at each time step may not be useful [6]. We describe our framework including the monitoring cost. For this, we assign a quality penalty on each monitoring step. The controller also decides on a proper monitoring schedule along with control adjustment strategy.

Strategy 2 (Adaptive Control Strategy ($\mathbf{A_M A_C}$)) *The adaptive control strategy $A(s_i, t_k)$ is a mapping from a solution state (s_i, T_{left}) to a strategy decision $(\Delta t, v_c)$, where Δt is the time allocated before the next interruption and v_c is the control vector chosen for the interval (Δt). A solution state includes the current population state s_i and the time left before reaching the deadline (T_{left}).*

The control strategy takes a combined decision based on the following two parameters – (i) the additional time to run the algorithm (Δt) before the next interruption, and (ii) the value of the control vector for this period (v_c). The monitoring schedule is inherently generated with the choice of Δt. From the above formulation, a cost sensitive adaptive control strategy can be obtained optimizing the following Expected Quality.

Definition 3 (Expected Quality). *The Expected Quality $EQ(s_i, T_{left})$ at a solution state (s_i, T_{left}) is defined as:*

$$EQ(s_i, T) = \underset{\Delta t, v_c}{max} \begin{cases} \Sigma_j P(s_j|s_i, T_{left}, v_c) * Q(s_j) & If\ \Delta t = T_{left} \\ \Sigma_j P(s_j|s_i, \Delta t, v_c) * \\ EQ(s_j, T_{left} - \Delta t) - M_p & Otherwise \end{cases} \tag{3}$$

The probabilities are obtained from the DPPP, s_i is the current population state, T_{left} is the time left before reaching the deadline and $Q(s_j)$ is the quality component of the population state s_j. M_p is the quality penalty for each monitoring step.

We use dynamic programming to solve the above equation. The DPPP is stored as a 4-tuple $< P, s, \Delta t, v_c >$ with all its components discretized using chosen intervals. If there are x discrete states, y time intervals before the deadline and z distinct control parameter choices in DPPP, then the complexity of the dynamic programming algorithm is $O(|x|^2|y|^2|z|)$.

Maximizing the expected quality (using dynamic programming) an off-line optimal policy, for control vector choice and the monitoring schedule can be computed for each solution state (s_i, t_{left}), for a given time limit. Once an optimal policy is decided, the controller should monitor the progress of the algorithm and decide on the parameter setting based on the available solution features and the time spent.

Theorem 1. *The cost sensitive adaptive control strategy which maximizes the above expected quality function, represents an optimal strategy in terms of control setting an interrupt schedule for an anytime algorithm when quality improvement is taken to be Markov.*

Proof. The proof follows directly by the application of dynamic programming under the Markov assumption [1]. According to the Markov assumption probabilities of reaching different state levels in future depends only on the current state of the system (in this case the anytime algorithm). In our formulation, states are described using the quality and the diversity of the currently available population. The dynamic programming formulation ensures non-myopic optimality.

5 Results

We performed experiments using steady state GA for the traveling salesman problem (TSP). Individual chromosomes are encoded as integer strings. We used two cross-over strategies, namely, the partial matching cross-over and the edge recombination cross-over [7]. For mutation we used the sequence insertion strategy [11]. The individual quality values were obtained using the approximation ratio given in Eqn. 1. (Cost of the expected optimal tour is estimated using the Concorde TSP Solver[2].) To approximate the population quality, we used the metrics defined in Section 4. We found that the weighted average method is more suitable for describing the population quality. Best results were obtained using a mixed geometric series weights, gradually decreasing from better to worse children. For diversity calculations we used both

[2] http://www.tsp.gatech.edu/concorde.html

the standard deviation and genotype methods. The genotype method is more reliable for prediction as it is directly related to the encoding. However, the standard deviation method provides a good approximation with low complexity. The quality and diversity space has been discretized using 11 steps $(0-10)$ as shown in Tables 1 and 2.

Table 1. Quality Specifications

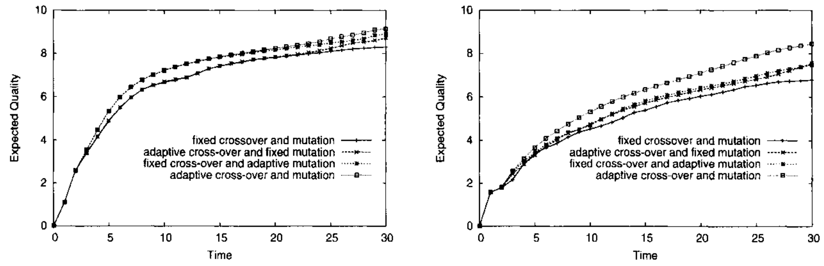
Quality Level	Approximation Ratio
10	1.05 - 1.00
9	1.10 - 1.05
8	1.15 - 1.10
7	1.20 - 1.15
6	1.25 - 1.20
5	1.30 - 1.25
4	1.40 - 1.30
3	1.60 - 1.40
2	1.80 - 1.60
1	2.00 - 1.80
0	∞ - 2.00

Table 2. Diversity Specifications

Diversity Level	Diversity Index
10	1.00 - 0.95
9	0.95 - 0.90
8	0.90 - 0.85
7	0.85 - 0.75
6	0.75 - 0.65
5	0.65 - 0.55
4	0.55 - 0.45
3	0.45 - 0.35
2	0.35 - 0.20
1	0.20 - 0.05
0	0.05 - 0.00

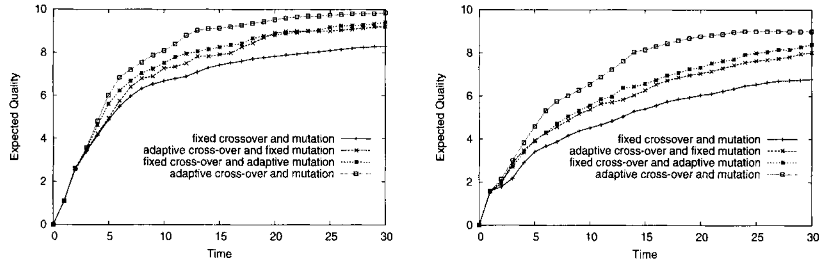
For the initial set of choices we considered five cross-over rates with a minimum of 0.6 (which is widely regarded as the minimum rate for good convergence) and others in increment of 0.1. mutation rates are chosen from set of five options $(0.1 - 0.5)$, spread in in equal intervals. For approximating the cost of monitoring, we considered the following rule – if standard deviation is used as the diversity measure, penalty is 0.01 unit per step, else if genotype method is used for diversity calculation penalty is 0.02 (as it has higher time complexity).

We obtained the DPPP using a training set of 1000 randomly chosen 100-city TSP instances. The expected quality values are calculated for the deadline specification range of $0 - 30$ units (1 time unit = 100 generations). Results obtained using weighted average method for quality approximation and standard deviation for diversity approximation are included in Figs. 1(a) and 1(b). We observe that for all possible time allocations, the expected quality is maximized using a complete adaptive strategy. The improvement increases with more time allocation. Figs. 2(a) and 2(b) include the expected results with DPPPs where genotype method is used for calculating the population diversity. Similar to earlier observations, we see an adaptive strategy is expected to perform best for both cross-overs. With genotype method, the qualities expected is generally higher than with standard deviation method, which indicates that genotype is a better metric for approximating the population diversity. In our experiments, we found that the edge recombination method almost always outperforms the partial matching cross-over method.



(a) Edge recombination cross-over (b) Partial matching cross-over

Fig. 1. Comparison of parameter control strategies for EA applied to TSP using weighted average (for quality approximation) and standard deviation (for diversity calculation) for profiling.



(a) Edge recombination cross-over (b) Partial matching cross-over

Fig. 2. Comparison of parameter control strategies for EA (edge recombination crossover method) applied TSP using weighted average (for quality approximation) and the genotype method (for diversity calculation) for profiling.

From the *initial* experimental results obtained, we observe that a dynamic profile based methodology can be a good choice for parameter adaptation. It is expected to produce better results than a fixed parameter setting validating the claim in [4]. The strategy generation is simple and the revision methodology is less time consuming than most of the adaptive and self-adaptive methods proposed. Most importantly, the methodology formalizes a profiling model and shows how the data obtained from empirical experimentations can be stored in a structured fashion and used for different environmental specifications. Obviously, the predictive framework can be made more sophisticated including other salient features of an intermediate state, but as we are considering constrained environments, the time required to extract these features may become prohibitive after a certain limit.

6 Conclusions and Future Work

In this work, we attempted a meta-reasoning framework for automatic parameter control of evolutionary algorithms under specified run-time constraints. Experiments with TSP depicted that the solution quality obtained under a given time limit, can be improved by an adaptive parameter control strategy. Although, the framework is presented for solving a single problem under time constraints, it can be extended for other constrained environments.

The framework uses a feature based prediction depending on pre-computed parameterized profiles. Efficacy of the proposed framework is dependent on the effectiveness of the feature extraction methodology. Designing improved heuristics to model intermediate state features is the current research activity. Another direction of future work is to adopt the parameter controlling strategies for multi-objective domains.

References

1. D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* Prentice-Hall, Englewood Cliffs, NJ, 1987.
2. M. Boddy and T. Dean. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67:245–285, 1994.
3. V. Cicirello and S. Smith. Modeling GA performance for control parameter optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference:GECCO-2000*, July 2000.
4. A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.
5. A. S. Fukunaga. Genetic algorithm portfolios. In *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1304–1311, La Jolla, CA, USA, 2000. IEEE Press.
6. E. A. Hansen and S. Zilberstein. Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence*, 126(1-2):139–157, 2001.
7. A. Jaszkiewicz. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137(1):50–71, 2002.
8. J. Smith and T. C. Fogarty. Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of the 7th International Conference on Evolutionary Computation*, pages 318–323, 1996.
9. L. D. Whitley. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 116–123, 1989.
10. David Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.
11. K. Q. Zhu and Z. Liu. Population diversity in permutation-based genetic algorithm. In *Proceedings 15th European Conference on Machine Learning*, pages 537-547. Springer Verlag, 2004.

Role of Chaos in Swarm Intelligence - A Preliminary Analysis

Hongbo Liu[1], Ajith Abraham[2], Yonggang Li[1], and Xiujin Yang[1]

[1] Department of Computer Science, Dalian University of Technology, Dalian, 116023, China
lhb@dlut.edu.cn

[2] IITA Professorship Program, School of Computer Science and Engineering, Chung-Ang University, Seoul 156-756, Korea
ajith.abraham@ieee.org

This paper investigates the relation between chaos and swarm intelligence. The swarm intelligent model is represented as an iterated function system (IFS). The dynamic trajectory of the particle is sensitive on the parameter values of IFS. The Lyapunov exponent and the correlation dimension were calculated and analyzed. Our preliminary research results illustrate that the performance of the swarm intelligent model depends on the sign of the maximum Lyapunov exponent. The particle swarm with a high maximum Lyapunov exponent usually achieved better performance, especially for multi-modal functions.

1 Introduction

In recent years, there has been a great interest in the relations between chaos and intelligence. Previous studies by Goldberger et al. [3], Sarbadhikari and Chakrabarty [8] illustrate that chaos has a great important influence on brain and evolutionary relationship between species etc. Recently chaotic dynamics in neural networks has also been investigated. The motivation for this research is to investigate the relation between chaos and swarm intelligence. The particle swarm provides a simple and very good case for the study. The simple swarm intelligent model helps to find optimal regions of complex search spaces through interaction of individuals in a population of particles. The model is based on a metaphor of social interaction, originally introduced as an optimization technique inspired by swarm intelligence and theory in general such as bird flocking, fish schooling and even human social behavior [5]. This paper focuses on the relationship between chaos and swarm intelligence. The particle swarm is investigated as a simple case. The swarm intelligent model is represented as an iterated function system (IFS) [9]. We simulate and analyze the dynamic trajectory of the particle based on the IFS. The Lyapunov exponent and the correlation dimension are calculated and analyzed.

2 Particle Swarm Model

A simple particle swarm model consists of a swarm of particles moving in an d-dimensional search space where the fitness f can be calculated as a certain quality measure. Each particle has a position represented by a position-vector $\mathbf{x}_i$ (i is the index of the particle), and a velocity represented by a velocity-vector $\mathbf{v}_i$. Each particle remembers its own best position so far in a vector $\mathbf{p}_i$, and its j-th dimensional value is p_{ij}. The best position-vector among the swarm so far is then stored in a vector $\mathbf{p}_g$, and its j-th dimensional value is p_{gj}. During the iteration time t, the update of the velocity from the previous velocity is determined by (1). And then the new position is determined by the sum of the previous position and the new velocity by (2).

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(p_{ij}(t) - x_{ij}(t)) + c_2r_2(p_{gj}(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

where r_1 and r_2 are the random numbers, uniformly distributed within the interval [0,1] for the j-th dimension of i-th particle. c_1 is a positive constant, called as coefficient of the self-recognition component, c_2 is a positive constant, called as coefficient of the social component. The variable w is called as the inertia factor, which value is typically setup to vary linearly from 1 to near 0 during the iterated processing. From (1), a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm. In the particle swarm model, the particle searches the solutions in the problem space with a range $[-s, s]$ (If the range is not symmetrical, it can be translated to the corresponding symmetrical range.) In order to guide the particles effectively in the search space, the maximum moving distance during one iteration must be clamped in between the maximum velocity $[-x_{max}, x_{max}]$ given in (3), and similarly for its moving range given in (4):

$$v_{ij} = sign(x_{ij})min(|x_{ij}|, x_{max}) \quad (3)$$

$$v_{ij} = sign(v_{ij})min(|v_{ij}|, v_{max}) \quad (4)$$

The value of v_{max} is $\rho \times s$, with $0.1 \leq \rho \leq 1.0$ and is usually chosen to be s, i.e. $\rho = 1$. The main pseudo-code for particle-searching is listed in Algorithm 1.

3 Iterated Function System and its Sensitivity

Clerc and Kennedy have stripped the algorithm down to a most simple form [2]. If the self-recognition component c_1 and the coefficient of the social-recognition component c_2 are combined into a single term c, i.e. $c = c_1 + c_2$, the equation can be shortened by redefining $\mathbf{p}_i$ as follows:

Algorithm 1 Particle Swarm Model

```
Initialize the size of the particle swarm $n$, and other parameters.
Initialize the positions and the velocities for all the particles randomly.
While (the end criterion is not met) do
    $t = t + 1$;
    Calculate the fitness value of each particle;
    $\mathbf{p}_g(t) = argmin_{i=1}^{n}(f(\mathbf{p}_g(t-1)), f(\mathbf{x}_1(t)), f(\mathbf{x}_2(t)), \cdots, f(\mathbf{x}_i(t)), \cdots, f(\mathbf{x}_n(t)))$;
    For $i$= 1 to $n$
        $\mathbf{p}_i(t) = argmin_{i=1}^{n}(f(\mathbf{p}_i(t-1)), f(\mathbf{x}_i(t))$;
        For $j$ = 1 to $Dimension$
            Update the $j$-th dimension value of $\mathbf{x}_i$ and $\mathbf{v}_i$
              according to (1), (4), (2), (3);
        Next $j$
    Next $i$
End While.
```

$$\mathbf{p}_i \leftarrow \frac{(c_1\mathbf{p}_i + c_2\mathbf{p}_g)}{(c_1 + c_2)} \tag{5}$$

Then the update of the particle's velocity is defined by:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + c(\mathbf{p}_i - \mathbf{x}_i(t)) \tag{6}$$

The system can be simplified even further by using $\mathbf{y}_i(t)$ instead of $\mathbf{p}_i - \mathbf{x}_i(t)$. Thus we begin with single particle by considering the reduced system:

$$\begin{cases} \mathbf{v}(t+1) = \mathbf{v}(t) + c\mathbf{y}(t) \\ \mathbf{y}(t+1) = -\mathbf{v}(t) + (1-c)\mathbf{y}(t) \end{cases}$$

This recurrence relation can be written as a matrix-vector product, so that

$$\begin{bmatrix} \mathbf{v}(t+1) \\ \mathbf{y}(t+1) \end{bmatrix} = \begin{bmatrix} 1 & c \\ -1 & 1-c \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{y}(t) \end{bmatrix}$$

Let

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{v}_t \\ \mathbf{y}_t \end{bmatrix}$$

and

$$A = \begin{bmatrix} 1 & c \\ -1 & 1-c \end{bmatrix}$$

we have an iterated function system for PSO:

$$\mathbf{P}_{t+1} = A \cdot \mathbf{P}_t$$

Thus the system is completely defined by A. Its norm $\|A\| > 1$ is determined by c. The varying curve of A dependent on c is illustrated in Figure 1. Considering the IFS, we discuss the particle swarm model with c in the interval [0.5,

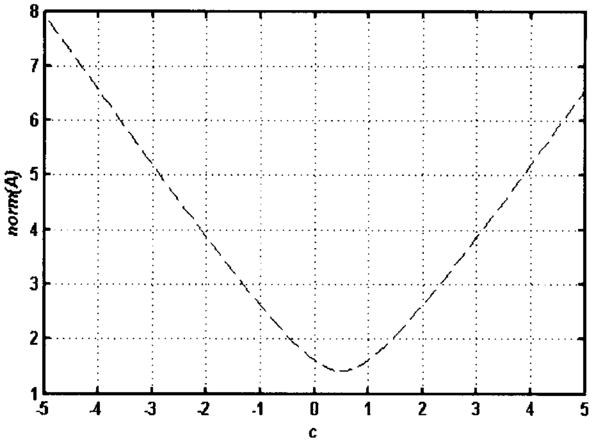


Fig. 1. Norm of A for varying values for c

4], i.e. $1.4142 < \|A\| < 5.1926$. IFS is sensitive to the values of c. We can find different trajectories of the particle for various values of c. Figure 2(a) illustrates the system for a torus when c=2.99; Figure 2(b), a hexagon with spindle sides when c=2.999; Figure 2(c), a triangle with spindle sides when c=2.9999; Figure 2(d), a simple triangle when c=2.99999. As depicted in Figure 2, the iteration time-step used is 2000 for all the cases.

4 Dynamic Chaotic Characteristics

Chaotic dynamics is defined by a deterministic system with non-regular, chaotic behavior [7]. They are both sensitive to initial conditions and computational unpredictability. The Lyapunov exponent and correlation dimension are most accessible in numerical computations based on the time-series of the dynamical system [6]. In this section, we introduce the algorithm to compute the Lyapunov exponent and correlation dimension for quantitative observation of dynamic characteristics of the particles, and then analyze the relation between chaos and the swarm intelligent model.

4.1 Lyapunov Exponent

Lyapunov exponents provide a way to identify the qualitative dynamics of a system. This is because they describe the rate at which neighboring trajectories converge or diverge (if negative or positive, respectively) from one another in orthogonal directions. If the dynamics occur in an n-dimensional system, there are n exponents. The sum of the Lyapunov exponents is the rate of system expansion. Since chaos can be defined as divergence between

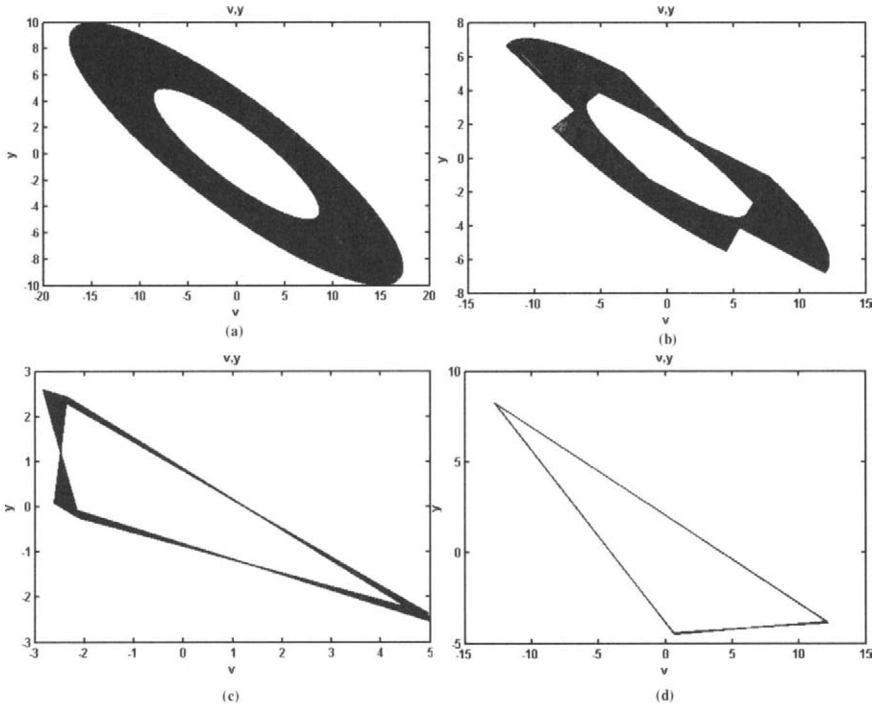


Fig. 2. Trajectory of the particle (a) $c = 2.99$, (b) $c = 2.999$, (c) $c = 2.9999$, (d) $c = 2.99999$.

neighboring trajectories, the presence of a positive exponent is the diagnostic of chaos. For an IFS, Lyapunov exponents measure the asymptotic behavior of tangent vectors under iteration. The maximum Lyapunov exponent can be found using [11]:

$$\lambda_1 = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} log_2 \left(\frac{d_n}{d_1} \right) \tag{7}$$

Where d_n is the distance between the n-th point-pair. λ_1 can be calculated using a programmable calculator to a reasonable degree of accuracy by choosing a suitably large "N". We calculated the maximum Lyapunov exponent of the IFS and is illustrated in Figure 3. The maximum Lyapunov exponent steadily increases with the value of c in the interval [0.5, 4].

4.2 Correlation Dimension

The dimension in a chaotic system is a measure of its geometric scaling property or its "complexity" and has been considered as the most basic property.

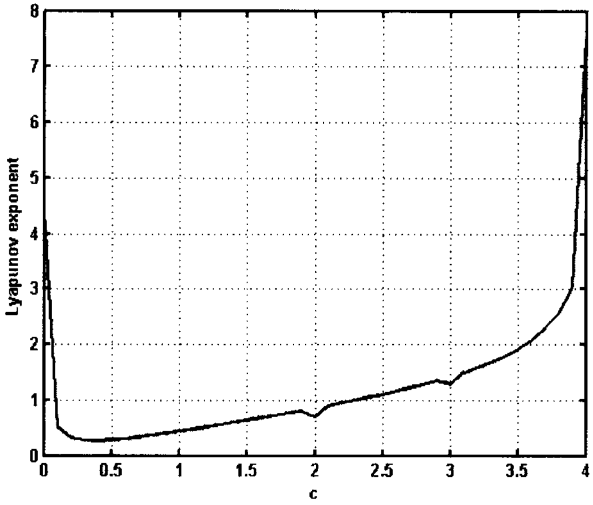


Fig. 3. Lyapunov exponent in PSO

Numerous methods have been proposed for characterizing the dimension produced by chaotic flows. The most common metrics is the correlation dimension, popularized by Grassberger and Procaccia [4]. During the past decades, several investigators have undertaken nonlinear analysis using Grassberger and Procaccia's algorithm to evaluate the correlation dimension of time-series data [10, 1].

Given by N points $\{x_1, x_2, \cdots, x_N\}$, under iteration of IFS, the correlation integral is defined by (8):

$$D = -\lim_{r \to 0} \frac{lnC(r)}{ln(r)} \tag{8}$$

In practice, $C(r)$ is calculated for several values of r and then a plot is drawn for $lnC(r)$ versus $ln(r)$ to estimate the slope, which then approximates the correlation dimension D_2. When $c = 3.9$, the slope, i.e. D_2 is illustrated in Figure 4. The correlation dimension is depicted in Figure 5. There are no obvious differences for c values increasing in the interval [0.5, 4]. D_2 is fluctuating mainly within 1 ± 0.2.

4.3 Discussions

For analyzing the relation between chaos and the swarm intelligent model, we optimized two unconstrained real-valued benchmark functions, and then investigated contrastively the performance of the model with the dynamic chaotic characteristics. One is the sphere function, given in (9). It is a continuous, unimodal function, $\mathbf{x}^* = (0, \cdots, 0)$, with $f(\mathbf{x}^*) = 0$. The other is the

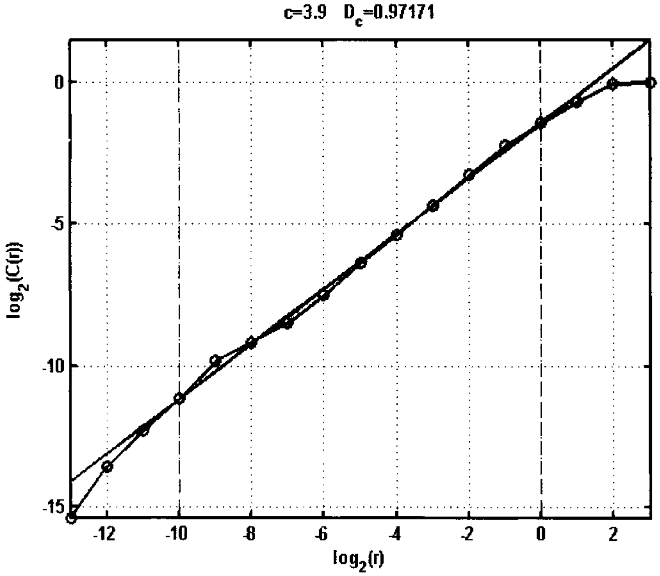


Fig. 4. $lnC(r) - ln(r)$ curve ($c = 3.9$)

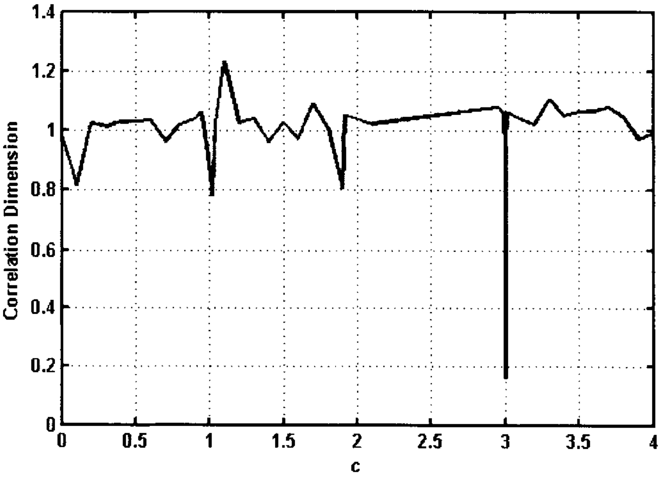


Fig. 5. Correlation dimension for varying values of c

Rastrigin's function, given by (10). It is a continuous, multimodal function with multiple local minima. And it has a "large scale" curvature which guides the search towards the global minimum, $\mathbf{x}^* = (0, \cdots, 0)$, with $f(\mathbf{x}^*) = 0$.

$$f(\mathbf{x}) = \sum_{n=1}^{n} x_i^2 \tag{9}$$

$$f(\mathbf{x}) = \sum_{n=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10] \tag{10}$$

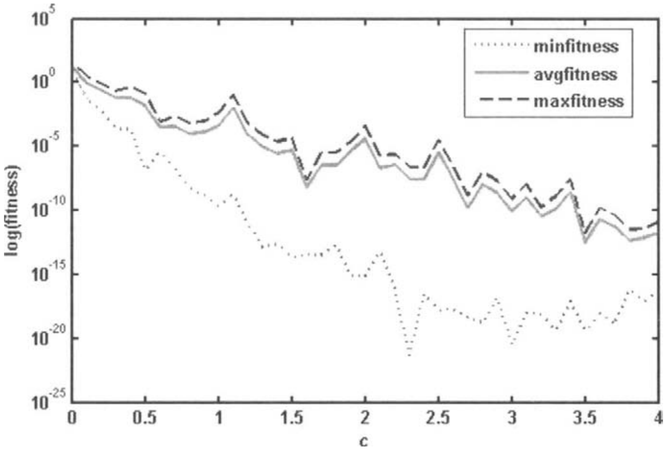


Fig. 6. The performance curve for varying values of c for 5-D Sphere function

For the two functions, the goal of particle swarm is to find the global minimum. In our experiments, V_{max} and X_{max} are set to 5.12. All experiments for both functions were run 10 times, and the maximum fitness (maxfitness), minimum fitness (minfitness) and the average fitness (avgfitness) were recorded. The swarm size is set at 10, and 200 iterations for sphere function and the result are illustrated in Figure 6. The swarm size is set at 20, and 2000 iterations for Rastrigin's function and the results are illustrated in Figure 7. Compared to the results showed in Figure 3, it is obvious that the particle swarm with a high maximum Lyapunov exponent usually achieved better performance, especially for the multi-modal functions, as showed in Figure 7. The positive Lyapunov exponent describes the rate at which neighboring trajectories diverge. A high Lyapunov exponent in the particle swarm system implies that the particles are inclined to explore different regions and find the better fitness values. Since the dimension of the particle swarm is determined by the

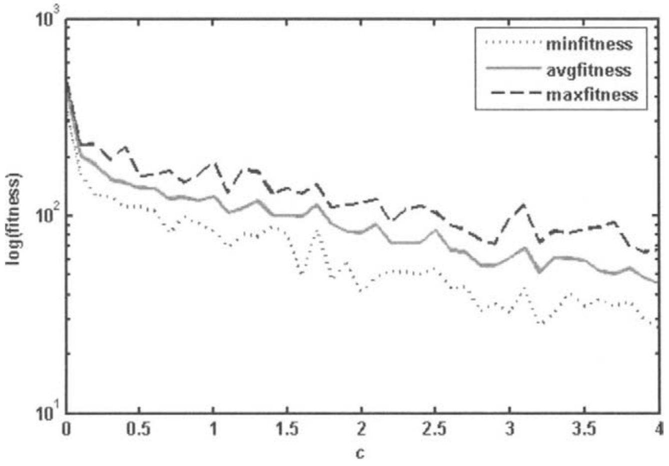


Fig. 7. The performance curve for varying values of c for 5-D Rastrigrin function

objective function, there is no significant difference in the correlation dimension. The explicit relation between correlation dimension and the performance of particle swarm can not be found in our experiments. It certainly deserves some further study.

5 Conclusions and Future Work

In this paper, we focused on the relation between chaos and swarm intelligence. The particle swarm was investigated as a simple case and the swarm model was represented by iterated function system (IFS). The dynamic trajectory of the particle was sensitive on the value of the IFS parameters. We introduced the algorithm to compute the Lyapunov exponent and correlation dimension for quantitative observation of dynamic characteristics of the particles, and then analyzed the relation between chaos and the swarm intelligent model. The results illustrated the performance of the swarm intelligent model depended on the sign of the maximum Lyapunov exponent. The particle swarm with a high maximum Lyapunov exponent usually achieved better performance, especially for the multi-modal functions.

It is noted that the real intelligent model is more complex than one which we investigated in the present paper. But it provided more aspects for further research on swarm intelligence. There are at least two works for future: 1) We could introduce chaos to overcome the problem of premature convergence in PSO, which would enjoy the ergodicity, stochastic behavior, and regularity of chaos to lead particles' exploration. Taking advantage of this characteristic feature of the chaotic system, more efficient approaches for maintaining the population diversity could be designed for some interesting problems. 2) We could design more iterated function systems to construct better models or algorithms.

Acknowledgments

This work is supported by NSFC (60373095), MOE (KP0302) and MOST (2001CCA00700). The second author acknowledges the support received from the International Joint Research Grant of the IITA (Institute of Information Technology Assessment) foreign professor invitation program of the Ministry of Information and Communication, South Korea.

References

1. Chlouverakis K E, Sprott J C (2005) A comparison of correlation and Lyapunov dimensions. Physica D. 200:156-164.
2. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1):58-73.
3. Goldberger A L, West B J, Rigney D R (1990) Chaos and fractals in human physiology, Scientific American 262:42-49.
4. Grassberger P, Procaccia I (1983) Characterization of strange attractors. Physical Review Letters 50:346-349.
5. Kennedy J, Eberhart R (2001) Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, CA.
6. Mosekilde E (1996) Topics in Nonlinear Dynamics, World Science, London.
7. Mullin T (1993) The Nature of Chaos. Claredon Press, Oxford.
8. Sarbadhikari S N, Chakrabarty K (2001). Chaos in the brain: a short review alluding to epilepsy, depression, exercise and lateralization. Medical Engineering and Physics, 23:445-455.
9. Sebastian B, Pascal H (2004) Logic programs, iterated function systems, and recurrent radial basis function networks. Journal of Applied Logic, 2:273-300.
10. Stefanovska A, Strle S, Kroselj P (1997) On the overstimation of the correlation dimension. Physics Letters A. 235:24-30.
11. Wolf A, Swift J B, Swinney L H, Vastano J A (1985) Determining Lyapunov exponents from a time series. Physica D. 16:285-317.

Multi-parent Recombination Operator with Multiple Probability Distribution for Real Coded Genetic Algorithm

M.M. Raghuwanshi

RCERT,Chandrapur. (M.S.), Pin: 442403, India.
m_raghuwanshi@rediffmail.com

O.G. Kakde

VNIT,Nagpur. (M.S.), India.
ogkakde@vnitnagpur.ac.in

Abstract. In order to solve real-world optimization problems using real-coded genetic algorithm (RCGA), up to the level of satisfaction there have been attempts with hybrid crossover operators, replacement schemes, selection schemes and adaptive crossover operator probabilities. It is also possible to solve them by using efficient crossover (or recombination) operator. This operator can be a specialized to solve for particular type of problems. The neighborhood-based crossover operators used in RCGA are based on some probability distribution. In this paper, multi-parent recombination operators with polynomial and/or lognormal probability distribution are proposed. The performance of these operators is investigated on commonly used unimodal and multi-modal test functions. It is found that operators with multiple probability distributions are capable to solve problems very efficiently. The performance of these operators is compared with the performance of other operators. These operators are performing better than other operators.

1. Introduction

Many real-world problems that could be transformed into optimization problems have complex search landscapes. These landscapes are multi-modal, epistatic and having strong local minima. RCGA possess several characteristics that are desirable to solve optimization problems. The recombination operator is main search operator in the GA as it exploits the available information about previous samples to influence future searches thus explores new search areas. The detailed study of many recombination and mutation operators can be found elsewhere [1][2].

To solve real-world optimization problems up to the level of satisfaction using RCGA there have been attempts with hybrid crossover operators, replacement schemes, selection schemes and adaptive crossover operator probabilities. Ballester & Carter [3] proposed vSBX(version of the simulated binary crossover) operator with scaled probabilistic crowding replacement scheme (SPCGA) to solve multi-modal problems. Ono *et al.* [4] proposed a robust real-coded GA (UX, UNDX)+EMGG to solve multi-modal and epistatic problems. This GA employs both the Unimodal Normal Distribution crossover (UNDX)[5] and the Uniform crossover (UX), complementary crossover operators with self-adaptive mechanism of crossover probability to choose crossover operator efficiently. Herrera *et al.* [6] proposed hybrid crossover operators that generate two offspring for every pair of parents, each one from a different crossover operator. Deb *et al.* [7] proposed a real coded steady state GA with Generalized Generation Gap (G3) model and new vector based multi-parent crossover operator PCX (parent-centric recombination operator). They have shown that G3-PCX has better convergence than gradient methods on some unimodal functions. Lozano & Herrera [8] proposed Uniform Fertility selection (UFS) and Negative Assortative mating (NAM) techniques for diversification. They have shown that UFS along with NAM and Parent centric BLX-α (blend crossover) Crossover operator (PBX-α) gives a robust scheme to solve test functions with different characteristic.

The neighborhood-based crossover operators used in RCGA are based on uniform, polynomial, triangular or lognormal probability distribution. It is interesting to note that, more efficient results can be produced, when nature of probability distribution suits to the structure of problem to be solved or when nature of probability distribution adjusted according to the structure of problem to be solved.

In the proposed work multi-parent recombination operator with mixed probability distribution and hybrid recombination operator are designed. Multi-parent polynomial distribution recombination operator (MPX) and Multi-parent lognormal distribution recombination operator (MLX)[9] are multi-parent extensions of the simulated binary crossover operator (SBX)[10] and the SBX with lognormal distribution (SBX-l)[11] respectively. Multi-parent polynomial & lognormal distribution recombination operators (MPLX & MMX) are based on the mix of two probability distributions. In multi-parent hybrid recombination operators (MHX) there are two channels to create offspring each based on lognormal or polynomial probability distribution.

This paper is organized as, Section 2 covers discussion on MPX & MLX. Section 3 describes experimentation methodology like algorithm, test problems. Section 4 gives design of multi-parent operators with polynomial and/or lognormal distribution. Section 5 covers discussion on empirical results of proposed multi-parent recombination operators and their comparison with other operators and algorithms. Finally we draw some conclusion in Section 6.

2. Real-Parameter Crossover Operators

Any good search algorithm desires exploration of large search space in the beginning and the search should then narrow down as it converges to the solution. If more than one parent is used in the perturbation process, the range of perturbation may be adaptive and can be determined from the diversity of the parents on the fly. In self-adaptive recombination operators the extent of perturbation is controlled at run time. Operators like SBX, UNDX have been tested for self-adaptive behavior [12].

2.1 Exploration and Exploitation

Real-parameter crossover operators are capable to produce exploration or exploitation (to different degrees) depending on the way in which they handle the current diversity of the population. They may either generate additional diversity starting from the current one (therefore exploration takes place) or use this diversity for creating better elements (therefore exploitation come into force). This is possible because of their self-adaptive features [6]. The performance of an RCGA on a particular problem will be strongly determined by the degree of exploration and exploitation associated to the crossover operator being applied.

2.2 Multi-parent Polynomial Distribution Crossover Operator (MPX) and Multi-parent Lognormal Distribution Crossover Operator (MLX)

MPX is a multi-parent extension of the simulated binary crossover (SBX) operator. MLX is a multi-parent extension of the simulated binary crossover with lognormal distribution (SBX-l) operator.

A prototype algorithm for MPX or MLX operator is as follows:

- From population select the best parent and (μ-1) other random solutions.
- For each gene (i=1,n) in real-parameter chromosome execute the following steps

1. Choose u_i randomly from the interval [0,1].
2. Compute β_i for MPX using

$$\beta_i = \begin{cases} (2u_i)^{1/(\eta+1)} & \text{if } u_i \le 0.5 \\ \left[1/(2(1-u_i))\right]^{1/(\eta+1)} & \text{Otherwise} \end{cases} \tag{1}$$

Or compute β_i for MLX using

$$\beta_i = \begin{cases} e^{-zu_i} & \text{if } u_i \le 0.5 \\ e^{zu_i} & \text{Otherwise} \end{cases} \tag{2}$$

3. Calculate

$$D = \left(\sum_{k=1}^{\mu} \left(\sum_{j=1}^{\mu} |x_i^j - x_i^k| \right) / \mu \right) / \mu \tag{3}$$

4. Generate two genes around gene of best parent (say x^1) using

$$y_i = x_{i1}^1 \pm (\beta_i * D) \tag{4}$$

Operators based on polynomial distribution are more exploitative and exploitation range decreases with increase in η. Operators with lognormal distribution are more explorative i.e operator is capable to generate offspring away from the parent. Its exploration range increases with increase in η.

3. Experimental Setup

Generalized Generation Gap (G3) Model is a steady state elite preserving, scalable, and computationally fast population alteration model proposed by Deb *et al.* [7]. G3 model without mutation operation is used for the simulation. Population size (N) of 100 is used. For parametric study, crossover probability parameter (p_c) varies from 0.3 to 0.8 in step of 0.1; distribution index (η) varies from 1.0 to 5.0 in step of 1.0 and no of parents (μ) takes value from 3 to 20. p_c =0.5 means on an average 50% of genes get crossed and rest of them passed unchanged to the offspring solution. The stopping criteria are: either a maximum 10^6 function evaluations or an objective value of 10^{-20} is obtained. Following parameters are recorded for each setting of (p_c, η, μ) after 50 runs:

1. Average number of function evaluation (ANFE)
2. Average fitness
3. Number of runs converged to global minima

The performance of operators is analyzed on these parameters. Best results obtained for particular set of parameters are tabulated for each operator.

The test suit contains 20 (n=20) variable unimodal and multi-modal functions with or without epistasis among variables. These functions are evaluated for global minima. It is shown that an initial population center around the true optimum produces an undesired bias for some recombination operators. To avoid this, for experimentation all variable are initialized at random within [-10, -5].

Table 1. Test functions

Function Name	Function
Ellipsoidal	$f_{elp}(x) = \sum_{i=1}^{n} ix_i^2$
Schwefel	$f_{sch}(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$
Rosenbrock	$f_{ros}(x) = \sum_{i=1}^{n-1} (100 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$
Rastrigin	$f_{rtg}(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i))$
Griewangk	$f_{gri} = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Ackley	$f_{ack}(x) = -20 \cdot \exp\left(- 0.2 \cdot \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2.0 * 3.14 * x_i) \right) + 20 + \exp(1)$

4. Multi-parent Crossover Operators

Simulation run on unimodal functions using MPX operator with different (p_c, η, μ) have shown that polynomial distribution gives better results with low value of p_c (0.3 to 0.5) and small value of η (1 or 2). Test results with MLX operator shows that it is capable to solve both unimodal and multi-modal problems. For unimodal functions best results are obtained with low value of p_c (0.3 to 0.5) and small value of η (1 or 2). For multi-modal functions best results are obtained with moderated value of p_c (0.4 to 0.6) and large value of η (3 or 4). Study on μ suggests the range of 4 to 8 is better for convergence to global minima.

4.1 Multi-parent Multiple Probability Distribution Operator (MMX)

MMX is Multi-parent crossover operator with polynomial & lognormal distributions with same value of η for both probability distributions. For given value of η & u_i calculate β_i for each distribution. To generate offspring gene mix values of β_i in some (p:l) proportions, where l=100-p and p takes value from 10 to 90 in step of 10. The following is a prototype algorithm for MMX operator with 50:50 proportions.

- From population select the best parent and (μ-1) other random solutions.
- For each gene (i=1,n) in real-parameter chromosome execute the following steps

1. Choose u_i randomly from the interval [0,1].
2. Compute β_i (sayβ_1) using (1).
3. Compute β_i (sayβ_2) using (2).
4. Calculate D using (3)
5. Generate two genes around gene of best parent (say x^1) using (5)

$$y_i = x^1_{i1} \pm ((0.5 * \beta_1) + (0.5 * \beta_2)) * D \tag{5}$$

Table 2. Test results of MMX operators

Function	p_c	η	μ	ANFE	Avg. Fitness	Success
Ellipsoidal	0.4	1	6	5665.7	9.55E-21	(50/50)
Schwefel	0.4	2	4	52027.2	9.84E-21	(50/50)
Rosenbrock	0.6	2	5	868806	9.98E-21	(50/50)
Rastrigin	0.6	4	5	129077	0	(50/50)
Griewangk	0.4	4	6	908563	0.044961	(5/50)
Ackley's	0.5	4	5	1000002	2.28E-14	(0/50)

Table 2 shows that, MMX with 70:30 proportions converged to global minima in all runs for all functions except for Griewangk's & Ackley's function.

4.2 Multi-parent Polynomial and Lognormal Distribution Based Recombination Operator (MPLX)

Test results of MPX, MLX, and MMX operator shows that they have better performance with low value of η (like1or 2) for polynomial distribution and high value of η (like 3 or 4) for lognormal distribution. MPLX is Multi-parent crossover operator with polynomial & lognormal distribution with η=1 for Polynomial distribution and η=4 for Lognormal distribution. To generate offspring gene, mix values of β_i in (p:l) proportions, where l=100-p and p takes value from 10 to 90 in step of 10. The following is a prototype algorithm for MPLX operator with 50:50 proportions.

- From population select the best parent and the (μ-1) other random solutions.
- For each gene (i=1,n) in real-parameter chromosome execute the following steps

1. Choose u_i randomly from the interval [0,1].
2. Compute β_i (sayβ_1) with η =1.0 using (1).

3. Compute β_i (sayβ_2) with η =4.0 using (2).
4. Calculate D using (3)
5. Generate two genes around gene of best parent (say x^1) using (5)

Table 3. Test tesults of MPLX operator

Function	p_c	μ	ANFE	Avg. Fitness	Success
			30:70 proportions		
Ellipsoidal	0.4	5	60591.1	9.14E-21	(50/50)
Schwefel	0.4	5	327390	9.79E-21	(50/50)
Rosenbrock	0.4	5	6.79E+06	1.9891	(48/50)
Rastrigin	0.5	5	135684	0	(50/50)
Griewangk	0.4	5	8.12E+06	0.038589	(10/50)
Ackley's	0.5	5	1000002	8.88E-15	(0/50)
			70:30 proportions		
Ellipsoidal	0.4	5	34969.9	9.05E-21	(50/50)
Schwefel	0.4	5	273974	9.69E-21	(50/50)
Rosenbrock	0.4	5	2.81E+06	9.99E-21	(50/50)
Rastrigin	0.6	6	170979	0	(50/50)
Griewangk	0.6	4	8.65E+06	0.038289	(7/50)
Ackley's	0.6	5	1000002	9.33E-15	(0/50)

Table 3 shows that, MPLX (70:30) performed better than MPLX (30:70).

4.3 Multi-parent Hybrid Recombination Operator (MHX)

In MHX operator two channels are used to generate offspring, each based on polynomial or lognormal distribution. The chromosome formation process is hybrid in the sense that genes are generated from either of the channels. The contribution of each channel is set to some proportion. Operator is tested for (p:l) proportions, where l=100-p and p takes value from 10 to 90 in step of 10. The following is a prototype algorithm for MHX operator with 30:70 proportions.

- From population select the best parent and (μ-1) other random solutions.
- In chromosome generation process, the genes contribution of polynomial distribution channel is 30 % and for lognormal distribution channel is 70%. For each gene (i=1,n) in real-parameter chromosome execute the following steps

1. Choose u_i randomly from the interval [0,1].
2. Compute β_i with η=1.0 using (1) or with η=4.0 using (2).
3. Calculate D using (3)
4. Generate two genes around gene of best parent (say x^1) using (4)

Table 4. Test results of MHX operator

Function	p_c	μ	ANFE	Avg. Fitness	Success
		30:70 proportion			
Ellipsoidal	0.5	4	57796	9.10E-21	(50/50)
Schwefel	0.5	4	279487	9.75E-21	(50/50)
Rosenbrock	0.5	4	5.22E+06	9.99E-21	(50/50)
Rastrigin	0.5	6	881027	0.099502	(46/50)
Griewangk	0.5	5	9.10E+06	0.043183	(5/50)
Ackley's	0.4	5	1000002	8.88E-15	(0/50)
		70:30 proportion			
Ellipsoidal	0.5	4	17755.8	9.14E-21	(50/50)
Schwefel	0.5	4	84521.6	9.80E-21	(50/50)
Rosenbrock	0.5	4	983507	9.99E-21	(50/50)
Rastrigin	0.5	6	8.80E+06	2.96515	(6/50)
Griewangk	0.5	5	8.52E+06	0.032231	(9/50)
Ackley's	0.6	6	1000002	7.55E-15	(0/50)

Table 4 shows that, MMX-37 has given best performance for all functions except Griewangk function.

5. Discussion

MMX, MPLX, and MHX operators are multi-parent recombination operators with polynomial and lognormal distribution. All these operators generate two ($\lambda=2$) offspring from μ parents.

To select robust operators we emphasized on number of successful run, average number of function evaluation and average fitness. We have selected three operators for this category:

- MPLX with p_c=0.6, μ=4 and 70:30 proportion.
- MHX with p_c=0.5, μ=6 and 30:70 proportion.
- MHX with p_c=0.5, μ=5 and 70:30 proportion.

Performance of these operators is compared with results given in literature [3] for SPCGA (N, λ) with vSBX-0.01 operator. The results are shown in table 5. MHX (0.5,5,70:30) performs better for Ellipsoidal, Schwefel's and Rosenbrock's function.

We have compare the performance of proposed robust operators with results of NAM, UFS & PBX based algorithm taken from literature [8]. For these 50 runs,

Table 5. Performance comparision with SPCGA

Fun	Operator	ANFE	Av. Fitness	Success
f_{elp}	MPLX(0.6,4,70:30)	101954	9.19E-21	(50/50)
	MHX(0.5,6,30:70)	78781.1	9.18E-21	(50/50)
	MHX(0.5,5,70:30)	19917.7	8.95E-21	(50/50)
	SPCGA(6,1)	50952	--	(10/10)
f_{sch}	MPLX(0.6,4,70:30)	402142	9.66E-21	(50/50)
	MHX(0.5,6,30:70)	337547	9.74E-21	(50/50)
	MHX(0.5,5,70:30)	87046.8	9.71E-21	(50/50)
	SPCGA(6,1)	294231	--	(10/10)
f_{ros}	MPLX(0.6,4,70:30)	7.09E+06	9.99E-21	(50/50)
	MHX(0.5,6,30:70)	6.04E+06	9.99E-21	(50/50)
	MHX(0.5,5,70:30)	1.03E+06	9.99E-21	(50/50)
	SPCGA(12,1)	--	--	(0/50)
f_{rtgt}	MPLX(0.6,4,70:30)	510166	0.039801	(48/50)
	MHX(0.5,6,30:70)	881027	0.099502	(46/50)
	MHX(0.5,5,70:30)	9.60E+06	4.13927	(2/50)
	SPCGA(40,3)	721401	--	(10/10)
f_{gri}	MPLX(0.6,4,70:30)	8.65E+06	0.038289	(7/50)
	MHX(0.5,6,30:70)	9.22E+06	0.037242	(4/50)
	MHX(0.5,5,70:30)	8.52E+06	0.032231	(9/50)

each one with a maximum of 100000 evaluations, applied on each test function. The results are shown in table 6. MHX (0.5,5,70:30) performs better for Ellipsoidal, Schwefel's, Rosenbrock's, and Rastrigin's function.

Table 6. Performance Comparison with NAM-UFS-PBX-0.8 based GA

Operator	f_{elp}	f_{sch}	f_{ros}	f_{rtg}	f_{gri}
MPLX(0.6,4,70:30)	1.66E-19	0.00418	91.15	0.024875	0.0459
MHX(0.5,6,30:70)	3.33E-26	0.00039	42.68	0.074626	0.0372
MHX(0.5,5,70:30)	1.63E-114	5.0E-23	3.904	3.85569	0.0424
NAM-UFS-PBX-0.8	1.44E-81	6.0E-08	14.7	30.8	0.0068

6. Conclusion

The empirical results shows that, the incorporation of multiple probability distributions and multiple parents in the design of recombination operator has produced efficient operators. As mentioned earlier, polynomial distribution is more exploitative and lognormal distribution is more explorative in nature. The combination of these two has produced search-bias that helped operators to overcome number of local minima. Also uneven proportion of contributions adjusted the search-bias that suits to the structure of problems to be solved. It is also observed that 70:30 proportions of polynomial distribution to lognormal distribution have efficiently solved some multi-modal test problems.

Test of proposed operators on other multi-modal test functions, multi-objective test functions and real-world optimization problems is also a subject of future study.

References

[1] Deb K (2001) Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, New York.

[2] Herrera F, Lozano M, Sánchez AM (2003) A taxonomy for the crossover operator for real-coded genetic algorithms, an experimental study. International Journal of Intelligent Systems 18(3): 309-338.

[3] Ballester PJ and Carter JN (2004) An Effective Real-Parameter Genetic Algorithm for Multimodal Optimization. In: ACDM conference (April-04, Bristol, UK). Adaptive Computing in Design and Manufacture VI. I.C. Parmee (Ed.), Springer, pp.359-364

[4] Ono I, Kita H, and Kobayashi S (1999) A Robust Real-Coded Genetic Algorithm using Unimodal Normal Distribution Crossover Augmented by Uniform Crossover: Effects of Self-Adaptation of Crossover Probabilities. In: the Genetic and Evolutionary Computation Conference (GECCO-1999), Morgan Kaufmann, San Mateo, CA, pp. 496-503

[5] Ono I, Kobayashi S (1997) A real-coded genetic algorithm for functional optimization using unimodal normal distribution crossover. In: Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-7), pp 246-253.

[6] F. Herrera, M. Lozano, and A.M. Sánchez (2004) Hybrid Crossover Operators for Real-Coded Genetic Algorithms: An Experimental Study. Soft Computing, in press

[7] Deb K, Anand A, Joshi D (2002) A computationally efficient evolutionary algorithm for real parameter optimization. Evolutionary Computation Journal 10(4): 371-395.

[8] Lozano M, Herrera F, and Martínez C (2004) Diversification Techniques as a Way to Improve the Performance of Real-Parameter Crossover Operators: The Case of Parent-Centric BLX-α. Technical Report SCI2S-2004-07. Research Group on Soft Computing and Intelligent Information Systems. University of Granada

[9] Raghuwanshi MM, Kakde OG (2005) Gene-Level Multi-parent Recombination operator with Polynomial or Lognormal Distribution for Real Coded Genetic Algorithm. The Genetic and Evolutionary Computation Conference-2005 (GECCO-2005) (accepted)

[10] Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. Complex System 9,pp 115-148.

[11] Raghuwanshi MM, Singru PM, Kale U, Kakde OG (2004) Simulated Binary Crossover with Lognormal Distribution. In: Proceedings of the 7th Asia-Pacific Conference on Complex Systems (Complex 2004).

[12] Deb K, Beyer H (2001) Self-adaptive genetic algorithms with simulated binary crossover. Evolutionary Computation Journal 9(2): 195-219.

Part IX

Tutorials

Special Tutorial - State of the Art Face Detection: Cascade Boosting Approaches

Rodrigo Verschae and Javier Ruiz-del-Solar

Dept. of Electrical Engineering, Universidad de Chile, CHILE
Center for Web Research, Universidad de Chile. CHILE
rverscha@ing.uchile.cl, jruizd@ing.uchile.cl

Abstract. The face detection problem consists on finding the position and size of all faces (if any) appearing in an image or video. It is a complex problem, mainly because of the real-time and real-world (variability of human faces, image acquisition conditions and image background) requirements. In the present tutorial we first introduce the face detection problem and we explain its peculiarities. Then we present the most commonly employed approaches, especially the ones based on statistical learning. Finally, we describe the so-called cascade boosting approaches, which have recently solved the face detection problem in a fast and robust manner.

1. Introduction

Computational face analysis (facial expression and face recognition; face and eye detection and tracking, etc.) is an expanding research field, mainly driven by applications related with surveillance and security. Face analysis also plays an important role for building human-computer interfaces. Face detection is a key step in almost any computational task related with the analysis of faces. In this general context the aim of this tutorial is to present the most successful face detection approaches on grayscale images. We will focus on the problem of detecting frontal faces, i.e. the detection of faces with low roll, yaw and pitch rotations. For detecting rotated faces (so-called multiview face detection), the usual approach is to build several classifiers; each one detecting faces which presents a given rotation. This is not an optimal solution, but is the most employed one (see [9][6]).

2. Problem Definition

The face detection problem consists on finding the position and size of all faces (if any) appearing in an image. Face detection is a complex problem, mainly because of (1) of variability of human faces (different races, expressions, presence/absence or glasses, etc.), (2) variability in the process of the image acquisition (variable illumination, rotations, translations, scales, etc.), and (3) variability in the image background. Many applications also add the fast processing constrain (4).

Considering these four requirements, most of the face detection systems follow the structure shown in figure 1. This is a multi-scale approach in which for each scale an exhaustive search is performed. First, for detecting faces at different scales a multiresolution analysis is performed by downscaling the input images (*Multiresolution Analysis* module). Afterwards, for each of these scaled versions of the input image, windows (e.g. of size 24x24 pixels) are extracted in the *Window Extraction* module. These windows can be then processed for obtaining illumination invariance by using different techniques [8][4]. Afterwards, the preprocessed windows are analyzed by a classifier *H(x)*, which discriminates between face and non-face windows. After all windows have been classified, the *Overlapping Detection Processing* module analyzes and fuses overlapped face windows for determining the final size and position of the detections.

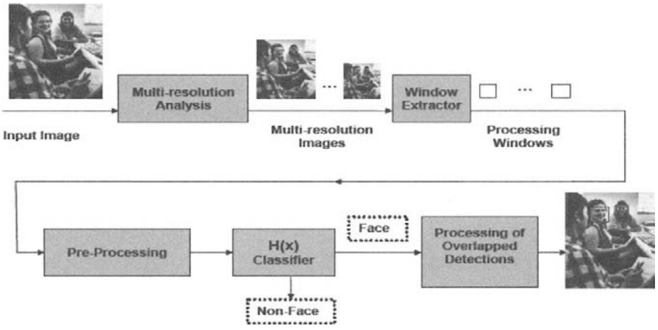


Fig. 1. Block diagram of a typical multi-scale face detection system.

3. Historical Development and State of the Art

Several approaches have been proposed for the computational detection of faces in digital images. Comprehensive reviews can be found in [3][1]. Main approaches can be classified as: (i) feature-based, which uses low-level analysis, feature analysis or active shape models, and (ii) image-based, which employs linear subspace methods, neural networks or statistical analysis. Image-based approaches have shown a much better performance than feature based [3][1]. Starting with the seminal works or Rowley [4] and Sung & Poggio [7], successful image-based proposed approaches include the use of neural networks [2][4], SNoW classifiers, SVM, and boosted cascades. The boosted cascades approach is new and is based on a learning paradigm introduced in [8], extended in the last few years, being [9][6][1] some of the most important works in this area. This paradigm consists on using a cascade of classifiers to obtain a fast system that is capable at the same time of achieving high detection rates.

Table 1 shows results for some methodologies at a given operation point for the commonly used CMU-MIT database [4]. We can notice that cascade based methods achieve processing times up to 1 order of magnitude smaller than state of the art non-cascade methodologies and that they have similar or better performance.

Table 1. Face detection results comparison: Detection Rate (DR), number of false positives (FP), average processing time for a give image size and used methodology.

Authors and Ref.	DR FP	Processing Time	Methodology
Schneiderman[6]	89.76	180ms, 320x200	Feature-Centric Cascade
Wu et al.[9]	90.110	80ms, 320x240	Boosted Nested Cascade
Delakis & Garcia[2]	90.510	~700ms, 330x240	Convolutional Neural Network.
Rowley [4]	83.210	~1000ms, 320x200	Neural Network

4. Boosted Cascades for Face Detection

The face detection problem is a classification problem with an asymmetric distribution of the classes (face and non-face), i.e. in an image there are much more non-face than face windows. Thus, the average processing time of windows is defined by the processing time of the non-face windows. Moreover, most non-face windows can be easily classified as non-faces (they don't look like faces), so we can design a classifier that dedicates less time, in average, to the "easy" non-face windows. Therefore we use a cascade of (filters) classifiers, with each filter rejecting non-face windows and letting face windows pass to the next layer of the cascade. A window is considered as a face only if it arrives to the end of the cascade. The average processing time of the non-face windows depends mainly on the false positive rate and on the processing speed of the first layers, so we design the filter *i* of the cascade to: (1) reject a large number of non-faces windows, (2) to let pass the larger possible number of face windows, and (3) to be evaluated as fast as possible. There is always a trade-off between these three objectives.

Nested cascades [9] are an improvement over standard cascades [8], and produce more accurate and fast systems. A nested cascade of boosted classifiers is composed by several integrated (nested) layers, every one containing a boosted classifier. The whole cascade works as a single classifier that integrates the classifiers of every layer (see Figure 2).

4.1 Boosted Classifiers

Boosting is a classifier building paradigm in which a so-called robust classifier is built by using different instances of a so-called base classifier. The instances are obtained by training the base classifier on different training sets or input distributions. In this article we will focus on the Adaboost algorithm [5], although other boosting methods have been used for building face detectors. Adaboost adapts a distribution of weights; each weight is associated to an example of the training set:

$$\{x_i, y_{i,} D_i\}, x \in X^N, y_i \in \{-1,+1\}, \sum_{i=1}^{n+m} D_i = 1,, i = 1,...,(n+m)$$

with y_i the class of the example x_i, and $D_i(t)$ its weight at the iteration t of Adaboost. n and m are the number of positive and negative examples respectively. At each iteration t, the base classifier is trained considering the distribution $D_i(t)$, obtaining a weak classifier (and instance of the base classifier). This base classifier is trained to minimize an exponential loss function that depends on the weights $D_i(t)$. After that, the weight distribution D_i is updated and normalized previous to the next iteration. The example's weights $D_{i,}$ are updated in a way such that the new classifier will focus more on the examples wrongly classified by the previously trained classifiers. This process is repeated until achieving the desired classification rates. The strong classifier is a weighted sum of the weak classifiers:

$$H(x) = \sum_{t=1}^{T_k} \alpha_t h_t(x)$$

The output of the boosted cascade is the sign of $H(x)$ and α_t represents the accuracy of the weak classifier $h_t(x)$. We refer to [5] for a detailed description of Adaboost and how to calculate α_t and how to update D_i.

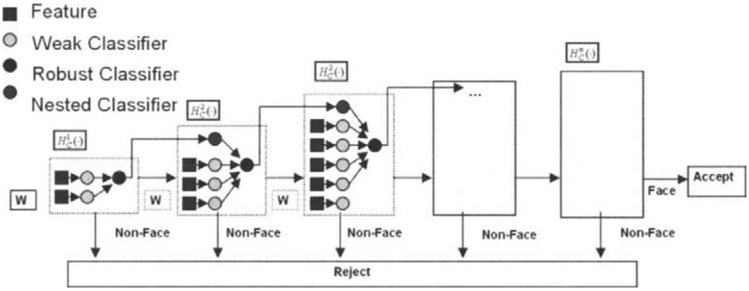


Fig. 2 Block diagram of an Adaboost nested cascade.

4.2 Weak Classifiers

The weak classifiers can be designed in different ways. In [8] a binary partition of the feature domain is used. This idea was later extended in [9], by partitioning the feature domain in several blocks and then training the weak classifier using *domain-partitioning weak hypotheses* [5]. Under this paradigm the weak classifiers make their predictions based on a partitioning of a feature domain F (or of the input domain X). F has associated feature extraction function *f(x) and* is partitioned into disjoint blocks $F_1,\ldots,F_n$, which cover all F and for which *h(f(x))=h((x'))* for all $f(x),f(x') \in F_j$. Thus, the weak classifiers prediction depends only on the block F_j the sample falls into. During training, the output obtained for each weak classifiers is stored in a LUT for speeding up its evaluation. Each weak classifier has associated a single feature (e.g. Haar-Wavelet-like [8] and LBP based [1]).

4.3 Training Boosted Cascades

When training of a cascade classifier we must consider that: (1) there is an asymmetrical distribution of the two classes and (2) a given layer of the cascade must be trained taking into account the previously trained layers. For training a face detector, every window in any image not containing faces is a valid non-face training example. Including all possible non-face patterns in the training database is not an alternative. It is Important to have training samples that correctly define the classification boundary, i.e. non-face patterns that look similar to faces. These patterns can be selected using the bootstrap procedure [7]: after a given instance of a classifier is trained, the current set of non-face examples can be enlarged with new non-face patterns that the current classification system wrongly classifies as faces and then we can train the system with the enlarged database. However, when training a cascade consisting of several layers it is not obvious where and how many times to make the bootstrap. The commonly used procedure for the training of the layers is [8]: (1) to randomly select non-face examples for the first layers and (2) for the later layers to select non-face examples that are incorrectly classified by the previously trained layers. In [1] bootstrapping is also used to train several times each layer of the cascade.

Acknowledgements

This research was funded by Millenium Nucleus Center for Web Research, Grant P04-067-F, Chile.

References

[1] Fröba B, Ernst A (2004), Face detection with the modified census transform, 6th Int. Conf. on Face and Gesture Recognition-FG 2004, pp. 91–96, Seoul, Korea, May 2004

[2] Delakis M, Garcia C (2004), Convolutional face finder: A neural architecture for fast and robust face detection, IEEE Trans. PAMI., 26(11):1408-1423

[3] Hjelmås E, Low BK (2001), Face detection: A survey, Computer Vision and Image Understanding 83: 236-274

[4] Rowley H, Baluja S, Kanade T (1998), Neural Network-Based Detection, IEEE Trans. Pattern Anal. Mach. Intell. 20:23-28

[5] Schapire RE, Singer Y (1999), Improved Boosting Algorithms using Confidence-rated Predictions, Machine Learning, 37: 297-336

[6] Schneiderman H (2004), Feature-Centric Evaluation for Efficient Cascade Object Detection, IEEE Conf. on CVPR 2004, pp 29 -36

[7] Sung K, Poggio T (1998), Example-Based Learning for Viewed-Based Human Face Deteccion, IEEE Trans. Pattern Anal. Mach. Intell. 20: 39-51

[8] Viola P, Jones M (2001), Rapid object detection using a boosted cascade of simple features, IEEE Conf. on CVPR, Kauai, HI, USA, pp 511-518

[9] Wu B, AI H, Huang C, Lao S (2004), Fast rotation invariant multi-view face detection based on real Adaboost, 6th Int. Conf. on Face and Gesture Recognition - FG 2004, pp 79–84, Seoul, Korea, May 2004.

[10] Yang M, Kriegman D, Ahuja N (2002), Detecting Faces in Images: A Survey, IEEE Trans. Pattern Analysis and Machine Intelligence 24: 34-58

Special Tutorial - Particle Swarms for Fuzzy Models Identification

K.K. Aggarwal[1], Shakti Kumar[2], Arun Khosla[3]* and Jagatpreet Singh[4]

[1]GGS Indraprastha University, Delhi – 110006, India.
[2]Haryana Engineering College, Jagadhari – 135003, India.
[3]National Institute of Technology, Jalandhar – 144011, India.
[4]Infosys Technologies Limited, Chennai – 600019, India.

The problem of fuzzy system modeling or fuzzy model identification is generally the determination of a fuzzy model for a system or process by making use of linguistic information obtained from human experts and/or numerical information obtained from input-output numerical measurements. The former approach is known as knowledge-driven modeling while the later is known as data-driven modeling. It is also possible to integrate the two approaches for developing models of complex real systems. In this tutorial, attention is focused on building optimized fuzzy model from the available data based on relatively new identification technique viz. particle swarm optimization (PSO).

The PSO technique is a member of the broad category of swarm intelligence techniques for finding optimized solutions. The motivation behind the PSO algorithm is the social behavior of animals, viz. flocking of birds and fish schooling. The algorithm traces its origin to early attempts to simulate the synchronized choreography of bird flock, which was later modified and fine-tuned for optimization. [1, 2, 3].

The tutorial reviews the concepts of fuzzy logic and fuzzy model identification process, followed by some of the commonly used techniques for fuzzy modeling [4, 5]. A framework for the identification of fuzzy models identification through PSO algorithm is presented [6].

For the presentation and validation of the approach, the data from the rapid Nickel-Cadmium (Ni-Cd) batteries charger developed by the authors has been used. The purpose of development of rapid charger was to reduce the charging time using high charging current without doing any damage to them [7, 8].

* Corresponding Author. Email: khoslaak@nitj.ac.in

In this tutorial, a Matlab based toolbox, *PSO Fuzzy Modeler for MATLAB*, is introduced. This toolbox provides the features to generate the optimized fuzzy model (Mamdani and Sugeno) from the available data automatically using PSO algorithm. This is an open-source initiative and is hosted on SourceForge.net [9].

The performance of PSO algorithm, genetic and evolutionary algorithms and other similar algorithms largely depends upon the choice of appropriate parameters. Before running the algorithm, the user is required to specify a number of parameters. Generally these parameters are selected through a hit and trial process, which is very unsystematic and requires unnecessarily rigorous experimentation. The tutorial has presented a systematic approach based on Taguchi method [10, 11, 12, 13] for the identification of the strategy parameters of PSO algorithm used for the fuzzy model identification. Taguchi method, which is a robust design approach, uses fractional factorial design that is used to study a large number of parameters with a small number of experiments.

Although the focus of the tutorial is to present the use of PSO algorithm and Taguchi method for fuzzy modeling, these techniques have much broader use and applications.

References

[1] Kennedy J, Eberhart R (2001), Swarm Intelligence, Morgan Kaufmann Publishers

[2] Parsopoulos KE, Vrahatis MN (2002), Recent approaches to global optimization problems through Particle Swarm Optimization, Natural Computing, Kluwer Academic Publishers, pp 235-306

[3] Eberhart RC, Shi Y (2001), Particle Swarm Optimization: Developments, Applications and Resources, Proceedings of the Congress on Evolutionary Computation, Seoul, Korea. pp 81-86

[4] Hellendoorn H, Driankov D (Eds.) (1997), Fuzzy Model Identification - Selected Approaches, Springer-Verlag

[5] Yen J, Langari R (2003), Fuzzy Logic - Intelligence, Control and Information, Pearson Education, First Indian Reprint

[6] Khosla A, Kumar S, Aggarwal KK (2005), A Framework for the Identification of Fuzzy Models through Particle Swarm Optimization Algorithm, To be published, IEEE Indicon, December 11-13, 2005, Chennai, India

[7] Khosla A, Kumar S, Aggarwal KK (2002), Design and Development of RFC-10: A Fuzzy Logic Based Rapid Battery Charger for Nickel-Cadmium Batteries HiPC2002 Workshop on Soft Computing, Bangalore, pp 9-14

[8] Khosla A (1997), Design and Development of RFC-10: A Fuzzy Logic Based Rapid Battery Charger for Nickel-Cadmium Batteries, M.Tech. Thesis, Kurukshetra University, Kurukshetra. India

[9] PSO Fuzzy Modeler for Matlab http://sourceforge.net/projects/fuzzymodeler

[10] Ross PJ (1996), Taguchi Techniques for Quality Engineering, McGraw Hill

[11] Bagchi TP (1993), Taguchi Methods Explained - Practical Steps to Robust Design, Prentice Hall of India

[12] Taguchi G, Chowdhury S, Wu Y (2005), Taguchi Quality Engineering Handbook, John Wiley and Sons

[13] Tsai J-T, Liu T-K, Chou J-H (2004), Hybrid Taguchi-Genetic Algorithm for Global Numerical Optimization, IEEE Transactions on Evolutionary Computation 8: 365-377

Special Tutorial - Project Management: Issues in Computer Based Monitoring and Control with Soft Computing Approaches

Gautam Vasappanavara[1], Anand Vasappanavara[2], D Ravi[3], Ramesh Vasappanavara[3]

[1] Project Engineer GE Infrastructure Ltd Hyderabad, BITS PILANI,
gappu_007@yahoo.com
[2] Project Engineer Tata Motors Pune , IISc Bangalore
iitanand@yahoo.com
[3] Professor Dept of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering, Visakhapatnam, India
ramesh_vasappanavara@yahoo.com

Abstract. Project planning and control for a large, complex, multi-layered project is a difficult process, requiring effective control tools. This task becomes all the more important when a company is embarking on a new project with no history data base backing or it is a start up company. In project control, there are several occasions, where in a project needs to be compressed to meet amended deadlines or costs. In such an event, selection of optimum PERT Chart amongst available options becomes critical. Usually organizations depend on the experience of Project Manager to bail them out of the situation. We present an innovative approach of application neural networks and fuzzy logic for selection of PERT charts amongst options available, based on the history data base and past behavior pattern of the Project Manager.

1 Introduction

The project manager invariably benefits from a large history database with previous projects categorized appropriately as belonging to specific case studies. The project manager can also draw from his own experience and intuition for laying down the rules for selection of PERT chart amongst options available. While there is no obvious way of capturing the project manager's experience and intuition, it is still possible to build an expert system based on soft rules of thumb that imitates closely the human thinking process. The history database can be used to provide answers to certain statistical questions. However, with the history database, a carefully designed expert system can answer even questions relating to projects that have never been tried before with reasonable accuracy.

For details on building and using expert system the readers are referred to [1]. We refer the reader to [3] for detailed approaches to software project estimations. In [4] authors have presented an Expert System for estimations using soft computing techniques for handling imprecision and uncertainty, [2] discusses the background and methods for project control using pert chart techniques. In this tutorial paper, we discuss soft computing approaches using fuzzy logic and neural networks, for selection of optimal PERT chart amongst available alternatives.

2 Selection of Optimal Pert Chart

This section of tutorial paper deals with application of neural networks for selection of optimal PERT charts based on the history data base and past behavior pattern of the Project Manager. In project control, there are several occasions, where in a project needs to be compressed to meet amended deadlines or costs. Owing to possible existence of parallel paths (more than one) to critical paths, we can generate more than one PERT chart by either by compressing the activities on critical path or by compressing all affecting activities on parallel paths. In such an event, selection of optimum PERT Chart amongst available options becomes critical. Usually organizations depend on the experience of Project Manager to bail them out of the situation. For selection of most suited pert chart we need signature for the chart, a suitable identification, using which we can compare charts. For this purpose, we will encode the charts using a suitable scheme and use a scheme to select one of the available options amongst pert charts. Procedure is detailed below.

Step 1: Generate identification number for pert chart: P_{id}, based on the parameters like project identification number, number of edges, edges in critical path, total cost, and cost of critical path. We can use any algorithm similar to message digest algorithm for generation unique identification number.

Step 2: A PERT chart can be represented by a matrix say AM. The fields of interest are activity id (A_{id}), predecessor, successor, earliest start time (TE), latest start time (TL), slack, cost, and crash cost.

Step 3: Store the PERT Charts generated with project identification number P_{id} and associated matrix AM.

Step 4: Select the optimal PERT chart amongst options available, using fuzzy logic or neural network approaches described in next section.

3 Issues in Comparing PERT Charts

We would like to choose a PERT Chart that affords a minimum **Total Minimum Cost (TMC).** Allocate a weight **WTC** for this cost function. Usually the project scheduling and allocation of manpower to a project has already been carried out during initial phase of the project and hence reallocations / re-appropriations would cause turbulence. Therefore we would like to select a PERT chart that has smaller requirement of additional manpower. Let Weight age allocated for manpower be **WMP** and its associated cost **MP**. We will define a factor called **Deviation Quotient (DQ)** as the variation in matrices before and after compression with weight age $\mathbf{W_{dq}}$. For example in a PERT chart with activity identification no $A_{id,}$ we have details at table 1. In this table, there are two activities that incur additional costs.

Table 1. cost/addle cost for an activity

Aid	Cost/addl. cost
2	50/75
4	100/125

Therefore DQ is a measure of number of activities affected and associated additional cost. In the above example number of activities is 2 and total additional cost incurred is 50 units. Let W1 be weight allocated for this parameter, i.e. number of activities with additional cost and let W2 be weight allocated to man power relocation parameter for additional cost incurred.

DQ = (No of activities affected * W1+additional cost*W2) / W1+W2 (1)

We define a factor for comparison of PERT charts and call it PERT Chart Quotient (PCQ)

PCQ = (TMC* Wtc + MP * Wmp + DQ * Wdq) / (Wtc + Wmp + Wdq) (2)

3.1 Neural Network Based Selection of PERT Charts

On compression or expansion let us say that **m** number of PERT charts have been produced. The procedure for optimal selection is described below.

Compute PCQ and rank PERT charts based on equations 1 & 2 shown above for a project with identity number P_{id}.

Depending on the priorities, resources, policies, and past practices make a final selection. This based on Project managers behavior pattern, intuition, and experience. For example a scenario could be to complete and deliver the project on compressed date, even by sacrificing some of the functionality. The other factors that dictate priority are quality assurance, vitality of the project etc. Availability of resources like the excess manpower available from other ongoing projects affords selection that would demand higher allocations of manpower. Policy of the company regarding delivery, quality assurance, profitability etc also would play a dominant role in final selection. Past practices by project manager and end results from history data base are excellent pointers to criteria to be followed. We would allocate cost drivers to factors such as Quality Assurance cost (QAC) and weight (WQA). Similarly consider costs and allocate weights to other relevant factors like Function Point (FPC, WFP), vitality of the project

(VITC, WVITC) and deliver date (DUC, WDUC) . We can now compute a factor called Adjusted Per Chart Quotient (APCQ)

$$APCQ = PCQ * (QAC*WQA + FPC*WFP + VITC*WVITC + DUC*WDUC) / WQA + WPC + WVITC + WDUC) \tag{3}$$

We would store such a transaction, comprising the options in PERT charts available, criteria for ranking in the form of weights and quotients, company preferences etc, and thus creating history data base. The fields in the data base are project identification (P_{id}) and associated matrix A, PCQ, TMC, Wtc, MP, Wmp, DQ, Wdq, QAC, WQA, FPC, WFP, VITC, WVITC, DUC , WDUC.

Choose a back propagation type of neural network and train the network using the history data base to accept input member pert charts and produce an output based on Project Managers signature and thinking.

3.2 Fuzzy Based Selection of Pert Charts

The Fuzzy Logic Controller, displayed at Fig 1, maps input member PERT charts, represented by project identification number (P_{id}) and matrix (AM) to output chart based on fuzzy rules and parameters like weights and costs. Input mapping function is placed at Fig 2. For each PERT chart obtain the weights and set minimum for each variable. This minimum is a fuzzy minimum chosen so as to tune the controller and is based on History data base. Set of fuzzy logic rules for selection of optimal PERT chart is at Fig. 3.

Per Chart Quotient (PCQ) is computed using fuzzy rules. PCQ is adjusted based on priorities, policies and past practices as per procedures out lined at Section 3.1, to obtain APCQ. Selection of optimum pert chart is based on APCQ.

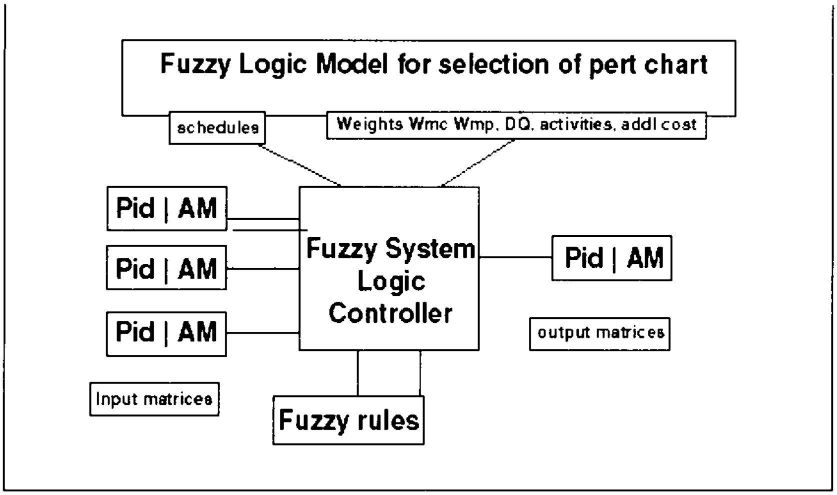

Fuzzy Logic Model for selection of pert chart
schedules
Weights Wmc Wmp, DQ, activities, addl cost
Pid | AM
Pid | AM
Pid | AM
Fuzzy System Logic Controller
Pid | AM
output matrices
Input matrices
Fuzzy rules

Fig. 1 Fuzzy logic controller

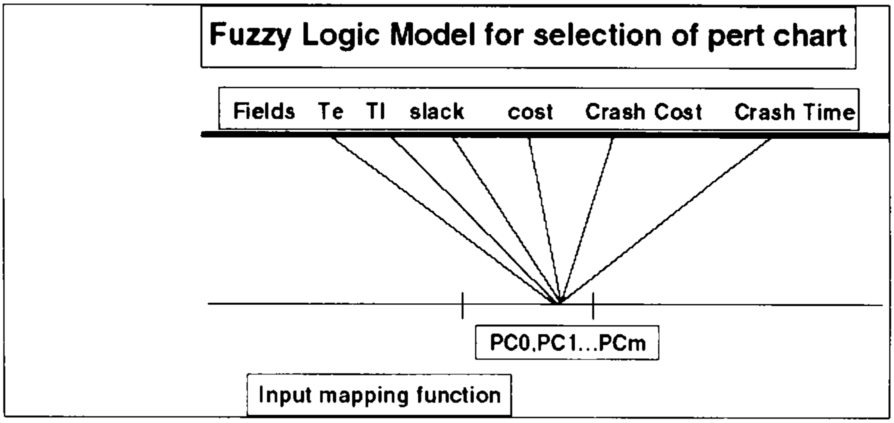
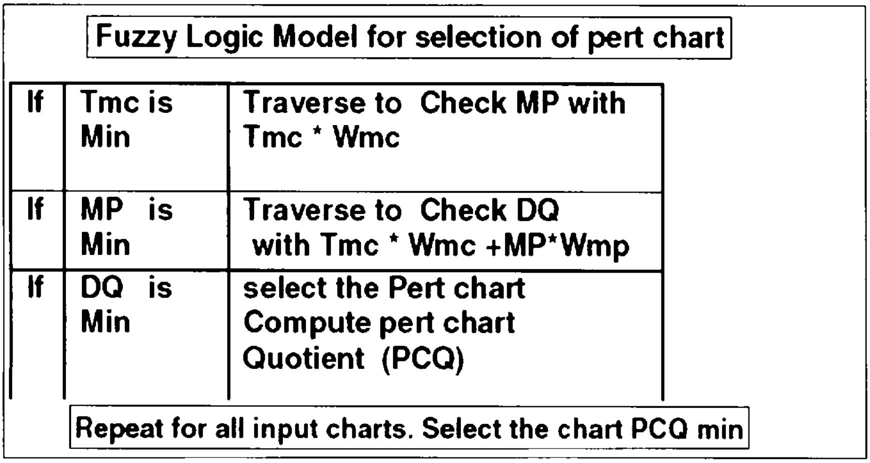

Fuzzy Logic Model for selection of pert chart
Fields Te Tl slack cost Crash Cost Crash Time
PC0,PC1...PCm
Input mapping function

Fig. 2. Input mapping form pert chart to fields



Fuzzy Logic Model for selection of pert chart

If	Tmc is Min	Traverse to Check MP with Tmc * Wmc
If	MP is Min	Traverse to Check DQ with Tmc * Wmc +MP*Wmp
If	DQ is Min	select the Pert chart Compute pert chart Quotient (PCQ)

Repeat for all input charts. Select the chart PCQ min

Fig 3. Set of fuzzy logic rules

4. Conclusions

Complex projects need firm and well defined plans, effective monitoring and control to prevent cost and time over runs. This aspect calls for reliable and accurate forecasts for project estimations like effort, duration, complexity, and costs etc and accurate project control mechanisms like PERT/CPM control. We have presented a prototype for selection of optimal pert chart amongst options available, that is based on fuzzy logic and neural network based control system. This model becomes useful when faced with a situation of compressing the PERT chart to suit changed conditions like advancement in project completion date etc. Further this model formalizes the thinking process of project manager, and thus has potential to reduce excessive dependence on an individual, and thereby reduces risk.

References

[1] Alison C (1997), Essence of Artificial Intelligence (Essence of Computing), Prentice Hall

[2] Srinath LS (1989), PERT and CPM: Principles and applications, 3rd edn. New Delhi: East-West Press Pvt. Ltd.

[3] Jalote P (2004), An Integrated Approach to Software Engineering, 2nd Edition, Narosa pub. House, pp 166-170

[4] Vasappanavara Ramesh , Ravi D , Gautam V, Anand V (2005), Fuzzy based selection of paths in a decision tree with imprecision and partial specifications, proceedings of Applied Asian Computing Conference (AACC2005) Khatmandu Nepal , Dec 10-12, accepted for publication